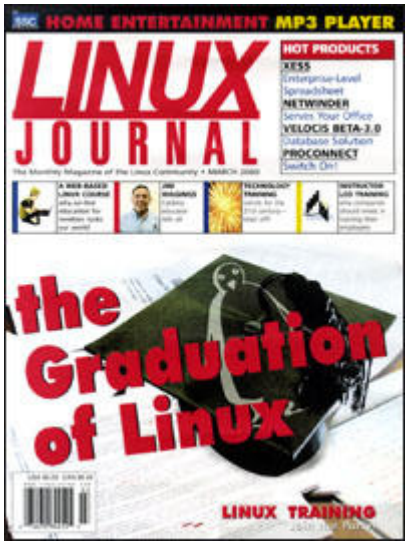
Linux Journal Issue #71/March 2000



Focus

Linux Training *by Marjorie Richardson*

Features

A Web-Based Linux Training Course *by Dr. Giovanni A. Orlando*
The president of Future Technologies describes his FTLinux Course training program for Linux.

Technology Training: Trends for the 21st Century *by Brian Holt*
A look at how training courses will be taught in the future.

Jim Higgins, Caldera Training *by Marjorie Richardson*
What's happening in the Education Department at Caldera Systems? Read on...

Instructor-Led Training *by Marty Ferguson*
Training is a sound investment networked to future growth.

Forum

Simplified Wrapper and Interface Generator *by Wael Hassan*
This article introduces the uses and advantages of SWIG.

Using Apache Proxy to Suppress Banner Ads *by Raj Mathur*
If you find Internet advertising annoying, time wasting and resource consuming, get rid of it.

Home Entertainment Linux MP3 Player *by Goran Devic*
Here's a way to store your CDs and tapes away forever, while still enjoying the music.

Linux DVD Update *by Craig Knudsen*

Reviews

Columns

Departments

Strictly On-Line

Focus: Linux Training

Marjorie Richardson

Issue #71, March 2000

Since our readers are Linux enthusiasts who would prefer to work with a Linux system rather than a Windows one, this month's focus is on Linux training.

When I was in high school, many speakers came to talk to the students about our futures: jobs and options available after graduation. One thing every one of them stressed was the importance of going on to get further education and training. Without it, we were told we would face a life of work in the food industry or as sales clerks or low-level car mechanics. We were shown charts of how much more money we would make with the proper university degree or training certificate for the job we wanted. These things are still a top priority for most people entering the work force today. Developing skills that are fun as well as attractive to business consumes the minds of students approaching graduation and entering the real world.

Since our readers are Linux enthusiasts who would prefer to work with a Linux system rather than a Windows one, this month's focus is on Linux training. Training is needed so we can get out there and actually support ourselves using our favorite operating system. Many ways of getting that training are available, and we take a look at the ones most in use: traditional classrooms and web-based courses. All are geared toward getting Linux certification so that you can prove your worthiness to potential employers. We also have updates from the two existing certification programs: Linux Professional Institute and SAIR Linux.

Columns

We hope you have been enjoying our new “Cooking with Linux” column, which premiered last September and began in earnest with the January issue. Marcel Gagné is a talented author who has written for us in the past, and we are happy to have him join us as a regular contributor. He also writes a column called “SysAdmin's Corner” which appears on our web site (http://

www.linuxjournal.com/). Marcel is a Linux geek with a nice sense of humor—what a guy!

Another web column you might like is “Currents” by Bryan Pfaffenberger, a university professor with a keen intellect, who probes the issues facing the computer and software industry today. He challenges us to think and form our own opinions about open source, copyrights, software patents and more.

—Marjorie Richardson, Editor in Chief

Contents

Archive Index Issue Table of Contents

Advanced search

A Web-Based Linux Training Course

Dr. Giovanni A. Orlando

Issue #71, March 2000

The president of Future Technologies describes his FTLinux Course training program for Linux.

Some time ago, I heard that success in mathematics means being able to formulate a good question and answer it. When I decided to write a book for Linux, the question I proposed was, “What is the best way to teach Linux to novice Linux users?” My answer was to write a course in Hyper Text Markup Language (HTML). HTML is cross-platform, so the course can be read on any PC: Windows, Macintosh, OS/2, BeOS, UNIX and Linux. HTML is the most-used language on the Web, where Linux users generally go to download software, browse and consult resources.

I also include the **man2html** program, so customers can:

- learn Linux in a nice environment, such as a web site
- browse Linux files on the Linux file system
- consult Linux man pages
- move quickly through the course
- go directly to the Web or FTP site to browse for information or download software

My intention for this course was to support as many languages as possible. I regularly speak Spanish, Italian and English, so it was natural for me to write the course in these languages. Demand for German Linux technology is high from a commercial and a technical point of view, and France moved to Linux some months after the rest of Europe, so I also include these languages. Our contract with ADC Japan (http://www.adc.co.jp/) allows us also to include Japanese. Future languages will probably be Portuguese, Turkish, Hebrew and Chinese, but this will depend on the market.

The Project

I drew on my twenty years in computer science and my various technical jobs with UNIX to write this course. I organized the project into the following sections:

- BASE: dedicated to new Linux users
- WebMaster: for HTML programmers, webmasters and web-server installers
- X Window: dedicated to X Window System users and programmers; also Tcl/Tk and Java X techniques
- System Integrator: how to integrate MS Windows, Macintosh and other operating systems in Linux networks
- System Administrator: the specific role of the system administrator in companies, includes in-depth shell programming
- Programming: for C/C++ programmers, also network programming techniques

The BASE course was originally written for Caldera OpenLinux in Italian. I started to write it for OpenLinux 1.0 in 1997, slowly. The course has been updated to 1.1, 1.2, 1.3, 2.2 and now 2.3. The first edition, FTLinuxCourse 1.0 for OpenLinux 1.3, was released in Italy for the Italian market in December 1998. In February 1999, we released the version for the U.S. market.

The Red Hat course was written directly in English, then translated into Italian. The latest version, 2.3, covers the latest Red Hat, SuSE and LinuxPPC distributions.

I have developed some useful shells and samples that are included in the FTContribs directory. I also contacted O'Reilly & Associates to include all the examples from the O'Reilly book series in the BASE CD-ROM. My friends at Xi Graphics gave me authorization to include the latest AX and LX demo.

In addition to the 40MB of coursework (excluding the course material in PostScript and Adobe PDF formats), there are 10MB of shells, TeX, LaTeX and PiCTeX examples available, our contributions in the FTContrib, plus more than 250MB of ExternalContribs, including examples from books (such as the O'Reilly series and others), articles from the Web, Netscape checkers, RFC and other software.

The product is continuously updated and includes Linux news, free software and more. The latest decision was to include the Linux OS itself.

The WebMaster course is new and important. It includes HTML programming and Internet web-server installation.

The X Window System was created in 1984. After several years of use on expensive UNIX workstations, Linux helped move X to the desktop. Today, for less than $80 US, it is possible to design and write X programs. Understanding and learning how to program X can be hard work. The X Window course covers many topics including C/C++, Tcl/Tk and Java programming, from the command line or by using builders such as **dtbuild** or KDevelop.

Why a Training Product for Linux?

Linux is free. People can download it off the Internet or install it from a CD-ROM. Each distribution includes a manual that explains Linux installation duplication. Users will then need a way to enter the Linux universe and learn the commands, how to search files, how to change the resource on an X application, how to create users, how to run a program in the background, etc. To supply tutorials for this method was the main reason I wrote the course.

Certification or Not?

This is another interesting question. The answer must be vendor-independent. Linux is not a single vendor, but comes from many working together. To be certified on Linux means to know Linux regardless of distribution. After learning the distributions, students can then study Networking (covered in the System Integrator course), X (covered in the X Window course), Programming (covered in the WebMaster: HTML Programming, X Window and Programming courses), System Administration (covered in the relative course) and so on.

Instead of becoming certified, the student must become a master or guru. The word “certified” means certified by someone or some company. A single, fixed choice in Linux is wrong. Linux is a successful operating system because it is horizontal; it is graphical, supports shell programming from the command line, supports binaries from other operating systems, includes games, is stable, is serious, is funny—it is everything. Having a certificate in Linux should mean knowing about all Linux topics and all Linux distributions, not just one.

Who Are the Linux Experts?

A Linux expert may be a young hacker or an old UNIX expert. The UNIX experts need to know how Linux is different from UNIX. For example, Linux kernel compilation is completely different from any previous UNIX models. The concept of kernel modules and the kernel organization is new, so the “old” UNIX experts will need to learn it. From the other side, both the great and young Linux experts should study the UNIX story. Linux history starts with the UNIX

story, not from Torvalds' first kernel release. The kernel was the conclusive link in a long chain including the GNU software, the X Window System, TeX and other freeware.

Going Further on Training

During the first months of 2000, FTLinuxCourse (also called Fast Training Linux Course) will reach version 3.0. In this version, we will expand the Linux Command Reference from 163 commands with examples to 1,000. We will also include a search tool and an analytical index. We will cover more on Tcl/Tk, Python, Perl and other programming tools.

At the moment, FTLinuxCourse includes the course on CD-ROM and a Linux CD-ROM. The GOLD edition comes with a course for Caldera, Red Hat and SuSE, along with their respective CD-ROMs. We are talking to an Italian publishing company about presenting FTLinuxCourse and to Sun Italy about including StarOffice. When this step is concluded, FTLinuxCourse users will have a complete course with more than 1,000 questions and answers, a Linux distribution and StarOffice. In February 2000, we will begin to develop a DVD version—a video course for Linux in all languages.

We are also working on a new project called “Linux Web Campus”. Like a university campus, we will recruit Linux experts as teachers and teach people from the Web. Linux enthusiasts can subscribe to a specific course from any city in the world and take classes on the Web. The campus is based on the original “Learning Networks” idea. In the beginning, we plan to use only HTML pages to teach Linux and test Linux knowledge with CGI exams. The final goal of LinuxWebCampus is to create Linux experts who can teach Linux to others, from the same web campus or elsewhere.

Another project is creating Linux Utilities (dedicated to Peter Norton), to simplify Linux tasks such as kernel recompilation. This project will also extend Linux from a UNIX point of view. These utilities will be offered for download from the LinuxUtilities.org web site or on a CD-ROM at low cost.

The last project involves teaching customers how to build their own Linux distribution. This is a very futuristic project. We will develop utilities to locate, test and install remote Linux programs. We will show how to create a minimal Linux distribution including the Caldera LIZARD (available for free at OpenLinux.org), the RPM from Red Hat, KDE, our LinuxUtilities and other FT applications.

Resources

Acknowledgements

Dr. Orlando is the author of FTLinuxCourse and the president of Future Technologies, Italy. His degree is in mathematics. He founded Future Technologies in 1994, in Pordenone, near Venice. Future Technologies started as a consulting company for UNIX support, programming and training. He can be reached at gorlando@futuretg.com.

Archive Index Issue Table of Contents

Advanced search

Technology Training: Trends for the 21st Century

Brian Holt

Issue #71, March 2000

A look at how training courses will be taught in the future.

One hundred million. That's the number of Americans expected to be involved in adult or continuing education by the year 2004. Many of those individuals will be technology users who need to be trained on evolving software packages and hardware systems. A 1998 study published by International Data Corporation's Information Technology (IT) Training and Education Service projected that the worldwide IT training and education market will experience a growth rate of over 11%, surpassing $28.3 billion by 2002.

Whether these individuals are CEOs, managers, supervisors or general staff, all need to stay current in order to perform their responsibilities with maximum effectiveness and efficiency. To meet the needs of an expanding marketplace, leaders in IT training will strive to stay in touch with a basic law of teaching—take complex material and make it easy to understand. To accomplish this goal, the computer training industry needs to be aware of four escalating trends. Some of them are here already; some of them will loom larger in the coming decade. The key, however, is to recognize these trends as the foundation of standards for a rapidly changing industry.

Individualized Learning

“The initiative has shifted from the corporate structure to the student, from the trainer to the trainee,” says Paul Swanson, president of Oasys, a corporate training firm in Toronto, Canada. “We're talking about individualized learning for individualized applications.” The traditional classroom setting is more expensive and less effective than self-paced training videos, CDs or intranet courses. To meet the growing demand, developers of software training packages such as Keystone Learning Systems provide convenient, cost-effective courses on video and CD-ROM which trainees can use at home or in the office.

There is still no substitute for excellence in instructional materials, nor will there be. Red Hat 6.0, an eight-part series of training videos, was especially helpful for Liz Peterson, owner of Depco Business Services in Oklahoma City. Peterson works with trainees in units of about 50 platforms each, where interactivity is essential for smooth-running sessions in a stable learning environment. Self-paced training allows the student to go as quickly or slowly through a course as the need arises. Hands-on “see it, hear it and do it” training solutions are the wave of the future. Peterson is an avid supporter of video-based training, especially her Keystone program, because the videos hold the attention of the trainees. She reports, “Ordinarily, this stuff is dry, yet the Keystone videos make it interesting.”

The Move to the Web

The legend of the Internet is growing. On-line training will escalate in the next decade. Not that there aren't downsides here; there are. Most suppliers of training materials for hardware and software applications are moving aggressively onto the Web, trying to connect training to the world of cyberspace. The development of speedy, effective and secure Internet applications is becoming a watchword across the whole world of IT. Office applications, networking, graphics/design, finance and operating systems will be affected as companies and independent users shift en masse to the World Wide Web. Trainers and their suppliers will have to respond.

Interactivity vs. the Hierarchical Model

The computer industry practically invented the term “hands-on”. This trend will continue. The training modules that will be in demand for effective lab work will continue to emphasize realistic configuration and administration tasks where, in effect, the learner is in the driver's seat. “The reduction in instructional time when compared to the traditional instructor-led approach is typically in the range of 20-75%,” wrote Verl E. Dennis in the *Journal of Instruction Delivery Systems* (Winter, 1994). He added: “Interactive instruction offers a solution for minimizing training time without sacrificing desired training outcomes.”

The benefits are obvious:

- no waiting for scheduled training
- flexibility of access, no waiting to get to “relevant” bits
- self-paced curricula, not at the mercy of the slowest or fastest learner
- time efficiency—program stops when the trainee has mastered the skill
- immediate feedback on mistakes
- visual and audio aspects work together to reinforce each other

Multimedia, Voice Activation and Simulations

These three terms form a cutting-edge trend all by themselves. In IT, there is one thing that trainers and executives can reasonably expect—the unexpected. Software marketers and producers are already acting on this notion. Voice activation and recognition will begin to play a larger role in how we compute. There is also a trend in the desire for simulations. This cutting-edge technology allows the learner to “try the job” prior to actually “doing the job”.

It's a matter of thinking ahead. Red Hat, Inc., which has been building the enterprise credibility of Linux for several years now, tells would-be trainers and trainees that “staying current with technology, best practices and user communities are all essential.”

Not so long ago, Price-Waterhouse introduced a CBT (computer-based training) Multimedia program as a prerequisite for a week-long classroom course. The program reduced by 50% the time needed for users to get up to speed. The cost per learner was $760 for traditional instructor-led training, versus $106 per learner for multimedia training. Price-Waterhouse estimated that sustained use of such multimedia instruction could amount to a savings of some $10 million US.

Top Trainers

Content is important with any course, but high-quality, competent instructors are also a must. You don't send a boy to do a man's job, and you don't send an actor to represent the challenging world of network training. Along with experts in knowledge, you need expert teachers. It helps if trainers are tops in their field.

Paul D. Sheriff, a past president of the Orange County Visual Basic User Group, has more than 14 years of experience in programming business applications. “I like to teach people real-world concepts, not just how to use the syntax of the language,” he says. “I stress good programming standards, how to approach problems, and give real examples that I take right from my consulting business.”

Dr. Hany Greiss works as a Senior Technical Instructor for PEAC, Inc., in Ottawa, Ontario. His teaching philosophy shows why he is a sought-after teacher and trainer. “I say to all those I teach, do not become encyclopedias, but rather understand concepts, and when you're on the job, you can look things up.” In hiring trainers, he says to look for people with a passion for their product, people with an eye for conveying the big picture. Small details can always be tended to in the field.

Self-Pacing a Plus

Everyone likes to learn at his or her own speed. Taking time away from an already-crammed schedule to attend a one-week seminar in a far-flung city requires lots of time and manpower. Here is where video technology comes to the rescue. For organizations that prefer quality training at a fraction of the cost of traditional classroom and web-based training or distance-learning applications, videos and CD-ROMS are an effective and creative alternative. Students are able to start, view, stop, review and start again at their own speed. Even better, there's no one in the front row dominating the lecture or asking all the questions.

Individualized learning means the freedom, if not the necessity, of learning when and where you want. “On-line users and students need specific, detailed and often repetitive instruction,” says Paul Swanson. One of the most valuable advantages of self-paced training over a classroom setting is that while the classroom instruction will be over in an hour or two, the videos will still be around. The material is always there if the trainee forgets a concept or application.

The Certification Equation

A 1998 study by *PC Computing* Business Labs found that training increases employee performance by nearly 20% and pays for itself within weeks. The Gartner Group of Stamford, Connecticut, surveyed 250 enterprise managers in 1997 and found that 93% of companies surveyed pay for certification and 89% pay for training. Alert executives are closely scrutinizing trends in certification and testing.

It's good to be realistic. Training, too, is affected by the old law of diminishing returns. Red Hat's web site puts it this way: “Training alone is not enough to become a competent user, operator, system administrator or engineer. Good training can provide a foundation; the participant must do the rest.”

Realistic Training

Some challenges of the 20th century will remain to test 21st-century trainers. Gwen Wakal, president of the Edmonton-based Active Computer Service, a pioneer in computer training in Canada, says: “It's still not as easy as it seems out there for one network to talk to another, even with Intranet and CD-ROM going for you.” Training for bridge-building platforms and cross-supporting technologies will be called for more and more as software packages try to find the middle of the road between greater complexity and a call for standardization. This means that at least as far into the future as we can see, affordable, high-quality, self-paced computer training products and trainees will

be a mainstay as IT enters the new millennium. “On a scale of 1 to 10, I rate training as a 10,” says Liz Peterson. “Training will always be necessary,” concludes Wakal. “Make sure you get the best you can.”

The age of the laptop learner and the cyber-student is here. As the 1990s dawned, America's Secretary of Education warned that traditional learning methods were in a time warp when measured against the mushrooming explosion in technology. This warning has specific application for the technology training industry. Too often, the chalkboard or out-of-town seminars are still the norm. Yet in this fast-moving world of intranet and Internet education, it is vital to sort out the glitz from the gold. What's needed is a holistic and realistic look at the trends affecting technology training today and in the future.

Brian Holt is Marketing Director of Keystone Learning Systems.

Archive Index Issue Table of Contents

Advanced search

Interview: Jim Higgins, Caldera Training

Marjorie Richardson

Issue #71, March 2000

What's happening in the Education Department at Caldera Systems? Read on...

In 1985, Jim Higgins joined the Novell Education department as an instructor/course developer and, except for a three-year period as an independent CNI, continued with Novell until 1998. In April 1999, he accepted an invitation to join the Caldera Systems team to further develop their Education Department. I "talked" to him by e-mail on December 23, 1999.

Marjorie: Training seems to be a popular issue in the Linux world today, and no one can doubt its importance. What motivated Caldera to begin giving Linux training?

Jim: We have always focused on "Linux for Business" (now eBusiness), and as we used an indirect distribution model, we had channel partners who needed training on how to use Linux and make money with Linux. We started in September 1998 and had Ford, Boeing, MCI and others who ended up becoming direct end users as well as partners. Business has been booming since.

Marjorie: That's good to hear. How did you go about designing your classes? Deciding what to offer, what to emphasize, etc.

Jim: Both of the first two courses were designed as an introduction to many of the solutions Linux provides. At the very start, Caldera decided to support an independent certification effort instead of creating our own proprietary version of certification, so we helped the key proponents of Linux Professional Institute (LPI) get together and form an organization. With this decision made and the process put in motion, we needed to provide training that would prepare students for the certification examinations. We focused first on teaching about the objectives to be tested by LPI, and then added appropriate materials to round out the courseware and make it instructionally sound. We ended up with

three one-week certification preparation courses, and then many additional courses to take care of advanced and specialty issues.

We also focused on the needs our partners had for integration into existing environments, and we ultimately set product training and technical training high on our priority list.

Marjorie: When did Caldera first start offering Linux training classes?

Jim: The first class was taught as a beta in September of 1998.

Marjorie: That was an early start—you must have been one of the first to offer it. How many classes do you offer? Subjects?

Jim: Education is an ever-evolving product and process. By the end of 1999, we will have ten courses completed out of more than 15 currently planned. The completed courses include both of our week-long overview courses entitled:

- Linux Administration for UNIX Professionals
- Linux Administration for Network Professionals

Neither of these courses is designed to prepare a student for certification. The first two of the three one-week-long certification preparation courses are also complete:

- Linux System Administration
- Linux Network Administration

The third certification preparation course, Linux Enterprise Administration, will be complete by mid-January 2000.

Two elective courses are also complete:

- The very complete and in-depth five day course entitled Linux TCP/IP Internetworking, for those wanting to know a lot more than the basics about TCP/IP.
- Since Linux literally runs on shell scripts, a thorough approach to Linux Shell Scripting makes this four-day course essential for complete integration and management.

Four additional one-day specialty courses designed primarily for VARs and system integrators will also be complete. They include the following:

- Linux TCP/IP Essentials
- Linux Scripting Essentials

- Linux and Windows with Samba
- Linux Solutions for eBusiness

Several additional solutions-based courses are being defined and developed.

Marjorie: Sounds like you have a wide selection of all the essential courses. How often do you give classes? Can schedules be found on your web site?

Jim: Classes are taught nearly every week at our Authorized Linux Education Centers (ALEC) in various locations around the world. In addition, we conduct "Train the Trainer" courses for our Authorized Linux Instructors (ALIs) at our Lindon, Utah education center. These instructors then go out to teach even more classes at their own centers, or under contract for customized training situations. Class schedules for most centers can be accessed through our education web pages at http://www.linuxeducation.com/.

Marjorie: I think on-line training is the wave of the future. Do you offer any sort of training on-line? If not, do you think you will in the future?

Jim: A wide range of alternative learning methods is a part of our overall education strategy, with some of them already in development. On-line subscription services, as well as free access to certain pieces of educational materials, are definitely part of our plans. Beside providing the largest number of Linux training centers worldwide, Caldera Systems recognizes the various needs and training styles required by individuals and is developing a full range of options to help meet those needs. No matter where someone is located or how they like to learn, we feel Caldera Systems is the full-service solutions provider of choice.

Marjorie: What do you consider the most important class you offer? That is, if a person could take only one class, which one should it be?

Jim: Some may consider this a political sidestep, but there is no one way to answer this question. What someone wants and needs to do with the information they learn in class will determine the type and length of class they attend. For example, if you are starting from scratch and your main goal is to become LPIC (Linux Professional Institute Certified) so you can assure others of your abilities, you would take the full series of three certification courses:

- Linux System Administration
- Linux Network Administration
- Linux Enterprise Administration

On the other hand, if you're already well-versed in UNIX or have used Linux in a minor role, you can usually learn what's needed by attending only the second and third certification preparation courses. If you want just an introduction to Linux and why it should be considered for either a part of your overall network or how it can provide the best solutions for your clients, then the Linux Solutions for eBusiness class is the best to attend.

The options continue to depend on your experience and your purposes for learning. You can read a description of each course by visiting http://www.calderasystems.com/partners/alec/courses/.

Marjorie: Well, that certainly makes good sense—sidestep or not. Do classes concentrate on Caldera OpenLinux or do they cover most (major) distributions?

Jim: We provide distribution-neutral courseware for certification preparation training. Actually, there are two questions in one here. In essence, you're asking if we teach on our own platform, “OpenLinux”. The answer is, “Yes, of course we do.” Do we teach about, explain, demonstrate or use any other distributions in our classrooms? The answer again is, “Yes, of course we do.” While Linux is Linux, there are important differences among OpenLinux, TurboLinux and all the rest. System and network administrators are kidding themselves if they think they'll experience only one distribution of Linux. Just like Windows and NetWare, UNIX and Macintoshes co-exist on the same networks, so do various distributions of Linux on the same networks.

Because of this, Caldera Authorized Curriculum provides experiential learning on multiple Linux distributions. As an example, in the Linux Network Administration course, students actually install up to six different Linux distributions as part of the lab exercises. Beside installing them, they must make each distribution run different services such as DHCP, Sendmail, Apache, etc.

Then, in the Linux Enterprise Administration course, that concept is expanded and multiple LANs are created using different Linux distributions. All of the different LANs must then be made to communicate with each other and share those services across a WAN.

As students progress to some of our specialty courses, the classes focus only on Caldera Systems' OpenLinux. This will be true for such courses as those specific to our new eServer product. There is no need to teach about other distributions in situations like this.

Marjorie: Red Hat offers Red Hat Linux training and certifies its students as Red Hat Linux-certified. Does the successful student receive some sort of certificate? What does it say?

Jim: Caldera Systems does support and use certification. This is where LPI and Caldera Systems' ALECs come together. As I said before, multiple distributions will coexist on LANs and WANs around the world. Most of this will take place over the Internet. For all those who find themselves dealing with situations like this, they need a broad coverage with in-depth knowledge and experience about multiple Linux distributions.

LPI certification (LPIC) provides that “proof” of knowledge and experience. The bar for passing has been set very high. This is intentional so that the Linux industry doesn't have to go through the growing pains experienced by Novell and Microsoft when the “paper CNE” syndrome threatened to destroy the value of certification.

Obtaining LPIC status says, “I know my stuff and I've proven it. If you want an employee who can actually do what needs to be done, I'm it. You don't have to pay me for several months while I experiment with theory I learned in class. I've already done it and can do it again, *now*!”

That's a powerful statement to an IT manager. They can no longer afford the time and money to train someone new. In today's computer environment, LPIC will mean more than any certification available to date, and Caldera Systems proudly supports LPI.

Additionally, ALIs (our own Authorized Linux Instructors) must pass another exam specifically targeted to help us identify those good enough to represent Caldera Systems Education and to teach our clients.

Marjorie: I'd say the cooperation between LPI and Caldera is a win/win situation for everyone. There's a book on Red Hat Linux certification and what you need to study to pass (author Kara Pritchard). Is there a comparable book for your training courses? If not, is one planned in the future?

Jim: Specifically, we do not provide exam cram books to prepare for passing exams. Caldera Systems has not sponsored any such book, but numerous other books from Caldera Press about OpenLinux provide necessary background and detail information to help students and existing administrators. These books are often used as supporting references for those studying on their own. Our own course manuals have been developed from scratch to thoroughly and intelligently provide the best in-class text and still provide an excellent reference for use after class.

While there can be great value in texts such as the one you mentioned, when someone bases their exam-passing ability on such material alone, they are cheating themselves out of the hands-on experience absolutely essential for businesses to continue to be successful today. As easy as Linux can be for most users, in the background is the same depth and complexity existing in every other operating system. And that depth and complexity is what makes Linux so powerful and reliable at the same time.

Our classes are unique and well-sought-after because we spend considerable time with hands-on, practical, solutions-based exercises that books alone cannot duplicate.

Marjorie: Do you offer different levels of classes, i.e., beginner, intermediate, advanced?

Jim: Yes, we do. For certification purposes, there are three different classes, each building on the experience of the one before. If you know Linux somewhat in general, but need to learn more details on Linux shell scripting or TCP/IP, then you can take the longer, more detailed specialty courses to provide the expertise you need. On the other hand, if you need to learn only a little about how scripting works, the one-day course provides a fast, solutions-based experience that you can take back and use immediately.

Marjorie: What are your requirements for entry? Can they be rank beginners who don't know what a keyboard is or do you expect computer literacy?

Jim: There are many qualified, high-quality training companies where individuals can be taught about computers and applications in general. We do not offer classes at that level. However, many of our ALECs offer those classes. Our focus is squarely on making Linux and its associated uses an integral part of a LAN and WAN administrator's expertise. Again, students leave our classes actually knowing how to accomplish the tasks given them.

Marjorie: What are the costs for taking your training classes? Do you have any sort of scholarship program for those who can't afford the entry fees?

Jim: The SRP (suggested retail price) for our five-day courses is $1995 US. Each ALEC has the right to offer our courses at any price they choose. While most centers charge the SRP, they do have promotional ideas from Caldera Systems, and they can implement their own whenever they have reason. Scholarships have not yet been provided by Caldera Systems. There are suggested discounts for students signing up for multiple classes. There are often discounts for companies sending multiple employees registering concurrently. These options are entirely up to the individually owned and operated ALECs.

Marjorie: Anything you would like to add in closing?

Jim: By using my experience to eliminate the mistakes made by other companies entering the technical education market, we can provide even more of the potential benefits from the very start of our program. In designing an entire program with multiple offerings before we ever started pushing our classes and educational products, I believe we're ready to roll out the very best program with the most complete offerings ever available.

Students, no matter which path they take to learning from Caldera Systems, will be better prepared to advance in industry than they would by following any other path. I welcome everyone interested in learning why Linux Education from Caldera Systems and Linux certification through LPI are the two best moves they can make in their lives right now.

Marjorie: I'm sure many will take you up on that. Thank you very much for your time.

Archive Index Issue Table of Contents

Advanced search

Instructor-Led Training

Marty Ferguson

Issue #71, March 2000

Training is a sound investment networked to future growth.

I was recently talking with a close friend who works as a systems software developer for a small startup technology company. The architects are planning a transition to Linux servers as their platform, but the company does not want to provide Linux training for their staff. When I asked why, he replied, “It's too expensive.” Even though the OS license is free, the documentation is free, and the source code is free, this company was overlooking an important factor: the costs of inadequate training can far exceed the costs of educating staff.

Networked systems in the workplace continue to grow and expand. As an example, the current growth rate of the Internet is a factor of two every eight months. Linux is considered by many to be the most powerful and versatile network operating system to arrive on the scene in recent years. Due to its power and flexibility, its rate of deployment is also growing rapidly. More and more organizations are turning to Linux as a low-cost, reliable and increasingly supported network-computing platform for a variety of business and Internet applications.

As businesses strive to keep up with this fast-paced network development, their employees need to experience a parallel growth in skills for optimum job performance. Unfortunately, it's common for organizations to bring in training providers only after the need becomes evident—almost as an afterthought. This makeshift method of training prevents adequate planning for anticipated time and financial investments. And “reactive” vs. “proactive” training programs are a sure-fire way to knock off any company's competitive edge.

A Linux training solution that is designed to grow along with an organization brings benefits that can permeate operations across the board. Sufficient education ensures that senior administrators are not overwhelmed with on-the-job training, encourages employees to stay on for the long term, reduces

the likelihood of emergencies and strengthens employees' ability to deal with a crisis.

Reviewing the Basics of Linux Training

Employees who can benefit from Linux training range from NT administrators with little or no Linux command-line experience to experienced UNIX systems administrators with a broad background in heterogeneous environments.

When introducing or expanding the use of Linux into an organization, the phases of deployment must be dovetailed with a training plan. Clearly, the first training phase should be aimed at decision makers and system administrators. Key administrators and IT managers often attend training sessions together, working and learning as a team. The next phase involves a pilot program, with Linux installed in isolated situations. As the Linux system is implemented throughout the organization, the scope of training broadens to include system developers, content and media professionals and other users. Finally, ongoing training is used to refresh knowledge and ramp up employees new to their organization.

Every Linux distribution is based on the GNU (open-source software that functions as UNIX but is not UNIX) tools and shells, and the installed systems are frequently referred to as GNU/Linux platforms. Since the power and flexibility afforded by Linux begins at the command line, the first level of training should begin with the Bourne Again Shell (bash) along with GNU utilities. The second level of training should address installation of a GNU/Linux distribution, including the best techniques for planning and executing the installation. The third training level should instruct students in the configuration and maintenance of freely distributed software services and daemons running under Linux, along with security and lock-down practices.

Preparing for the Training Process

During the training cycle, system administrators will be unavailable for their regular duties. Larger organizations often arrange for on-site delivery of the courses. On-site training, where education providers bring the “classroom” directly to a company location, can reduce expenses such as travel costs and also reduce employees' training time obligations.

Providing sufficient operations support for available staff is critical to a successful training plan. Whether courses are held on or off site, students must be relieved of work responsibilities (or on-call status). If a student remains in the classroom while attempting to provide ad-hoc hotline support for an operational crisis, it's a lose/lose situation. The student is unable to focus sufficiently on both the training and the crisis support—the value of training is

lost, and the quality of support is marginal. Students must mentally and physically detach to immerse themselves in classroom experience to maximize the return on the educational investment.

Reaping the Benefits of Linux Training

The payback for instituting a well-planned instructor-led Linux training program is both progressive and far-reaching. The following are areas where an organization can expect to find substantial benefits as a direct result of successful Linux training.

Acquiring Linux Fluency

The GNU/Linux culture is inherited from the various flavors of UNIX. The evolution of Linux that occurred during the 1990s mirrors that of UNIX during the 1970s and 80s. Due to the nature of the GNU General Public License, the open-software culture and peer-review process have created an evolution that provides universal accessibility to computer professionals and power users alike.

The legacy culture of Linux makes an understanding of the roots of UNIX and the nature of free software critical. Gaining Linux fluency allows new users and administrators to strengthen their abilities to communicate with seasoned veterans of the Linux environment. Learning about basic Linux resources, the structure of a typical distribution, the wealth of contributed documentation available and the philosophies of the system's design all work to jump-start Linux fluency.

Easing the Way for Linux Administrators

In larger organizations, senior administrators often provide on-the-job training for their staff. But the senior administrators' most critical role lies in planning for growth and evolution of the network of systems for which they're responsible.

When Linux instruction is provided in a classroom setting, students can dedicate time to intensive, hands-on training that achieves a well-rounded knowledge base faster and more effectively. Additional one-on-one training and mentoring—aimed specifically at the needs of the organization's workplace—is a better investment of senior administrators' time.

Learning That You Don't Know What You Don't Know

For both users and administrators, the scope of the tools and facilities available under GNU/Linux systems is enormous. When approached with “how do I...”

questions, veteran administrators often respond with “Well, that depends.” With a GNU/Linux-based training solution, a wide arsenal of methods can be taught to solve potential or actual problems in many different ways.

A holistic classroom approach, which includes lectures, labs and a hands-on interactive learning environment, allows students to be rapidly exposed to methods, tools and techniques they otherwise may not discover for months. It is not uncommon to hear a student exclaim, “I never really understood how that worked!” or “What I just learned in that last session alone made it worth being here this week!”

The instructor-led setting allows students to pose questions and explore many aspects of conventional approaches to Linux-based solutions. Students interact with one another and discover how others have addressed similar configuration and maintenance issues in the past. This “shared” learning environment enables students to benefit from the experiences and discoveries of fellow learners and instructors.

Linux training programs should provide each student with access to a Linux system during hands-on labs. The labs speed the process from learning the basics to putting the knowledge to work, and support different learning styles among the students.

Managing Employee Retention and Growth

To avoid being overwhelmed by the rapid growth in today's networking systems, Linux users and administrators need to receive training that enables them to stay ahead of the curve. The combination of accelerated technologies and inadequate training is a setup for failure; staff members become frustrated and often begin to look at new opportunities for professional growth.

Furthermore, training is often perceived by employees to be a reward or a job perk. By providing a proactive training and development program, employers demonstrate that they value their employees and both parties feel there is something at stake in the continued employee-employer relationship.

Regarding training, there is a paradox that some organizations are afraid of: if they offer top-notch training, employees may eventually take their enhanced skills elsewhere. In fact, employees are far more likely to remain in an environment where they can grow their skills and develop their careers. Since employee recruitment has become so competitive, many employers seek opportunities to train and promote from within. Employees have ample prospects for advancement, while employers fill staffing needs without paying relocation fees, signing bonuses and other recruitment expenses.

Achieving Diversity through Cross-Training

Providing cross-training for employees who maintain a specific job focus reduces operation risks by enabling staff members to deploy contingency plans smoothly and rapidly. For example, an administrator who maintains an NT or Novell server can be cross-trained in Linux. When the inevitable human resources crunch occurs, the workload can be easily redistributed. If a staff member is out for a work emergency or due to a personal crisis, another can step in and help fill the role. Diagnosing a problem over a long-distance connection to someone who doesn't know the first thing about the bash interactive shell can be a frustrating experience at best and can lead to potentially greater catastrophes than the initial problem.

Distributing knowledge and skills to employees through cross-training also provides staff with a better understanding of multiple problem and solution domains available in their networked computing environment. By providing cross training of Linux skills to staff members who are responsible for other systems (such as NT or Novell), the collective of knowledge can assure the very best overall practices are employed. By the same token, appropriate NT systems training for staff members who are primarily focused on Linux will lend them understanding of how best to leverage the strengths and weaknesses of each operating system environment.

Increasing System Security and Reducing Risks

In today's high-speed networked computing environment, both inadvertent damage and malicious attacks can cripple a system in the blink of an eye. Damage recovery is a painfully slow and meticulous process. Data is almost never completely restored, and the time devoted to system recovery has net-zero productivity.

With proper training and planning, however, preventive measures can be taken to avoid risks and speed recovery in event of a breach. Knowledge of how best to protect the system through management of permissions and system services allows the administrator to maintain tight system control, while determining allowable access. Initial planning for system installation will accommodate the most-favored backup strategies within the organization.

Considering the Costs of Inadequate Training

The costs associated with Linux-based training are easily determined. It's far more difficult to ascertain the tangible and intangible costs of not investing in training. The backlash of not instituting an educational program can include reduced staff productivity and the loss of potential employee innovation and creativity. And the company's customers are another critical factor; clients

depend on a networking organization for high-end technical knowledge and service. It just takes one slip to lose credibility, and perhaps business as well.

Finally, lack of adequate training could pave the way to security risks. Linux has become a big player on-line, and its growing popularity has led to an ever-expanding threat of potential damage from hackers. With sufficient knowledge of the Linux operating system, employees know which features or unused services to disable to ensure site protection.

The costs of inadequate training can be either obvious or subtle, but they trickle down through most phases of operations. Solid training allows a diverse staff to play from the same sheet of music, and the advantages are equal for both direct employees and paid consultants. In the final analysis, providing an ongoing education program for employees is the most important networking an organization can do.

Marty Ferguson is a Linux instructor with Global Knowledge Network, a worldwide IT training provider headquartered in Burlington, Mass. He has consulted with numerous Fortune 500 companies regarding Linux training and is a co-founder of APEX Integration Systems, a North Carolina-based company that builds turn-key Linux systems to meet specific customer applications. He can be reached at marty.ferguson@pobox.com.

Archive Index Issue Table of Contents

Advanced search

Simplified Wrapper and Interface Generator

Wael Hassan

Issue #71, March 2000

An introduction to the uses and advantages of SWIG.

SWIG (simplified wrapper and interface generator) is a software development tool that connects programs written in C, C++ and Objective-C with a variety of high-level programming languages. It is often used with common scripting languages such as Perl, Python and Tcl/Tk. In addition, it has been extended to include languages such as Java, Eiffel and Guile.

SWIG is used to create high-level interpreted programming environments for systems integration, and as a tool for building user interfaces and testing. It is distributed as open source and can be downloaded from http://www.swig.org/.

In the following sections, I will discuss some of SWIG's features and my personal experience with it as a testing engineer at Zero Knowledge Systems (http://www.zeroknowledge.com/) for the Freedom Software.

Environment and Mission

At work, I have to test e-mail, web, IRC, FTP, proxy servers and TELNET clients. My machine is a Pentium powered by a 266MHz processor running Red Hat Linux with 128MB of RAM. My responsibilities include testing the source code currently being developed—hundreds of thousands of lines of code in a mission-critical system, with no room for errors. Given that "the strength of a security system is the strength of its weakest link", there was no place for flaws. The code cannot be tested manually, because of the distributed architecture of client and servers. Thus, there was an urgent need for an efficient tool that would automate testing procedures. This tool had to be platform-independent and compliant with both C and C++.

Then There was Light!

Shopping around, I learned about SWIG from its web page. The source can be easily compiled for Linux; it is about 2MB in size. SWIG is multi-platform, i.e., there is no need to duplicate test procedures for Linux and Windows; it supports C and C++; and it can be integrated into MSVC++ (Microsoft Visual C++). SWIG proved to be the perfect tool.

Let Us SWIG Together

SWIG accepts as input an ANSI C-like interface file that describes the functions and objects constituting the program to be tested. The interface file can also include SWIG directives and documentation. SWIG wraps the functions in another C program. When both of these programs (the source code and the wrapped source code) get compiled, SWIG creates a library file that can be called from the Tcl shell.

Step By Step

- The Program: start by writing your C program to be tested. One thing to note is you have to modify the name of the "main" function. Listing 1 is an example of a C program.

Listing 1

- Interface file: in order to allow SWIG control over this program, we have to write an "interface file". An interface file for our C functions might look like the one in Listing 2.

Listing 2

- Build a Tcl module: at the prompt, type the following:

```
swig -tcl8 my_interface.i
```

This command will create a Tcl 8.0-compliant library.

- Compile wrappers for Tcl using the commands

```
gcc -fpic -c example.c example_wrap.c\
   -I/usr/local/include
gcc -shared example.o example_wrap.o\
   -o example.so
```

- Call the Tcl shell by typing **tclsh**.
- Load the example.so library with the command
- **load ./example.so example**

Now, feel free to call the functions implemented in the C program:

```
get_time
Sun Feb 11 23:01:07 1996
```

Taking Care of Business

SWIG helped me a lot, due to the flexibility of function calling it provides. The company had a secure mail system to be tested. In this system, all e-mail messages go through several servers before they reach their final destination, and they are encrypted each time they pass through a new routing server.

My approach toward testing this environment was to write an e-mail generator program in C which I called GenerateMail. GenerateMail accepts several options such as the number of To, CC and Bcc copies, the number of file attachments, etc. It produces a file ready to be piped to Sendmail.

A typical GenerateMail run would be something like:

```
tclsh generate_mail -Attachments 3 -CC 2\
   -output file msg.txt
tclsh send_mail msg.txt
```

The first line creates an e-mail message file. The message has three target addresses and two carbon copies. Three binary files were attached as MIME attachments. By default, GenerateMail uses bitmaps that are in its current directory.

The second line calls Sendmail with the appropriate options to accept that mail message and send it on to the wire. Doing that, it was easy to generate a large number of mail messages. In addition, comparing the source and destination message checksums was very easy with the help of SWIG.

SWIG Advantages

SWIG's advantages are evident in a number of ways. It adds flexibility to testing procedures, it can be used for prototyping and helps in system integration. Very few modifications of the code are needed. Moreover, SWIG directives can be easily extended.

On the other hand, the wrapper has some limitations when it comes to C and C++ programs, because SWIG's author did not mean it to be a fully blown parser.

The following are some examples of unacceptable input.

- Functions with variable-length parameters will not work.
- Function pointers and array declarations are problematic; they could be hidden from the interface file by using typdef.
- Most features of C++ such as templates and operator overloading are not supported.

Nevertheless, SWIG is still a powerful tool, especially since it can be integrated with Microsoft Visual C++.

The End

SWIG is a powerful tool that runs on multiple platforms and supports multiple languages. It served my goal of testing the system within the specified time constraints. It is a neat tool that still has much room for expansion. If you are in the business of testing, prototyping SWIG is the way to go.

Wael Hassan is a graduate student in computer science at Concordia University. He is studying for his master's degree and has interests in Linux, cryptography, CORBA and e-commerce. Wael can be reached by e-mail at wael@acm.org.

Archive Index Issue Table of Contents

Advanced search

Using Apache Proxy to Suppress Banner Ads

Raj Mathur

Issue #71, March 2000

If you find Internet advertising annoying, time wasting and resource-consuming, get rid of it.

No matter how much you have, Internet bandwidth never seems to be enough. At home, I dial up on an oldish analog exchange to the Internet, rarely getting connections better than 4800 bps; at the office, I have a (shared) 64KB leased line to our U.S. office and thence to the Internet. However, no matter where I surf (it's part of my job), I have always wished there were a way to avoid waiting for pages to display while loading banner ad graphics from the remote server.

I am running an old, completely hacked version of Red Hat (2.0 to be precise) with kernel 2.2.10 on my 66MHz 486 PC at home. I use this system for browsing (mainly using Lynx), software development and fooling around with Linux. My wife and two children also use it, primarily for browsing and e-mail using one of the web-based e-mail services with Netscape Communicator.

Since I first installed Linux on this system a few years ago, I've been running the Apache web server to serve local content. More recently, I've also enabled Apache's caching Proxy module to maintain a local cache of commonly accessed documents, speeding up Internet access somewhat. The version of Apache I'm running is 1.3.3 (yes, I should upgrade to the latest and greatest, but there's no pressing hurry).

The Apache web server (http://www.apache.org/) is a well-known HTTP server and is part of most, if not all, Linux distributions. Along with being the world's most popular web server, running on over 56% of web sites (Netcraft survey at http://www.netcraft.com/), Apache also has features which helped me replace graphics from well-known banner ad sites with an innocuous local graphic.

Since the distribution I'm running is so old, I will describe the methodology of replicating this on a more modern distribution, namely Red Hat 6.0. Most other popular distributions of Linux will require the same or similar steps.

Background

In order to use Apache for this purpose, I had to set it up as an HTTP Proxy, using **mod_proxy**, and enable and use the URL rewriting engine built into Apache (**mod_rewrite**).

In a nutshell, you must first set up Apache to act as a proxy server for the browsers on your local network (or for a single system, which is what I do at home) and tell the client browsers to use that Apache host as a proxy. Then use the powerful mod_rewrite URL rewriting engine to search for banner ad HTTP requests (these match the URLs of well-known banner ad server hosts) and substitute a local graphic for them. With this done, you can sit back and watch your web pages flow into your browser, while that irritating advertising is automatically replaced with a more soothing and altogether more relevant graphic.

You will need to be root while following the steps discussed here.

Enabling Proxy

The first step is loading and enabling the proxy module (mod_proxy) which is part of Apache. In most distributions, you enable mod_proxy by appending the following line to your Apache server's configuration file (/etc/httpd/conf/httpd.conf or /usr/local/apache/etc/httpd.conf):

```
ProxyRequests on
```

To check if the module was loaded okay, restart the web server with the command:

```
killall -1 httpd
```

Check the last few lines of the Apache error log (usually in **../logs/error_log** or **../var/log/error_log** relative to the **httpd.conf** file), and if there are no errors, celebrate! Apache's proxy feature is enabled. This has been tested on Red Hat Linux 6.0 and Debian's “Potato” GNU/Linux 2.2, and it should work with SuSE Linux 6.1.

If you get an error message like:

```
Invalid command 'ProxyRequests', perhaps misspelled or defined
by a module not included in the server configuration
```

then you need to either locate the Apache Proxy module, or download and install a version of Apache which has mod_proxy available. Installation is outside the scope of this article. See the Apache HOWTO at www.apache.org/docs-1.2/mod/mod_proxy.html.

Once you have the Apache proxy configured and running, you can tell your client browsers to use it. In Netscape Communicator, select Edit/Preferences/Advanced/Proxies and “Manual Proxy Configuration”. Click on the “View” button, and enter the name of the host running Apache in the text entry boxes for “FTP Proxy:” and “HTTP Proxy:”, using 80 as the port number for each.

For Lynx, edit the global Lynx configuration file found by default in /etc/lynx.cfg or /usr/local/lib/lynx.cfg and change these lines

```
#http_proxy:http://some.server.dom:port/
#http://some.server.dom:port/
#ftp_proxy:http://some.server.dom:port/
```

to look like:

```
http_proxy:http://apache.my.dom:80/
http://apache.my.dom:80/
ftp_proxy:http://apache.my.dom:80/
```

Then, replace apache.my.dom with the name of the host running the Apache server. Similar techniques will apply to other web browsers—consult the browser's manual for details.

Configuring the Rewrite Engine

Now that you have the Apache Proxy configured and the browsers set up to use it, you can get to the heart of the matter, which is redirecting banner ads to a local graphic, saving yourself bandwidth, time and money.

Before you actually decide which URLs to redirect, you need to do the basic setup for **mod_rewrite**. Add the following lines to Apache's configuration file, httpd.conf:

```
RewriteEngine on
RewriteLog logs/rewrite_log
RewriteLogLevel 1
```

You may need to twiddle the file name in the RewriteLog directive a bit to conform to your distribution's file system conventions. For example, in the default Apache build, the appropriate directive would be:

```
RewriteLog var/log/rewrite_log
```

In addition, you can increase the RewriteLogLevel from 1 up to a maximum of 9 to get more detailed information on what the rewrite engine is doing, which helps in debugging. However, once you are satisfied with the way it is working, it is a good idea to reduce the RewriteLogLevel to 0. This will stop all logging and reduce the load on the Apache server.

Having done this, you are now ready to redirect banner ads. Let us assume you want to redirect all graphics originating from URLs http://*anything*.unclick.net/adi_*anything*, http://*anyhost*.unclick.net/jump_*anything* and http://image_*anything*.click2com.net/*anything*. I chose to display the "Powered by Apache" logo (which comes with Apache in the file icons/apache_pb.gif) in place of the advertisements. Add the lines shown in Listing 1 to your httpd.conf file to display the Apache GIF instead of images from these sources, where, of course, you replace apache.your.dom with the name of the Apache server you have set up.

Listing 1

As you can see, each redirection consists of two parts: a RewriteCond and a RewriteRule. The RewriteCond identifies the host (**%{HTTP_HOST}**) we want to redirect from. The third string in the directive is the name of the host, given as a regular expression. In regular expressions, the period (.) stands for any character, so we use **"."** to denote a literal period in the domain name. **".*"** stands for any arbitrary string, and the caret (^) forces a match at the beginning of the host name. Thus, for example, the regular expression **^image.*\.click2com\.net** matches the hosts image.click2com.net, images.click2com.net and images05.adverts.click2com.net, but does not match ads.image.click2com.net.

Once we have found a host with RewriteCond, we use the following RewriteRule directive to see if any part of the URL matches the first string (regular expression) in the rule; if it does, we redirect to the appropriate (local) graphic. For example, having found a host in the unclick.net domain in the first RewriteCond, we then check in the RewriteRule which follows it if the string **/ad*anything*** occurs anywhere in the URL. If it does, we redirect the Apache proxy to send instead the image /icons/apache_pb.gif from the proxy host. Of course, you can use any image you like, and it need not necessarily be on the proxy server host—any URL is fine. The final [R] converts the HTTP request into an HTTP REDIRECT (302). Don't worry if you don't know HTTP; I don't either, but it's required.

You can add as many rewrite conditions and rules as you like to redirect banner ad sites. I'm still looking for a comprehensive listing of well-known ad URLs—let me know if you can add some.

Notes

The method of setting up the Apache proxy I have described is purely for the sake of redirecting banner ads. The Proxy module has many other features which are not covered here. Perhaps the most useful of these is its ability to cache often-accessed documents on a local disk, and further reduce bandwidth and time needed to browse the Web. Try playing around with the cache directives in httpd.conf, and judge the effect on your browsing speed.

Also, the method of configuring the rewrite engine I have described here is only one way. This method may not be optimal, as it is the result of one day's fooling around with mod_rewrite, but it works and serves my purpose.

mod_rewrite is an extremely powerful and versatile rewriting engine, and we have seen only one application of its features in this small redirection application. More information on Apache, its modules in general, and mod_rewrite in particular, can be found in Resources.

Another tip you may find useful is to combine the two secondary Apache config files, access.conf and srm.conf, into the primary config file, httpd.conf. This speeds up Apache quite a bit, especially on heavily loaded servers. In order to do this, copy the contents of access.conf and srm.conf into httpd.conf, preferably somewhere at the end, although location doesn't really matter. Now add the following two lines near the beginning of httpd.conf:

```
AccessConfig    /dev/null
ResourceConfig  /dev/null
```

Again, location isn't very important, as long as you don't insert those lines into the middle of a multi-line directive.

A rather whimsical description of this approach is available at the Apache site, listed in Resources (see “why are there three config files?”).

Conclusion

I have been using Apache to redirect Internet advertising for a while now, and am very happy with the results. Browsing on my slow link at home is a much more enjoyable experience with highly reduced wait time.

When I first started using Apache, I never expected I would be able to use it pro-actively for this particular purpose. Ralf Engelschall, the author of mod_rewrite, didn't expect his brain-child to be used for this purpose, either. The fact that it can underlines once again the beauty of the philosophy behind UNIX, Linux and Apache—if you make the parts general purpose enough and

give the facility of combining them, their sum invariably becomes greater than the whole.

Resources

email: raju@sgi.com

Raju Mathur (raju@sgi.com) ostensibly works for SGI in India, but manages to spend inordinately large amounts of time with his first love, Linux. He has been using Linux since the kernel 0.99.11 era and is currently the coordinator of the Delhi Chapter of the India Linux Users' Group. He is married to Aparna, a past-life therapist, and is the proud father of two children, Shiv and Ella.

Archive Index Issue Table of Contents

Advanced search

Home Entertainment Linux MP3 Player

Goran Devic

Issue #71, March 2000

Here's a way to store your CDs and tapes away forever, while still enjoying the music.

Imagine lying back on your sofa in the living room with your remote control unit. You press the channel selection button, and a synthesized voice speaks out, “Alternative”. You keep pressing the channel-up button and the voice speaks out different music categories: “Children's”, “Classical” and others. You chose the “Classical” category and press **ENTER** on your remote control, and the voice starts listing the albums in the same manner. You select “John Williams, Spanish Guitar Music”, and the near CD-quality guitar music starts playing.

No, the music is not playing off some CD changer, and the synthesized voice is actually interactive, responding to your remote controller actions. The music stored is your complete CD collection, plus those old tapes (somehow still around), digitized and compressed as MP3 files. The whole show is controlled and played by a Linux server hidden in your closet, connected by an audio cable to your home entertainment amplifier.

As soon as I realized the scenario described above is already within our reach, I decided to code the missing pieces and somehow glue it all together.

MouseREMOTE from X-10

The remote control unit is the central part of my design. The controller I used, called “MouseREMOTE”, is part of the “BigPicture” package that lets you transmit audio/video signals from your computer remotely to your TV. Being a “control freak”, I purchased the whole package some time ago and set up all the house lights to be remotely controlled. The remote mouse is especially useful, as it performs like any other controller for audio/video components (it can also control X-10-based home automation devices). In addition to every other button commonly found on a universal remote, it has a rubber mouse pad on

the face and two buttons on the back of the unit. The controller sends RF signals when its buttons are pressed to the receiver unit which, when its buttons are pressed, has a regular computer mouse pass-through, so you can still use your regular serial or PS/2 mouse. The remote-mouse software packets will be inserted in the stream of packets sent by your regular computer mouse. Unfortunately, the MouseRemote comes with only the MS Windows software to assign actions to different keys. I installed a high-quality audio cable from the Linux sound card line-out connector in the server room down to the living room amplifier unit's RCA input.

That was all I had to do on the hardware side. For software support, I had to modify the mouse-server program to accept the codes sent by the MouseRemote unit and pipe them to an MP3 player program, which will perform different functions depending on the remote-mouse selections.

Table 1

I determined the MouseRemote specification by reading codes using the modified **gpm** program. MouseRemote is detected as a bare-mouse type, and it returns packets of three bytes per event. Table 1 shows the codes returned for the events. As you can see, almost all of the keys return some code, giving this remote unit tremendous potential versatility.

There are few peculiarities: mouse button codes (left/right buttons located at the bottom of the controller unit) are transmitted three times in a row. Only one button can be pressed at a time; the other one is ignored while the first one is being pressed. You can press and hold one mouse button and press any key. If you hold down mouse buttons, the codes will not start repeating; only mouse pad (mouse move) codes repeat with no apparent delay. All other keys are repeating with a short delay of approximately 1/20 of a second.

The mouse movement pad detects three levels of pressure that can be used by the mouse driver software to accelerate movement. These levels set bits 0, 1 and 2 of the second and third byte in the mouse packet. Those bytes are used to determine the difference in movement between two consecutive packets, so the firmer the pressure, the faster the movement appears.

Modifying gpm

I decided to modify mouse server gpm (version 1.13). The key function added is **x10codes** in the file gpm.c (see Listing 1). The function argument *data* is an array of three bytes making up the current mouse packet. The detection of the MouseRemote device is not deterministic. I rely on the property that all its buttons return codes in the specific range; that is, the range of the possible, but not probable, combination of codes returned by the normal mouse movement.

In particular, the first code is always in the range of 44 to 47, with the third code always being 0x3F. A regular mouse would have to be moved quickly at just the right speed in order to get the same code out. In the unlikely case it happens, the second level of safety is the existence of a named pipe, /dev/x10. If the pipe does not exist, the packet will be passed on for normal processing. I used that named pipe to connect to a reader of remote-mouse codes.

Listing 1

Transforming mp3blaster into Remote Player

Of all MP3 player programs available for Linux, only **mp3blaster** has a user interface nice enough for easy control of directories and files to be played. The program supports multiple groups and can interactively select among directories. I decided to use both features. All albums are stored in separate groups, and at any time, I can toggle to directory browsing and select albums hierarchically. The prerecorded voice files are good guides for where you are and what you are doing (see Speech Synthesis).

Given that all your MP3 files reside in some hierarchical directory structure, say at /home/mp3, you need to set the environment variable **MP3_ROOT** to it. That way, the player will know where the files are, and during the directory browse, it will not allow you to accidentally change directories to one above it. (Remember, we are physically too far from the keyboard and display to fix any mishap.)

The mp3blaster is invoked with the option **-x** which I added to activate all the remote features.

In order to get full use of the groups features of mp3blaster, you need to manually set the current working directory to the MP3_ROOT directory (where your music hierarchy starts), start mp3blaster and press the **F1**, **F5** keys. The **F5** function key will add all directories as groups, thus effectively listing all your albums. Then, you can save the list by pressing the **F6** key. So now, you would start mp3blaster with the following syntax:

```
/usr/bin/mp3blaster -l
```

You can run the program from the **init** script if you wish, or from an idle console; it doesn't matter, since it will connect to the remote control unit and perform its function in the background. Now, using your remote controller, you are able to browse the directories and play songs. As you will see, the speech synthesis is also coded in to give you feedback.

mp3blaster has two modes of operation: group and file. The group mode of operation accepts the following remote controller keys:

- Channel +: selects next album (group). Voice will announce album name.
- Channel -: selects previous album (group). Voice will announce album name.
- Play/Enter: plays the whole album.
- A*B: this key toggles between group and file selection modes.

The file selection mode is more complicated, as we are allowed to traverse directory structure and play arbitrary albums. The remote keys accepted in this mode are:

- Channel +: selects next subdirectory. Voice will say its name.
- Channel -: selects previous subdirectory. Voice will say its name.
- A*B: this key toggles between group and file selection modes.
- POWER: switch to groups mode.
- Play/Enter: enter the subdirectory and play all mp3 files there.
- Shift: step back one directory level.

During play, the following keys are available:

- Play: continue playing after a pause.
- Stop: stop playing, return to the mode where we initiated playing.
- Pause: temporarily stop playing.
- Volume +: increase volume.
- Volume -: decrease volume.
- Rew: play previous song.
- FF: play next song.

Technically speaking, I added two threads to the mp3player program. One thread is always busy waiting for the remote codes from the /dev/x10 pipe. As soon as it gets them, it sets some variables used by the player class. The other thread is looking for the sound files (I call them voice files) that need to be “spoken”. In essence, via some mutexes and signals, the player requests presynthesized sound waves to be sent to /dev/audio, and this thread is making sure they actually end up there. (All voice files are pre-recorded and stored in a known location.) Voice files are spoken representations of directory names, named after the music categories and artists, and contain their songs.

Looking at the code, the process of inserting actions into the input loop of mp3blaster can be viewed as somewhat hacky, but most codes are just inserted as keys that would be pressed for an equivalent action from the keyboard

anyway. The number of changes is rather large to be printed here, so please see the file src/main.cc for details.

Speech Synthesis

When wanting to browse your mp3 music albums with no computer monitor to guide you, the natural substitute for vision is the voice. I decided to use **festival**, an excellent speech synthesis package. It is not only a current research development project that is growing and improving daily, but one you can actually use as soon as you install it.

festival can generate speech on the fly, as you type any text interactively, or you can pipe in a text file and it will synthesize it. None of these real-time approaches seems to be fast enough for interactive menu selection. I needed immediate voice response, and generation on the fly introduced a delay proportional to the length of the album names, noticeable and annoyingly long for normal use. The solution was to create a subdirectory containing all voice files to be used during the browsing. This way, the MP3 player program does not have to call festival to generate each album name as we browse it, but can use wave files cached in that specific directory. A drawback to this approach is the disk space taken up for the voice files, but that space is negligible in comparison to the actual MP3 files which amount to 50 to 60MB per album.

Once you generate voice files using the festival program, you can test each of them by simply piping them to /dev/audio. Also, you may want to change diphone for some albums (I found the Spanish diphone to make much better pronunciation for the groups of International albums). Alternatively, you could manually record all your voice files, thus eliminating the need for a speech synthesis program.

The Perl script in Listing 2 is used to traverse all the subdirectories under the MP3 files root directory and in creating all necessary voice files used by the mp3blaster player.

Listing 2

In order to generate necessary voice files, you would run this script every time you add an album or change the directory structure. You can run the script with the option **-clean** to ensure all old files are deleted before creating a new set.

All voice files are stored in your root mp3 directory under the subdirectory .vocals. They are vocalized interpretations of all subsequent subdirectories, and thus all the album names as well (they are just subdirectories at some terminal node, and they contain only MP3 files).

The Perl script first creates text files (original subdirectory name with the extension .txt). They contain a slightly modified name stripped of all non-alpha characters. This is done to help the speech synthesis program generate more precise sounds. Lastly, the u-law audio files are created based on the content of those files. If you are not satisfied with how it sounds, you can change the phonetics inside the text files, delete the voice file and rerun the script in order to get the optimal pronunciation.

Conclusion

The technology to do almost anything with the music is already available. In my opinion, without Linux and open-source software, the task of building such a remote-directed MP3 player would be much more difficult. Since my Linux server is up 24/7, it makes sense to use it any time I want to listen to music as well. Although the system I dedicated for it is a rather modest one (Cyrix 6x86 running at 166MHz), the MP3 player is using around 40% of the CPU time when playing, and there are no audible interruptions even when concurrently serving web pages. In addition to compressing all my CD collections and storing them on my Linux server as MP3 files, I also digitized my old tapes and, after some processing of the sound in order to improve it, stored them in the same music hierarchy tree. Finally, I could remove all the CDs and tapes from my living room and store them away for good. Now, every musical piece is quickly accessible by a touch of a remote-controller button.

Resources

email: goran@3dfx.com

Goran Devic (goran@3dfx.com) has a BS in computer science from the University of Texas in Austin. He has worked on developing Cirrus Logic's Laguna 3D graphics accelerator (5464/65) and has three patents issued and ten pending in the area of graphics accelerators. He is currently working on new generation graphics accelerators at 3dfx Interactive, Inc. He spends his free time with his year old son, Siddartha. His hobbies include playing classical guitar, photography and Eastern spirituality.

Archive Index Issue Table of Contents

Advanced search

Linux DVD Update

Craig Knudsen

Issue #71, March 2000

Watching movies is a favorite leisure-time activity for many people. Today you can watch movies on your laptop computer as well as your TV screen. But can you do it with Linux?

Under most circumstances, when someone refers to a DVD (digital versatile disc), they're talking about a movie. After all, DVD originally stood for Digital Video Disc. However, there's much more to the DVD standard than just the DVD-Video format.

Many new computers come equipped with DVD-ROM drives and a Windows-based application for playing DVD-Video. The difference between DVD-ROM and DVD-Video is comparable to the difference between CD-ROMs and audio CDs. The DVD standard also includes DVD-Audio (the likely successor to the audio CD), and the DVD-RAM and DVD-R writable formats.

What's so great about these DVD formats? They store huge amounts of data compared to a CD. The typical DVD can hold 4.7GB of data, compared to the 650MB capacity of a CD.

Should your next computer have a DVD-ROM? It's clear that DVD-ROMs are well on their way to replacing CD-ROMs. The additional cost of a DVD-ROM over a CD-ROM is less than $100 and will continue to drop.

DVD-ROM players can also read audio CDs and data CDs, so there's no need to have both a CD-ROM and a DVD-ROM drive. However, some older DVD-Video players are unable to read CD-R discs. I was disappointed to find out that an audio CD-R I created worked fine in a cheap portable CD player, but not in my (much more expensive) DVD player. Strangely, the CD-RW media usually work fine. There's a good technical explanation of this problem in the DVD FAQ. Newer DVD-ROMs shouldn't suffer from this limitation.

Linux File Systems and DVD

What you're probably wondering is whether or not you can use a DVD-ROM under Linux. The good news is you can, and it's simple to do.

Most DVDs use a bridged file system so they can be mounted using either the ISO 9660 file system (used by CD-ROMs) or the newer UDF file system. Most likely, your Linux kernel already has support for ISO 9660 built in, so you should be able to mount a DVD in your DVD-ROM drive.

If you want to try out the UDF file system, you will need to either patch your 2.2 kernel or upgrade to the 2.3 kernel (or 2.4, if it's out). The UDF file system offers some additional features over ISO 9660, but none are necessary to read a DVD-ROM, unless it is a UDF-only DVD.

Being able to read a DVD and play a DVD movie are two different issues. You won't be able to pop your copy of *The Matrix* in and watch it right away.

Playing DVD Videos

It is possible to play DVD videos under Linux; however, at this point, it's a challenge to do so. (See section on Breaking DVD's Security.)

Two primary projects are underway to bring DVD Video playback to Linux: LiViD (Linux Video and DVD Project) and LSDVD. The focus of LSDVD is on a commercial, closed-source solution, while LiViD aims to develop an open-source DVD player. Both projects are early in the development process.

Movies can be decoded using either hardware or software. Hardware decoding takes place in a dedicated card, such as Creative's Dxr2 DVD Decoder. You can obtain a Linux driver for Dxr2 at Creative's open-source page. Software decoding requires a tool to decrypt the Content Scrambling System (CSS) copy protection.

If you don't have a Dxr2 card, you'll need a 500MHz Pentium III or better to handle software decoding. The LiViD Linux DVD HOW-TO provides instructions on setting up software decoding.

You'll need to download some extra software, including a kernel patch and the CSS decryption tool. However, due to legal pressures, the needed CSS utility disappears from web pages soon after it appears. If you're a skilled surfer, you should be able to find a copy of the tool somewhere.

Convergence Integrated Media is working on a PCI DVD Decoder card that will take care of CSS decrypting as well as video and audio decoding. The card and Linux drivers should be available soon.

If you're the adventurous type, try that DVD movie under Linux. However, most users will want to wait until a more user-friendly package for DVD-Video playback is available.

Breaking DVD's Security

DVD-Video uses a copyright protection scheme called the Content Scrambling System (CSS) which is designed to prevent people from duplicating the digital content of DVDs. Several groups of Linux developers are working on creating a Linux DVD player. LSDVD aims at creating a commercial player for Linux that will not be open source, due to licensing restrictions and costs. (See their home page for a good explanation of these costs.)

LiViD wants to create an open-source player. There are open-source decoders for both the AC-3 audio and MPEG-2 video formats found on DVDs. The CSS encryption was the last obstacle to playing DVD movies on Linux.

A Norwegian group called MoRE reverse-engineered the Windows-based software DVD player from Xing Technologies, a subsidiary of Real Networks. They discovered a mistakenly unencrypted key that could be used to unlock DVD movies.

It's interesting to note that U.S. export restrictions on encryption played a part in this. The CSS scheme uses 40-bit encryption, the U.S. export key-length limit, allowing the MoRE team to crack the rest of the unencrypted keys. Thus, the export restriction made MoRE's job easier.

MoRE made their work available to others by posting their DeCSS decryption program, the CSS algorithm and the encryption keys to the world.

The breaking of CSS caused wide speculation on the future of DVD. Some suggested that certain movies wouldn't be released on DVD. Others speculated that DVD-Audio players would be held up. Most seem to agree that none of it would have happened if someone had made a Linux DVD movie player available.

Resources

Talking to Jens Axboe

email: cknudsen@radix.net

Craig Knudsen (cknudsen@radix.net) lives in Fairfax, VA and telecommutes full-time as a web engineer for ePresence, Inc. of Red Bank, NJ. When he's not working, he and his wife Kim relax with their two Yorkies, Buster and Baloo.

Archive Index Issue Table of Contents

Advanced search

Linux and IBM PowerPCs

Daniel Lazenby

Issue #71, March 2000

If you have an RS6000, you could be running Linux.

Several months ago, IBM announced it would support Linux on IBM PowerPC hardware. Since then, IBM has given their sanction to a specific distribution of Linux for the IBM PowerPC-based RS/6000 platforms. To round things out, IBM has also made progress on arranging commercial-grade Linux software support.

Supported Machines

IBM's announcement of Linux support actually bridges two product lines: the Intel-based and PowerPC-based platforms. The Intel-based product line includes systems such as Netfinity, IntelliStations, Thinkpads and the PC300 series. Information on compatibility of Netfinity systems with Linux is available at the Key Labs web site.

Linux distributions supporting the Intel-based product line include Red Hat, Caldera, TurboLinux and SuSE. IBM has established a marketing and support alliance with each of these vendors. For example through these vendors, there is now a Linux release of IBM's DB2 Database. Things change so quickly, one should check with each of these Linux vendors to verify their current offerings. Running Linux on Intel machines is quite common, so I will not say more about Linux on the Netfinity platform.

The RS/6000 product line contains Power, Power2, Power3 and PowerPC architectures. Each of these architectures contains several different classes of machine capability. Within the RS/6000 product line, official Linux support is limited to just a few CHRP PowerPC architecture systems. At the time of this writing, the supported systems include RS/6000 Models 7046-B50, 7043-150 and 7025-F50. There are plans to add support for the model 7043-260. The F50 and 260 are multiprocessor machines.

If you are going to purchase one of the supported systems to run Linux, make sure optional components being purchased also support Linux. I've seen one posting by a 7043-150 owner having problems installing Linux. It turns out several different graphic cards can be purchased with the 150s. Only three models of the graphic cards were supported—guess which one wasn't in the machine. This is one area that must be verified prior to purchasing your supported RS/6000.

Those investing in the supported PowerPCs should have access to detailed information on the specific hardware configurations that were used to determine that Linux can be installed and runs properly. The IBM PowerPCs need a site like the one Key Labs has for compatibility of Netfinity products and Linux.

Sanctioned Linux Distribution

Terra Soft Solutions, Inc. produces the Yellow Dog Linux (YDL) distribution. This distribution is the official IBM-sanctioned distribution for the supported PowerPC RS/6000 platforms.

YDL Installation Guide

The YDL Installation Guide is well-written and laid out. Unfortunately, the install guide that came with my copy of the media is heavily written and illustrated for the Macintosh hardware and install processes. YDL made a couple of change pages available. These pages primarily added single-sentence references to the model numbers of the supported RS/6000 platforms. Included was one page on installing YDL on an RS/6000. There are six pages with illustrations discussing preparing the Macintosh for Linux installation. Hopefully, the RS/6000s will receive a little more print in the next release of the guide.

Sources of Support

For supported RS/6000 systems, Terra Soft Solutions, Inc. offers installation support for a reasonable fee. I would recommend purchasing the installation support if you are installing YDL on a supported RS/6000. The additional cost for this support is nothing when compared to the cost of a supported RS/6000. Terra Soft considers installation support to be at an end once the product has been installed. According to the installation support description, installation is complete once root access has been obtained.

An agreement has been established with Linuxcare, Inc. to provide post-installation support services and training for the RS/6000 and Yellow Dog Linux. Linuxcare offers four levels of on-call support. Prepackaged support levels have been grouped into Bronze, Silver, Gold and Platinum levels of support. There

are options for obtaining customized levels of support, including 24-hour, 7-day-a-week support.

Other IBM PowerPCs

Proud owners of older IBM PReP PowerPC machines may also be able to use a PowerPC Linux distribution. Yellow Dog Linux is heavily oriented toward the CHRP machines. Nevertheless, their web site indicates the official YDL distribution has been successfully installed on the following non-supported IBM models: the 830 and 850, the 860, the 40P, the Nobis and INDI platforms. I understand the Nobis and INDI are Italian machines. Several Motorola systems were also listed as working, but not officially supported for this distribution. Most of these PReP machines have not been sold for several years. It looks as if they are beginning to enter the secondary and hobbyist market.

The 830/850 machines came from the PC side of IBM. The 850's proper name is IBM Personal Computer Power Series 850. It is catalogued as Personal Computer Model 6070 and comes with either a 100MHz, 120MHz or 133MHz CPU. This IDE-based PReP PowerPC system is not a member of the RS/6000 product line, nor is it a member of the RS/6000 43P family of machines. While some folks have gotten a release of Linux running on this machine, getting Linux installed on this IDE-based machine can be a major, and very frustrating, challenge.

Another machine from this PReP era is the Thinkpad Power Series 850 (also known as a Model 7249-850). The RS/6000 Notebook 860 (Model 7249-860) replaced the ThinkPad 850. I have seen a few mailing list postings indicating some folks have their ThinkPad 850 and Notebook 860 running Linux. These SCSI-bus PReP machines seem to have had a little better luck running PowerPC Linux than the IDE-based machines.

The 40P is another system that is about the same vintage as the PC Power Series machines. The 40P is another desktop PReP SCSI-2-based PowerPC system. This unit is categorized as an RS/6000 platform and is formally known as a Model 7020-40P. This machine type is listed as having had Linux successfully installed. While listed, the install process is not at all smooth and does have its difficulties.

The RS/6000 Model 7248 is another IBM PReP PowerPC platform. On the surface, there appear to be several similarities between the 7248 and the PC Power Series 850. Don't let these similarities fool you; there are differences between the two machines. The 7248 machines came from the RS/6000 side of IBM. The model 7248 was the first generation of 43Ps. These came with either a 100MHz, 120MHz or 133MHz CPU. This SCSI-2-based system was replaced with the current CHRP model 7043 version of the 43P. I've seen a couple of postings

indicating success with installing and running PowerPC Linux on the model 7248 versions of the 43P.

I recommend caution if you plan to install YDL on a PReP machine. After several hours of trying, I still could not get my IDE-base PReP machine to recognize the YDL floppy boot files. I switched to another distribution and didn't have any better luck. The SCSI-based PReP machines seem to have an install success rate better than the IDE-based ones. Vendors such as LinuxPPC, Inc., Debian, TurboLinux and others offer their own versions of PowerPC Linux. One of these distributions may play better with your PReP machine. You will have to experiment to find out which one is best for you. Hopefully, PowerPC Linux support for these IBM PReP PowerPCs will expand as more of these older IBM machines come into the hands of the Linux-capable.

Do Your Homework

If you are going to venture into installing Linux on a PReP IBM PowerPC, spend some time reviewing both the current and archived mailing lists for Yellow Dog and LinuxPPC. Let people on the list know what you are planning. Be specific about which Linux distribution you plan to use and the model of your PowerPC. They can help you avoid some blind alleys and dead-end avenues.

Proper System Firmware

Not having the proper firmware installed on your PReP machine could affect your installation of Linux. It may not be important to some, yet most of these PReP machines require a specific firmware release to be Year 2000 ready. IBM maintains a Year 2000 web site where you can verify whether a different firmware release is required for your system. Caution: use only firmware designated for your machine—doing otherwise could break your machine beyond repair. IBM's Year 2000 site for firmware information is in Resources. Virtually all of IBM's PReP machines are listed.

Disclaimer

The information in this article was gleaned from various web sites, lists, newsgroups, flyers, presentations and conversations. The accuracy of this information is subject to change without any notice. Please check the referenced web sites for current information and offerings.

Resources

email: d.lazenby@worldnet.att.net

Daniel Lazenby (d.lazenby@worldnet.att.net) first encountered UNIX in 1983 and discovered Linux in 1994.

Archive Index Issue Table of Contents

Advanced search

Update on LPI Certification for Linux Professionals

Dan York

Issue #71, March 2000

Need to be Linux certified to get that dream job? The LPI provides the means to obtain certification.

Back in the July 1999 issue of *Linux Journal*, I introduced the community project known as the Linux Professional Institute (LPI) and its efforts to establish a program for certification of individual expertise with the Linux operating system. Since that time, a great deal of activity has occurred, and it is time to provide an update on what has been going on.

A Brief Review

For those of you who have not followed the growth of LPI, the project started in the fall of 1998 from a series of mailing lists where people were discussing what a certification program for professionals might look like for Linux. It was agreed that a solid and well-respected certification program would help increase the pool of people who could support Linux and accelerate the corporate adoption of Linux.

Throughout early 1999, the lists discussed the matter and emerged with a series of "consensus points" about how a program should be constructed. These points included:

- The cost of attaining Linux certification should be as minimal as possible.
- The certification program should be global and scalable to accommodate as many people as possible who want certification.
- The program should be distribution-neutral and vendor-neutral.

As part of the development, it also became clear that LPI would specify only the certification standards, and *not* the education one undertakes to prepare for certification. We would publish the exam objectives on our web site, but leave the education up to the individual and third parties. If someone simply wanted

to prepare on their own, they could. Alternatively, they could buy books, take training classes at a training vendor or participate in on-line or computer-based learning.

We also undertook to pursue the development of our program using as open a process as possible, by using public mailing lists, our web site and any other means to involve and include many more people.

In March 1999, we created an Advisory Council (www.lpi.org/ac.html) that included individuals and companies within the Linux community who were interested in helping us move the project forward. Shortly afterward, we created a sponsorship program to help provide the financing.

Later in the spring, we performed an extensive job-task analysis that involved over 1000 people completing surveys and helped us extract the data to back up the choices we made for exam objectives.

By the middle of 1999, we had solid funding in place, thanks in large part to our initial Platinum Sponsors of Caldera Systems, IBM, Linuxcare and SuSE. We had also outlined a program consisting of three levels of certification, with multiple exams at each level.

Progress Through 1999

Throughout the summer and fall of 1999, LPI activity took place on a number of levels.

Program: as we developed the program further, we found our first exam was going to be too long for most people, so we wound up dividing it into two exams called “T1a” and “T1b” and making the T2 distribution-specific exams a bit shorter. A candidate now must take both T1a and T1b, as well as one of the T2 exams.

Exams: with our objectives clearly defined and available on our web site, we started soliciting test questions from the larger Linux community. Over 60 people participated in submitting questions, which were then reviewed by a team of 20 reviewers before being compiled into the format for test deployment. By November, our first exam was ready to go, and we formalized our contract with VUE (http://www.vue.com/) to deliver the exams at their 1700 test centers worldwide. Active development continued on the other exams.

Publicity: as we started to increase our development activities, we also worked to increase our publicity. We began to generate news releases more actively and maintain a presence at Linux-related trade shows and conferences. We had booths at the Linux Business Expo, the Atlanta Linux Showcase, Oracle

OpenWorld, the Bazaar and other events. We also unveiled our new logo and created buttons as giveaways. Many volunteers joined in and helped us out tremendously.

Web Site: as part of our publicity, we assembled a team of people to help redo our web site, and came out with a new design in the fall. Development continues with more features added all the time.

Financial Sponsors: the fall of 1999 also brought us SGI and TurboLinux as Platinum Sponsors, as well as Wave Technologies as a Gold Sponsor and a series of other companies coming in at our Silver, Bronze and Contributor levels (see www.lpi.org/sponsors.html for a complete list). More companies have committed to joining, and we are continuing to recruit even more.

Advisory Council: by the end of 1999, our Advisory Council was around fifty members, including representatives of all major distributions, prominent Linux and IT service companies, publishers, training vendors and many other companies. The Council had three face-to-face meetings in conjunction with conferences, and started to become active on a number of issues supporting LPI.

Other Certification Programs: in 1999, two other companies started up Linux certification programs, but then decided to merge their program with that of LPI. DigitalMetrics had developed a web-based certification program, but decided they wanted to avoid fragmentation of the market and join. They contributed test questions and helped in a number of other ways. Likewise, ProsoftTraining.com developed courseware that led to a certification exam. They, too, agreed to merge their efforts with those of LPI and began working with us to make that happen.

Training: while we were working on matters within LPI, our partners and supporters were out developing education materials to prepare for LPI exams. Caldera Systems, Linuxcare and SGI all developed comprehensive training programs to teach people through instructor-led classroom courses. Wave Technologies started developing courseware to be sold to other training companies. Several publishers, especially Macmillan USA, publicly announced upcoming books to prepare individuals for LPI certification. The Linux Training Resources web site (http://www.linuxtraining.org/) I maintain grew to over 100 training companies offering Linux training, many of them preparing for LPI certification. It was clear the necessary education people would be available.

Throughout, a hard-core group of volunteers has stuck with the project and spent incredible amounts of time bringing the program to reality.

Moving Forward

As I write this (December 1999), the year is drawing to a close, and a new certification program is being rolled out early in 2000. Although our first exam was completed in November, it has taken a while for the exam to be deployed, and we expect it to be out globally in January 2000. The second exam, T1b, should be out shortly after that, with the T2 exams coming during February and March. By the time you read this, all the exams of LPI Level 1 should be out and people will be getting certified.

There will, however, be much more to do. Throughout the early part of 2000, we will be defining LPI Level 2 and undertaking the surveys and other steps necessary to define the objectives and develop the actual exams. We will be doing a great deal of publicity to get the word out in every way, shape and form possible. We may even be thinking about Level 3 by April.

It will be an exciting time, and there will be many opportunities to become involved. We will need the help of many people to complete everything we wish to do. Please visit our web site to learn more about how you can become certified and help us move the program forward.

We are very thankful to the hundreds of people who have helped us so far, and we look forward to working with many more as the program progresses. Thank you all for your support!

Dan York is the Chair of the Board of Directors of the Linux Professional Institute and can be contacted at dan@lpi.org.

Archive Index Issue Table of Contents

Advanced search

Validation of Linux Certification

Rick Hynum

Issue #71, March 2000

Mr. Hynum talks to Tobin Maginnis of Sair Linux about his certification and training program.

Sooner or later, like it or not, it had to happen. The corporate “suits” have discovered Linux. Wall Street went ga-ga over Red Hat, Inc. last August and swooned again for VA Linux Systems in December. Predictably, some financial analysts stood back and wrung their hands like fretful parents, warning of the perils of puppy love, calling it a passing infatuation. And they're partly right. The stock market is ever fickle, and investors are always cruising for a fresh, new face.

Professional pessimists aside, those in the know about Linux and corporate America have no doubt the two were meant for each other. The more pressing concern among the Linux faithful is that free enterprise may compromise the Open Source movement and the ideals it has long stood for.

Dr. Tobin Maginnis, a Linux devotee since its inception, worries about that, too. As founder and president of Sair, Inc. of Oxford, Mississippi, Maginnis has developed a certification program aimed at moving Linux into the IT mainstream without sacrificing its integrity. The program, called Sair Linux and GNU Certification (SLGC), is unmatched in the industry for its breadth, depth and scope, covering training and testing in all recognized distributions. Best of all, it has the blessing of open source's founding fathers—pioneers like Richard Stallman, Eric Raymond and Bruce Perens, all of whom serve on Sair, Inc.'s board of advisors.

Compaq (http://www.compaq.com/), the second largest computer manufacturer in the world, has already signed on, partnering with Sair, Inc. to become a Sair Internal Training Organization. John Wiley & Sons (http://www.wiley.com/) is publishing SLGC test preparation materials and making them available globally, and Sylvan Prometric (http://

www.sylvanprometric.com/), the international leader in IT certification testing, is offering SLGC exams at its testing facilities worldwide.

Other deals in the making include negotiations with a strategic partner in South Korea to establish Sair Asia, which would serve the emerging Pacific Rim markets including South Korea, Japan and China.

And that's just for starters. Dr. Maginnis, a professor of computer science and a Linux researcher at the University of Mississippi in Oxford, told me:

> The question is not what can or will Linux do for corporate America. The fact is, Linux has been serving corporate America for years. It's not uncommon to ask a CEO or CFO if their company uses Linux and to have them respond, “No.” But ask their IT department the same question, and the answer always seems to be “Yes!”

> Linux is truly unique among operating systems in that many corporations have already adopted it for internal use without upper management's knowledge or approval. Production goals and budgets are always in conflict, and management always expects to be able to do more with the same resources. So a clever programmer says, “Hey, why don't we use that old Intel 486 we replaced last month? I know Linux will run on it. We can set it up, do a quick configure and finally get management off our backs.” That's more common than a lot of people realize.

Only now, upper management is finally catching on, and they're definitely not complaining.

Maginnis caught on to open source's potential two decades ago and became one of its earliest converts. When AT&T released its Version 6 UNIX to universities in the late '70s and early '80s, Maginnis used it to teach operating systems in his classes at Ole Miss. Between 1979 and 1983, he and colleague Don Miller developed Thriftnet, a suite of networking programs for UNIX and other operating systems, and the software was distributed free.

As open source evolved, Maginnis stayed abreast of every new development, from Minix to FreeBSD, which he taught in graduate-level courses. He began using Linux as a network server and firewall in 1995. His current research topics include the construction and integration of distributed operating system components into the Linux kernel and the development of a performance-oriented distributed operating system.

Besides his academic credentials, Maginnis is an experienced businessman and entrepreneur. The seven-year-old Sair, Inc., a custom software programming

company, has developed more than 45 large computer graphic and simulation programs for Chicago-area companies. Its client list includes Abbott Laboratories, Illinois Superconductor, Siemens Medical Systems, Dade International and Kraft General Foods.

A student gave Maginnis the idea for Linux certification in 1997:

> One of my students stopped me in the hallway one day and asked if there was a Linux certificate. There wasn't one, but the more I thought about it, the more I liked the idea. The problem was, enrollment in computer science programs had been on the decline, but there was a strong demand for trained professionals in the workplace. Vendor certificates were clearly becoming a trend in the industry. It occurred to me that this was a chance to do something real for the promotion and acceptance of open-source software.

The result was SLGC, aimed at providing the highest-quality certification training material and exams in the industry. Sair's efforts are focused on the continued development and distribution of the most accurate, timely and progressive materials in the constantly growing and evolving market.

These materials cover the most popular and recent distributions as well as regional variations for premium packages offered to corporate clients, both in the U.S. and internationally. The materials are offered through established training centers, including New Horizons Computer Learning Centers Inc. (http://www.newhorizons.com/) and Productivity Point International (http://www.propoint.com/), as well as through independent facilities around the world. Exams are available through Sylvan Prometric and VUE (http://www.vue.com/) in more than 140 countries (for a sample exam, go to Sair's web site at http://www.linuxcertification.com/).

In addition, Sair's self-study materials can be purchased through traditional retail outlets such as Barnes & Noble bookstores and through Amazon.com.

Best of all, SLGC is structured to follow existing models of professional continuing education and testing. Offered in four levels, SLGC qualifies its graduates as administrators, engineers and master engineers.

Purists need not be alarmed. SLGC is designed to serve the needs of corporate America while advancing the cause of Linux itself, a cause that remains dear to Maginnis' heart. “Our certification program is built, first and foremost, on a commitment to the Linux community,” he said.

According to Kevin Seddon, vice president of operations for Sair, Inc., Linux certification is the obvious next step in the Open Source movement's evolution:

> IT professionals have long recognized Linux for its strengths—stability, flexibility and efficiency. Those are valuable qualities in the corporate world as well, which is why Linux is being seen as a viable alternative to Microsoft. Corporate acceptance, in turn, means increased funding for open-source research and development in order to meet the private sector's needs. With major corporate clients on board, they will pour in the resources. The possibilities are limitless.

In addition, all the major computer manufacturers—IBM, Compaq and Dell, to name a few—have made a major commitment to Linux and open-source software. Most already sell machines with a preinstalled Linux/GNU system. “That's a clear sign of Linux's initiation into the mainstream,” Seddon said.

Compaq, for example, has committed hundreds of millions of dollars to Linux, putting it to work on their new X86 and Alpha servers. Under the terms of Sair's partnership with Compaq, Sair will provide corporate training (using Sair's materials) to Compaq personnel and help develop new Linux-based training materials for the X86 and Alpha ASE programs. “Toward that end, Compaq and Sair, Inc. will work together to expand Linux's capabilities, which, in the long run, benefits the Linux community as a whole,” Seddon said.

Sair is also working closely with Linux user groups around the world to hone and improve SLGC's knowledge matrix and objectives. And, in the finest Linux tradition, Sair has pledged to give back a percentage of its revenues to the Open Source community to help fund additional research and development. Maginnis has said:

> We recognize that the Linux phenomenon is the result of many idealistic people who will never be paid for their years of hard work. We think the best remedy to this situation is to actively support the development of new open-source software. It's common sense that if free software creates a special business environment, that business should find a way to give back to the Open Source community. Other companies have done that, and Sair will do it, too.

In that regard, Sair's plan is to identify and fund key projects which will best serve the Linux movement. Possible projects include creating anti-aliased fonts for X11 or creating a grammar checker.

Finally, Sair, Inc. is working to develop an international consortium of universities and businesses actively involved in open-source research and

development. The goal is to create a focused, structured and prioritized research network in which researchers around the world share information and collaborate on projects for the private and public sectors. Seldon said:

> This consortium would bring together many of open-source's greatest minds—both in the public and private sector—and help them work together to solve common problems. It would attract more funding for research. It would help cut back on duplication of effort. And it would provide an organizational structure through which information can be shared and disseminated more effectively and promptly.

Maginnis and company are convinced that certification will be the spark plug that fires the Linux engine in the coming years. Certification is, simply put, skill validation, but it means more than that. Certification sets criteria and standards that must be met in order to serve the needs of the market. It's a focused, results-oriented process which IT professionals are accustomed to and which the business world expects. Seldon told me:

> That's the beauty of SLGC. It's a comprehensive program created by someone inside the Linux community who has successfully “translated” the process by working through traditional outlets such as training facilities, testing facilities and the corporate community. And, through our board of advisors, some of the greatest names in the Open Source movement have contributed to it.

Even so, Sair's work is just beginning. International Data Corp. (IDC) research states that Linux was the fastest-growing server operating environment in 1998, growing more than 190 percent in that year alone and capturing more than 15.8 percent of the 4.4 million unit revenue shipment server operating systems market segment. According to a December 10 article in the *Los Angeles Times*, many corporate investors sense the potential for Linux to become the backbone of the Web. They also believe it could play a significant role in providing the software for the expected wave of new digital appliances, such as video recorders and mobile computers.

If that's the case, Sair, Inc. is uniquely positioned to meet an ever-growing demand for IT professionals certified in Linux. In conclusion, Maginnis said:

> All the signs are that Linux certification is an idea whose time has come. We are obviously excited about the possibilities. At the same time, we recognize our responsibility to the Linux community. We strongly believe in the ideals of the free software system, and we plan to show our commitment to those ideals in very tangible ways. For me, personally, it's been a privilege to have worked within this community for 20

> years and to watch Linux evolve and thrive. There are a lot of unsung heroes out there who made it happen and who will continue to make it happen. We look forward to working with them in the future and doing our own small part for Linux. We know we can't do it without them.

Rick Hynum is a scientific and high-tech freelance writer living in Oxford, Mississippi.

Archive Index Issue Table of Contents

Advanced search

Velocis Beta-3.0 Database Server

Avygdor Moise

Issue #71, March 2000

In a nutshell, Velocis delivers all we expect of it.

- Manufacturer: Centura Software (Raima Corporation)
- E-mail: sales@raima.com
- URL: http://www.birdstep.com/
- Price: $7,695 (eight users)
- Reviewer: Avygdor Moise

Before I tell you about Velocis, let me tell you about my company's system and why we need a database server. Itres Research Limited is a developer of the Compact Airborne Spectrographic Imager (casi), a multispectral imaging instrument that combines the sophistication of satellite geo-positioning systems (GPS) with the practicalities of aerial photography. The casi is used worldwide in a variety of applications, and is distinguished by its high precision and flexibility of use. The casi is a user-programmable, all-digital system that is easily installed in light aircraft. It operates in the visible and near-infrared range of the spectrum. This makes it an ideal camera for forestry, agriculture, environmental monitoring, land-use planning and water-purity applications. It is widely recognized as a valuable tool in remote sensing.

One of the more important features of the casi instrument is its ability to generate absolute radiance values from the scene. In simple terms, it can not only generate color photos, but it can also tell you the absolute spectral

signature of the scene. Given that most plants, minerals and other land-based objects exhibit different spectral characteristics, it is possible to classify the type of vegetation and minerals present in the photographed scene, including estimates of densities, health and concentration.

Our Problem

Precision does not come without a price. The data collected has to be passed through a number of processing steps, including radiometric correction calibration, geo-correction (the projection of each image pixel to a unique earth coordinate), mosaic (the tiling of all image strips into a single photograph) and analysis (the interpretation of the image and creation of a data product).

Before one can make use of the data, it is necessary to obtain detailed information on the casi sensor (camera) used to acquire the images (e.g., when it was last calibrated, where the latest calibration files are located, etc.) and associated flight records.

When the data processing is complete and the final report is generated, it is also highly desirable to have the means to review the parameters and tools used to process the data. This is an integral and necessary part of the quality-assurance program.

Databases Come to the Rescue (Not)

Itres' processing power is based mainly on Red Hat Linux 5.2 and 6.0. There are a number of processing stations, each equipped with dual processors (400MHz per CPU) and each connected to a 100Mbps LAN (local area network). Access to the workstation is provided by personal computers, which (unfortunately) run Windows 95. Graphic services are provided through xwin32 (an X-Windows server for Win32) in conjunction with Samba file access.

All users of Windows 95 *must* have access to Microsoft's Office Suite 97, primarily for editing their mail. This led to an unfortunate side effect, the availability of MS Access—the Microsoft Database engine. The need for a database solution to the information processing was so great that a number of users created *private* MS Access databases, which contain important processing-related information and product inventory information. There is a storage requirements database, component inventory database, customer contact database, on-line help database, flight-planning information database and the list goes on. The unfortunate part is that those databases are located on personal computers and are not easily shared among users. We attempted

to move the data to our main servers which run Linux, but we met with resistance for basically two reasons:

- The appeal, effectiveness and ease of use of the MS Access database front end.
- The average user is not attracted to UNIX.

As a result, we searched for solutions which would enable the average user to gain access to their databases using MS Access as before, while taking advantage of the capabilities of Linux as an effective file server.

The Velocis Database Engine

We had a number of minimal requirements for a database server, and the Velocis database server 3.0 by Raima Corporation (now Centura Software) met all of them. The key features of Velocis that match our requirements are as follows:

- Velocis application-server architecture supports multiple network communication protocols, including RPC (Remote Procedure Calls) and ODBC.
- The Velocis application can launch and manage its own threads (starting with beta release 3.0), each thread controlling one or more Velocis “login” sessions.
- Velocis client programs are included to give third-party ODBC database tools access to the Velocis database.
- Velocis is available as both a client and a server for AIX, Linux, Solaris, BSDI, HPUX, Win95/98 and WinNT.
- Velocis supports SQL 89 and a subset of SQL 92.
- It comes with an extensive set of printed manuals (and a PDF version on CD-ROM), including User's Guide, Installation/Administration Guide, Reference Library and Language Reference (SQL and Native).
- Velocis supports a number of database models: relational, networked and hierarchical and a combination of the three (which is how Velocis operates internally).
- Velocis is a licensed product that is distributed on a CD-ROM or by download from Centura's web site. The cost of the license is proportional to the number of concurrent sessions supported. Without a license, it will support up to eight sessions for 60 days. During the demonstration period, Velocis is fully functional.
- Velocis is a supported product, for a fee. The support includes product updates and consulting services in support of limited API development and product integration.

Installation

Velocis is distributed on a CD, and it occupies approximately 10MB installed. The Linux (or UNIX) installation is accomplished in a few simple steps:

- Mount the CD.
- Extract the appropriate compressed tar file for your operating system (e.g., Red Hat Linux 6.0) into a directory of choice (e.g., /opt/velocis3.0).
- Run the provided installation script.
- Copy the configuration shell scripts into the desired locations (either ~/.cshrc, ~/.profiles or /etc/profile.d).
- Modify certain initialization files (e.g., /catalog/velocis.ini and /bin/connect.ini) to set up the host-access identification.

Installation on Windows 95/98 or NT is similar, except for the use of winzip and install shield setup programs.

Starting Velocis

Centura Software provides a number of tools (most are available as C source code in the "examples" directory) to invoke, manage and communicate with the Velocis database engine. With the exception of the administrative tool (admin.exe under Windows and **rdsadm** under UNIX), all tools are available on all platforms. The most notable tools are:

- **rds**: the velocis database server, which is started as a daemon process.
- **vping**: an effective **ping** tool. It can be used to test the socket connection to rds.
- **rdsadm** (Linux) and admin.exe (Windows): a command-line interface and a graphic user interface (respectively) for the creation of databases and user-access management. In addition, it is responsible for managing the archive and recovery processes.
- **rsql**: a simple command-line-driven interface, which enables the user to establish one or more sessions and attach to one or more databases and interactively execute SQL commands.
- **sddlp**: a Velocis SQL Schema compiler utility.
- **ddlproc**: a Velocis non-SQL Schema compiler utility.

The steps necessary to start Velocis are as follows:

- Execute the install script in the Velocis root directory (as superuser).
- Configure the connect.ini and velocis.ini files located in the bin and catalog directories. This step is of extreme importance, since it establishes

the linkage between the database server name (alias), a socket and a computer on the Internet.

- Source the **rdshome.sh** or **rdshome.csh** scripts located in the Velocis root directory.
- Start (as superuser) the rds application server.
- Start rdsadm (UNIX) or admin.exe (Windows) to set up your databases. Any user can start rdsadm; however, the tool is password-protected.

Although the steps above are rather simple to execute, they assume the user has some knowledge of Linux and networking. It would be nice if Centura Software used RPM to bundle the package. It would be even nicer if they were to perform the installation of the scripts automatically and provide a System V-equivalent Velocis **init** script (with start, stop and restart) capability.

Running Database Applications

If you have reached this point, you are in business. However, it is important to note that Velocis is a dual-mode database server. Since it operates internally as a hybrid-relational model (network and hierarchical database engine), one has to take care when creating databases. Two database structures are supported: database schemas for non-SQL applications (also referred to as native) and database schemas for SQL applications.

Both SQL and non-SQL databases are created by defining the schema in a text file, compiled using the utility **sddlp** for SQL databases and **ddlproc** for non-SQL databases. Database models are implemented at design time. Interestingly, the network database model can be implemented in database designs using the SQL Database Definition Language (DDL). This is done by using Velocis' **CREATE JOIN** statement to create a permanent, pointer-based join on the primary and foreign keys of selected SQL tables. This improves performance by eliminating the runtime need for indexes between tables. The ability to implement both network, relational or combined network/relational database models using either the SQL DDL or the non-SQL (C-based) DDL is a unique feature of Velocis. Adding to this flexibility, the SQL API and record-level C-API can both be embedded in applications and used to interact with databases created using either DDL.

Databases are registered either when compiled, by adding a parameter to the command line or via the rdsadm (or admin.exe) tools. These utilities are also invoked when adding or dropping users.

The Itres Application

As was stated earlier, we desired a database engine that had a C API interface (for precise, record-level access to the database) and supported the SQL and

ODBC industry standards. The Itres database is running under Linux, which is the intranet database server. The server listens to ODBC communication requests from Windows 95/98 and NT and lets certain individuals run admin.exe under Windows. Yes, I'm guilty of doing that, but the Windows administration API is much nicer than the command-line-driven Linux version. The ODBC interface is provided primarily to enable Microsoft Access users to access the database, without yielding to Linux.

In order to interface with the Velocis SQL engine, we developed a Microsoft Access database conversion application based on VBA (Visual Basic for Application). This tool can read the table structures and data contents of any Microsoft Access 97 database and generate an ANSI SQL text file for the transfer purpose.

Any SQL-compatible database engine, such as Velocis, can read the SQL text files and import their contents from the Microsoft Access database. The Itres Microsoft-Access-based inventory database was successfully transferred to the Velocis.

It was also considered desirable for intranet users to have access (or views) into databases. The obvious viewing tool of choice is a web browser. To that end, we wrote CGI (Common Gateway Interface) scripts using Tcl (Tool Command Language from Scriptics Corporation). Tcl was extended as a dynamically loadable library, using Velocis' extensive native C API, to provide access to the database using SQL. These functions essentially utilized the sample source code for **rsql** found in Velocis' examples directory. From this point on, the effort shifted away from API integration with Velocis to a database-design project using SQL, which was our original goal.

Conclusion

In a nutshell, Velocis delivers all we expect of it: reliable operation, interoperability among computing platforms, a strong C API that supports complex operations and multithreading, standards-based ANSI SQL interpreter and reliable archive and recovery tools. Most importantly, Centura Software provides customer support (for a fee) necessary for operating in commercial settings.

We could use (and actually are using) PostgreSQL for prototyping SQL code on many computers, but when it comes to critical code, we prefer to use a supported tool. This preference also drove our selection of Red Hat Linux as a server platform.

Velocis is not perfect, and it does seem to have some potential rough spots (in the installation area and startup), but once configured and running, it works smoothly without complaints.

[Editor's Note: The company has changed name to Birdstep Technology, and Velocsis has become Birdstep RDM server 3.5.]

VAST

Avygdor Moise (avy@itres.com) earned a Ph.D. degree in experimental space science and has extensive experience in microcomputer technology, operating system design and application development for real-time systems. He has extensive hands-on working knowledge of Linux, UNIX and Win32. He is president of Future DOS Research and Development Inc. and Vice President, Research, at Itres. He is an independent consultant providing services to large and small companies on network design and implementation and Linux.

Archive Index Issue Table of Contents

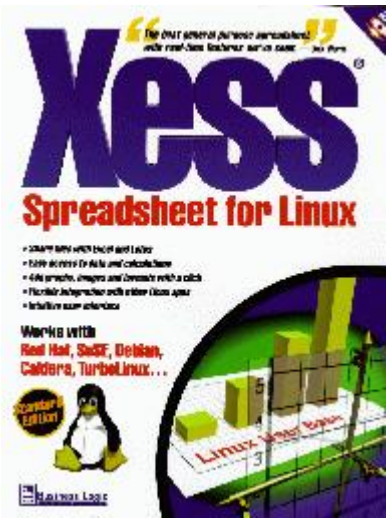
Advanced search

Xess Spreadsheet for Linux, Standard Edition

Jason Kroll

Issue #71, March 2000

What is so special about this spreadsheet, and why might you prefer it to the StarOffice or Applixware spreadsheets?



- Developer: Applied Information Systems
- Manufacturer: Business Logic Corporation
- E-mail: e-mail@blcorp.com or info@ais.com
- URL: http://blcorp.com/ or http://www.ais.com/
- Price: $69.95 US
- Reviewer: Jason Kroll

Welcome to yet another review of an application for Linux. Wait! This cannot be, for Linux has no applications! This is well-known, a fact we've heard since the days of “Windows is the future, get used to it!” and is probably the reason Linux has no user base and has failed miserably in the server market and the stock market. Hence, one must conclude that this spreadsheet is in actuality a spectre, and although you *think* it's real, it is a fantasy, as are the millions of Linux users you thought existed and the newbie Linux gazillionaires.

The subject for today is the Xess Spreadsheet for Linux, Standard Edition. We must specify Standard Edition, because there is also a professional edition which handles very large projects. However, the standard edition deals with up to 1024 columns and 99,999 rows per sheet, with 512 sheets per workbook, which should be more than enough for your next stat class assignment. Xess has been good about supporting several distributions of Linux, so you can almost certainly run it regardless of what distribution you have (which is, if I may opine, the way things ought to be). What is so special about this spreadsheet, and why might you prefer it to the StarOffice or Applixware spreadsheets?

First Impressions

To be honest, my very first impression was “Ack! What a stupid license!” You know the kind, those last-minute “installing this software indicates agreement” licenses full of the weirdest, off-the-wall restrictions, including a ban on un-assembling (do they mean disassembling?), de-compiling (if you figure out how to do this, let me know) and reverse engineering. Forbidding these things is rather absurd, and the legal reasoning behind these last-minute licenses is about as meaningful as the reasoning of little kids who always say, “I called it!”

Disassembly of a program (though not necessarily of a spreadsheet) is a superb way to begin learning assembly code, while reverse engineering had a very large hand in promoting the PC phenomenon (imagine if no one had reverse engineered the first IBM PC to enable cloning). De-compiling would be so clever and an excellent way of open sourcing binaries, but as far as I know, it hardly works. So why forbid these things, since they're no threat at all and the last-minute license probably isn't legally binding? Just in case. Gotta call it. The other problem is the bit about reserving the right to revoke the license, which is irritating. I'd certainly not depend on a product if someone had the right to revoke it whenever, but apparently that's how commercial software is. CD manufacturers don't bother with this licensing nonsense, and it hasn't hurt them any. At least the license is short, as is the small-footprint installation of the spreadsheet.

My second impression was “Geez, I wish we had this in college.” Instead, we had silly Windows boxes, with “blinky the dancing paper clip” et al. UNIX has such an academic history, I don't understand why it isn't the de facto standard in universities, but for some reason it isn't, although recent student protests may chase Windows out of colleges. Now that we have Linux, a free and superior OS, there's no excuse to keep running Windows in the schools (besides, a free-source OS is politically very correct). And, what with high-quality applications like Xess (or, for example, Word Perfect and the office suites), there's even less of an excuse.

Even though it comes from the commercial sphere, Xess is one more brick in a solid foundation of Linux applications. A free OS running a high-quality commercial application is, after all, preferable to a buggy, proprietary OS running a bloated proprietary application (with embedded flight simulator). What you may infer from the preceding sentence is that Xess is a high-quality spreadsheet, so operating from this assumption, let's have a look-see.

Xess Can Do

It's honestly a bit dull to list everything a package can do, and it risks sounding more like an advertisement than any sort of review. Still, if you're thinking of using a spreadsheet, you probably want to know what it's capable of doing. There isn't room to list everything, so I recommend going to www.ais.com/Xess/xess4_features.html for the complete list. I'll try to cover the major points here.

Xess looks like any other spreadsheet (Excel, Lotus, StarCalc, Applixware Spreadsheets, etc.), so its interface is obvious and intuitive. Xess has all the functions one expects of a spreadsheet, including some outstanding ones. At the basic level, Xess has the standard functions complete with conditional and Boolean operators, iterative and double-precision calculation, inter-sheet cell linking and formula constraint checking. The mathematical functions include the standards found on a good pocket calculator, such as exponents, logs, trig, matrices, as well as sigmoid, gamma and log gamma functions. The matrix operations in particular are extensive, even including Fourier transforms, correlation matrices, curve fitting and linear equation solving. Financial functions include the usual exciting things like rates of return, present and future values, interest rates, yields and all that. Statistical functions are thorough, and on the whole, there is more offered than I remember finding in StarOffice or Applixware.

Maybe it's funny that Xess is distributed by a company called Business Logic Corporation, because Xess, while wholly adequate in the financial department, is rather well-suited to scientific, statistical and mathematical operations. In any event, the test of whether a spreadsheet could be useful to me is if it can be my pocket calculator. Xess comes closer to substituting for my pocket calculator than other Linux spreadsheets, and even has some stat and matrix functions my calculator doesn't have.

Xess is largely a calculation-oriented spreadsheet. The calculation engine must be praised, because it's actually fast, although the graphing capabilities are less extensive. The graphics basics are covered, and you can make the standard scatter, line, area, bar, stacked bar, histogram, pie, surface, contour, polar, hi-lo, control and box graphs, and there is much technical flexibility. Still, there isn't much aesthetic control beyond standard representation, which is fine. As for calculation, there are more functions than I can count, nearly 250, all of which

are nicely referenced in the manual. I would be more than surprised if someone found the calculation capabilities inadequate.

Xtensible

Xess is endowed by its creators with certain indispensable utilities, such as direct read and write of Excel and Lotus spreadsheets (Excel 97 and WK4, but don't count on VB macros coming across), filtered text import and export, as well as HTML and LaTeX output. Obviously, it can print and do the usual generic things.

One big advantage for business users is that through the use of xsBasic, you can have macros to automate repetitive tasks (and many people find business boring precisely because of the endless streams of repetitive tasks). xsBasic is an expensive add-in ($200 US), but it's not bad for a Basic.

As a language, Basic has been constantly maligned, but I found the linear thinking involved in using Basic made the jump to assembly code particularly easy and obvious. The newer versions of Basic are similar enough to C that again, the transition is easy. In fact, when I finally had a computer with a hard drive and enough memory for a C compiler, I used to tell would-be C programmers, “Don't get intimidated; it's just like Basic.” (Of course, hackers who overheard would remark, “What Basic are you using, man?”) So, I think Basic is not such a bad language. It's particularly simple, and for some reason the business world demands simplicity to the point of idiocy, so (pardon the half-pun) Basic fits the bill. This particular implementation concentrates on dealing with the objects of the spreadsheet, from the borders and buttons to boxes, graphs and widgets. If you need this, you can buy it; contact AIS for information.

Apparently, some people are making interesting integration projects with Xess, although it's odd to me that anyone gets excited over spreadsheets. If you get really excited, there's even an API you can use that's based on the X Window System protocol.

Across Platforms

Another characteristic of Xess is that it is available across platforms. This includes not only the typical Linux and Windows NT, but also SunOS, Solaris, HP-UX, IRIX 5 and higher, SCO UNIX, Digital UNIX, Ultrix, OpenVMS, VMS, DG/UX and AIX. Check the web page for chip set and version specifics. You can probably put this spreadsheet on all the computers, even at extraordinarily diverse offices (and if you're using an unsupported platform such as C64 for some mission-critical application, I'd sure like to hear about it). This is probably more than the StarOffice and Applixware spreadsheets support.

Ad Infinitum

While it pains me to take sides in commercial affairs (because this implies siding against people who have put part of their lives into creating something which gets labeled “second-rate” or “not as good”), I am always compelled because it's important to help people make informed decisions; in this case the choice between Xess and other spreadsheets such as StarOffice and Applixware. While it's true that with the latter two you get whole office suites, their spreadsheet packages do not really compare. Applixware is probably the easiest to use, and StarOffice has a *lot* of features (not to mention being free of charge, which makes it nearly impossible to compete with), but Xess delivers more functionality and appears to calculate faster (it's hard to be sure on Linux boxes, which are already so fast). AIS concentrates exclusively on this one spreadsheet, whereas StarOffice and Applixware have whole office suites, so it stands to reason that the latter two would be hard-pressed to pay as much attention to the spreadsheet parts of their suites, while AIS can go beyond the basics and develop according to user needs faster. I should mention Xess pricing also goes beyond the basics; you can spend a fortune on it if you have such a mind. Still, one must economize, so it should be mentioned that most people will not find the StarOffice and Applixware spreadsheets to be limited at all; like many things in life, this is a case of good vs. better. But, if you do find other spreadsheets limited for your purposes, Xess is one solution. At the very least, get rid of Windows already—you don't need it anymore!

Further Rambling

For myself, the neat thing about Xess is that it brings to Linux a genuinely high-quality spreadsheet, like we'd expect to find on any other platform. Hopefully, this indicates that Linux users will no longer be stuck with almost-there-quality commercial wares. Too many producers go into business selling proprietary software that isn't up to par, and when I see their products, my only question is “why does this exist?”, because their packages are usually worse than the free offerings, not to mention being gratuitous (one point of free source is that we don't have to develop the same software several times over). While I usually can't stand proprietary software and seldom touch it outside the office (which makes some of my favorite activities completely impossible), I don't feel bad about recommending Xess to people who are willing to use commercial wares. AIS (the developer group) has actually delivered a high-quality proprietary application. A troupe of free software coders is hard at work on a spreadsheet named Gnumeric (part of Gnome Office), while the KDE folks are developing KSpread (a component of KOffice), so if you wish to contribute to a project, or just wait for the free spreadsheets to be delivered, those are your options. In the meantime, if you have a lot of money to spend on a commercial spreadsheet and aren't afraid to use it, Xess is probably your best bet.

Jason Kroll (info@linuxjournal.com) is Technical Editor of *Linux Journal*. In his mind, GNU/Linux is truly the miracle that saved computers.

Archive Index Issue Table of Contents

Advanced search

NetWinder Office Server

Jason Kroll

Issue #71, March 2000

"How many nets could a NetWinder wind if a NetWinder could wind nets?"--Dan Wilder, SSC Technical Manager

- Manufacturer: Rebel.com
- E-Mail: info@rebel.com
- URL: http://www.rebel.com/
- Price: $1000 US
- Reviewer: Jason Kroll

"How many nets could a NetWinder wind if a NetWinder could wind nets?"--Dan Wilder, SSC Technical Manager

And furthermore, what is a NetWinder? A NetWinder is a StrongARM-based Linux box, built as a network server. NetWinders come in three models: OfficeServer, Developer and ThinClient. Of these, the OfficeServer and Developer models are the familiar NetWinder server boxes, while the ThinClient is a new creation, a graphic terminal the size of a notebook, which one would presumably use with an OfficeServer. The Developer model is an expanded

OfficeServer designed, as one might infer, for developers. It has more interfaces and development tools, as well as an active developer community. The OfficeServer, our subject for today, is the standard NetWinder, a device on which to run your networks, whether they are dozens of office computers or just a few terminals around your house.

Any old Linux box can function just fine as a server (at last estimate, more of the Internet was running on “any old Linux box” than anything else), but the NetWinder isn't any old Linux box. The core of the NetWinder is the StrongARM architecture, RISC technology developed cooperatively by ARM and DEC (although DEC was bought by Compaq, Intel now produces StrongARMs). We'll get into the technical details of the chip later, but it's important to appreciate some superficial differences in the processor which enable the NetWinder to be so compact. StrongARM is popular with embedded systems because it runs on little energy and stays cool. Hence, the NetWinder can be tiny and resource-minimal, important factors since it is designed to be on all the time. It is intended to run headlessly; that is, without a monitor. Users need not know *anything* about Linux to use a NetWinder, and they may never even know which software it's running—it's just a small magic box, exactly what most people want. Before we open up the magician's funny box to see how it works, let's look at what it does.

Late one morning, my co-workers heard me whining quite loudly about my difficulties compiling some software on the NetWinder, followed by, “Good grief! It hasn't even got Netscape!” Needless to say, something had to be done to quiet me, so a fellow suggested, “Why don't you use it how it's *supposed* to be used?” In honor of this fellow, we shall now use the NetWinder as it's meant to be used; that is, as a headless Internet, file and print server.

Configuration of the NetWinder is fairly simple. As long as you assign the NetWinder a valid IP address, you can plug it in and immediately point a web browser at it. Configuration is supposed to take place under Windows, but it can be done through any browser. I actually prefer to plug in a monitor and keyboard, edit a couple of files and have it up in short order. However, if you know nothing at all about Linux, the web-based interface will walk you through the installation. Although I have heard reports of some people having trouble installing the NetWinder, I hardly had to do anything to it. Perhaps I have a more recent edition.

Once initially configured, the NetWinder will cheerfully perform (well, our last lightning storm that allegedly downed MS's “hacker-proof” box also rebooted the NetWinder), and if you want to make any changes, point your browser to its address and log in. You can log in as a user or as an administrator. The user menu offers directory information, infoplace document organization,

discussion group access, user profile access, a search menu and on-line help. What these things do is self-explanatory; they exist for generic users and office people, and the menus are clear and simple.

The administration mode menu is essentially root, with menus for editing user accounts, group data, host information, system management, service availability and network configuration. Again, the submenus are clear, allowing you to turn services on and off at a single click. Using the NetWinder web interface is as easy as accessing the web page of a printer (you may have noticed the curious trend of printers having their own web servers). One might expect a network server to be complicated, but it's not. Maybe that's the point of the NetWinder; it's not necessarily more powerful than a standard Linux box, but it's so accessible it makes the power of Linux available to those who have not devoted years to the study of UNIX-Linux networking power, with point-and-click simplicity.

What can you do with a NetWinder? Let's say you're rich from past mining exploits and you've got a mansion with over 60 rooms, some of which you would like to wire for the Internet. There's been a mammoth copy party and the guests have left computers all over: some Macintoshes from pre-Linux days, a dozen Linux machines, a couple of FreeBSD and NetBSD boxes, and even something running BeOS. An evil spirit that haunts your house insists on running Windows in the attic, so let's pretend you've got an NT box up there. While looking for a solution, you suddenly hear a child's voice say, “Welcome to NetWinder!” (which you'll hear *every* time you boot). Suddenly, you realize you are now ready to network all your machines.

First, get a really big hub and an ISP account with high-speed access: for example, DSL or cable modem. I recommend not wasting any effort on ISDN. If you already have a LAN, you can add the machine to the network and use the web interface to set everything up. Otherwise, hook up a monitor, keyboard and mouse and try manual configuration, either directly through editing files or with the **nwconfig** program. As soon as you can get to the NetWinder with a web browser, you can configure everything quickly. Plug the 10 BT Ethernet into your DSL or cable modem (or use the serial port if necessary), the 100 BT Ethernet into your hub's uplink, then plug all your computers into the hub. Configure individual addresses and names for them, and you've got a network.

Presumably, you also want services. The NetWinder menus and manual explain how to set up services. The obvious things to start with would be the Web and e-mail, since these seem to consume most on-line interest. Apache is always running so you can reach the administration menu, but you can also run a generic web site with it through virtual hosts; there's even a web page creator program, although it's not exactly advanced. Hence, residents in your house

can have their own web pages. E-mail is simple to set up; just enable SMTP through the administration menu, and point your local mail reader to the NetWinder. To read the mail from your terminal (instead of logging in to the NetWinder as a user), enable POP-3 or IMAP services and tell your local mail reader the NetWinder's address. You can establish e-mail forwarding and filtering and even set up automatic response messages for when you're away.

The NetWinder can do whatever you expect of a Linux box, including file and print sharing across UNIX, Macintosh and Windows machines. Advanced administrators can do various things, from establishing BBS-like threaded discussion forums to rather bureaucratic things such as imposing quotas all over the place. Using the NetWinder as it's designed to be used won't disappoint, but would these machines be viable as home computers?

The NetWinder has almost everything you could want for a home computer, except the software. The OfficeServer has Ethernet ports, serial and parallel ports, PS/2 keyboard and mouse ports, and even an infrared receptor (presumably for wireless keyboards that aren't supported), but it is missing some software. The most glaring omissions are Netscape and **gdb**, although I am told that Netscape should be ported soon and I suspect gdb was left out by mistake. Also, the NetWinder's distribution is an old Red Hat with FVWM95-2, and that's so passé these days. Debian works on the StrongARM and is now the number-one choice of hackers, so Rebel.com ought to modernize a bit. Anyhow, window managers and the Web enthuse me less than RISC chips, so I thought I would learn about the chip by using gdb, but that was not to be. However, I did find out about the chip through some e-mails and a process someone called “reverse engineering” (writing C programs, compiling with **gcc -S**, and deciphering the chip from there).

The StrongARM Architecture

If you haven't seen RISC assembly code before, it may be strange to see so many operands on a single line. After all, CISCers are used to simple syntax like MOV A,B (or MOV B,A). RISC technology, by comparison, allows you to specify several operands on a single line as well as communicate more than just the usual relative, absolute and immediate addressing information. Although many hackers these days avoid learning assembly code (ESR recommends C, Python, Perl and LISP in his advice on how to be a hacker), it is difficult to appreciate operating system design without such knowledge. Of course, *someone* has to know it or nothing could get implemented. When porting Linux to a new platform (such as StrongARM), assembly code is more important than is usually the case.

One fault of Linux that Tanenbaum criticized (not the least politely) back in 1992 was that Linux was platform-specific, that is, a monolithic kernel for Intel 386. Today, Linux has been ported to several hardware platforms, and the kernel is more modular. We need to be familiar enough with assembly code to be able to do the porting and debugging work, and we must not get too dependent on it, lest we write too much platform-specific code. The idea is, Linux is superior to any hardware platform, so we use a cross-platform assembler (meaning C) for the code and limit our involvement with the specifics of the chips. (This way, when the x86 dies, Linux lives.) Nevertheless, RISC chips are simple, with only a few instructions, yet one can do so much on a single instruction. RISC technology, particularly in the context of parallel processing, is affiliated with the microkernel, and hopefully we will see more of both in the future. In the meantime, let's get back to the chip.

The StrongARM SA-110 has the following features:

- 16 32-bit registers for user programs (r0 through r15)
- 21 basic opcode types, including 63 of what we would typically consider opcodes in the CISC world (which when multiplied by 15 conditionals would represent 945 instructions), with a host of other operations available
- one of fifteen conditionals available for every opcode
- 2.1 million transistors (RISC economizes on transistors, which is a bit ironic these days)
- 32KB of cache (16KB for instructions, 16KB for data)
- a 233 MHz operating speed, overclocked to 275 MHz on the NetWinder for 250MIPS (million instructions per second)

The SA-110 has no math coprocessor. The chip performs extremely well on tasks where the instructions and data can be entirely cached and no floating-point operations are involved. Tasks involving too much code and data to make use of the cache become slow, and floating-point operations grind the chip to a halt. While some of these differences have to do with how well the Linux kernel and gcc jive with the chip set, without an FPU, a chip is disadvantaged (but the StrongARM excels at floating-point emulation, if you code specifically for it). If you check out the benchmarks (see Table 1), you can see how scattered the results are. I would not rest too much on them; the machine is quite perky, despite its typically lower than K6/233 scores. My favorite benchmark, the chess test, has NetWinder evaluating between 17 and 19 thousand positions every second (about one million per minute). Rebel.com expects to have 600MHz StrongARMs soon, and if the micro-architecture techniques get seriously improved (low power consumption and desire to avoid pipeline stalls make speculative execution and multiple branch prediction respectively impractical, though unique qualities of ARM could make other tricks possible) and the

processor visits the debugger, future NetWinders should be much faster. (But then again, how much processor power do you need for e-mail, web, file, print, FTP and TELNET services?) The machine in question has 64MB of memory (34.8MB free), with most services turned on. As for disk access, Table 2 shows what Bonnie reports.

Table 1

Table 2

As you can see, the results are sporadic, with the processor scoring lower than a K6/233 in many cases and the disk drive operating much slower than a typical desktop hard drive. In spite of the low benchmarks, the machine has never lagged on me and does its tasks quite well. This is probably one case where the benchmarks don't mean very much, unless you plan on using the NetWinder for crunching numbers and maintaining huge databases. The chip set is fun to use, and I think hackers who are interested in the StrongARM might like to take a look at these machines, as well as anyone who wants a simple machine for use as a server.

A Different Chip Set

The joy of C is that we can bring programs written in it across hardware platforms without too much difficulty. This means Linux should be able to hop onto the 64-bit architectures fairly well. Desktop machines appear to be heading in the Alpha and Itanium direction (is anyone else frustrated by Intel architecture?), which bodes well for proponents of RISC. Ideally, we will be heading in a microkernel direction as well. What does this mean for the 32-bit StrongARM? Well, there's an active development community bringing Linux to StrongARM. At http://www.arm.uk.linux.org/, you can find the latest details, including screen shots of Enlightenment running on a NetWinder, along with support for Acorn, Archimedes, A5000 and various other ARM boxes. Intel makes StrongARM right now but is pushing the IA-64, while Compaq has recently been pointing out that Alphas are truly nice processors. It seems as though the Wintel stranglehold on operating systems has now been broken, and we are seeing more freedom in the chip area as well (even Intel finds itself with more freedom). As long as we keep Linux available across numerous chip sets (without network externalities favoring one too heavily over the other), we can offer a freedom to choose in the chip arena, which theoretically should get us on better chip sets at lower cost. As long as instruction sets stay free from patent and copyright restrictions, we chip users should expect more happiness. In any event, at least we're moving in a RISC direction, so the next logical steps would involve multiple processors on the hardware end and microkernel design in the software arena. The StrongARM itself seems destined largely for

embedded systems, but since embedded systems are a strong point of Linux (for one reason, we have no licensing issues), it's good to keep it supported.

If you get into StrongARM programming, you'll find yourself able to use four operands with one opcode (or more if you get into coprocessor instructions) and do things in a single line that would have taken several in x86 or other CISC chips. One thing I like is being able to perform a bit shift on the same line as a command. For example,

```
sub r1, r2, r3, lsl r4
```

would logically shift r3 to the left by the number in the lower byte of r4, subtract this from r2, and put the answer in r1. This is just one example of the efficiency of one command line in ARM (and other RISC) instruction sets. Additionally, all instructions are conditional (checked against the CPSR, current program status register) and you can specify one of 15 conditions for each instruction. Again, we're not supposed to get too dependent on assembly code, but at least this one is fun to use.

Improvements

No matter how Linux-based the NetWinder might be, it could stand a couple of improvements. For one, the graphics driver displays a VGA console with a 60Hz refresh rate, and that's too slow. Also, when scrolling quickly through a text file in vi, the NetWinder can skip a line; you might think you're editing one line, but when you save the file, you discover you've actually rewritten the line directly above. The perkiness and fast scrolling qualities are nice, so I hope there is a solution to the line-skipping problem other than slowing down the terminal. As for distributions, it has been mentioned that Debian would be preferred, and a modern window manager would be nice. I hope the newer StrongARMs get faster and their unique design doesn't make it too difficult to implement useful micro-architecture techniques which could accelerate the chip considerably. It's a nice design, and I wouldn't mind seeing these chips in 64-bit.

As for the software, criticisms of the UNIX model attack its accessibility rather than its performance. The neat thing about the NetWinder is that it brings accessibility to Linux networking. If free software developers looked at the NetWinder model and developed similar easy-to-use interfaces for configuring a network on generic Linux boxes, Linux might come across as less intimidating. The fact remains that computers *could* be of service even to people who don't want to devote their lives to them. Imagine if, without any knowledge of Linux or UNIX, you could use a web browser and a cable modem or DSL to have your own permanent net presence. At a click, you could turn services, firewalls, web pages, e-mail services, printers, etc. on and off. Essentially, this would have many of the practical rewards of learning UNIX,

without all the work. This is already available with the NetWinder on StrongARM architecture, but it won't become widespread until it's free. I would recommend that Rebel.com release the NetWinder configuration software under GPL (not a pseudo-free license); otherwise, we might want to consider using it as a model for a free source project.

Conclusion

As an office server, it performs flawlessly. The web-based interface makes Linux networking accessible to non-UNIX types. The functions and features are based on the Linux kernel, GNU software and various other projects such as Apache. (This is the “it's Linux, it has to be good” principle—true once again.) The benchmarks gave the NetWinder fairly low scores, but these are misleading. The NetWinder is an inexpensive solution to the server problem. If you have the technical know-how, you might be able to do better for the price. For example, you could build your own server and configure it yourself, delivering a faster machine without too much effort. Still, if you want the web-based interface, small size, low power consumption, ease of use and corporate support, you'll do fine with a NetWinder. Developers could really have fun with it, although I would recommend the Developer model for them, since it's even more fun. We still have to see if a good web browser gets ported to the NetWinder before we know whether it will be a viable platform for normal Linux use. Nevertheless, NetWinders may be only the beginning of clever uses for ARM-based Linux. Try to get your Windows-using acquaintances to buy one of these; it could be the start of something cool.

Jason Kroll (info@linuxjournal.com) is Technical Editor of *Linux Journal*. He is fond of animals, especially dogs, while he admires the independence and aloofness of cats which remind him of misfiring computer programs.

Archive Index Issue Table of Contents

Advanced search

ProConnect

Ralph Krause

Issue #71, March 2000

The ProConnect 4-Station CPU switch has a relatively small footprint, measuring just 10.25 x 7 x 3 in inches.

- Manufacturer: Linksys, Inc.
- URL: http://www.linksys.com/
- Price: $144.00 US
- Reviewer: Ralph Krause

After adding a second Linux box to my home network, I lost almost all of my work space. Juggling all those keyboards and mice was a hassle, not to mention trying to find enough electrical outlets. I had thought about getting a KVM (keyboard-video-mouse) switch, but held off getting one until a friend told me about his good experience with the Linksys ProConnect CPU switch.

I purchased the Linksys ProConnect 4-Station CPU switch, and currently use it to control two Linux PCs and one Windows PC. The switch works well, and I have been very satisfied with it.

The ProConnect 4-Station CPU switch has a relatively small footprint, measuring just 10.25 x 7 x 3 in inches. The CPU switch comes with a power adapter and must be plugged into a standard electrical outlet to work. The ProConnect CPU switch also comes with a small, but thorough, manual.

Several companies sell complete keyboard, mouse and monitor cable sets for KVM switches, so you can get all the cables you need for your systems in one convenient package. Be sure to check the mouse and keyboard ports of your PCs before buying the cables. You can use either a PS/2 or an AT-style keyboard, but you must use the same style mouse (PS/2 or AT) on the

ProConnect switch that your computers use. If you have a mix of PS/2 and AT-style mice, you will have to plug two mice (one PS/2 and one AT) into the mouse console ports of the ProConnect switch, according to the manual.

Setting up the ProConnect switch is straightforward. Plug your keyboard, monitor and mouse into the unit's console ports. Then connect the keyboard, monitor and mouse patch cables between the ProConnect switch and each of the PCs you wish to control. Finally, plug the ProConnect's power adapter into the wall, and turn it on.

The ProConnect CPU 4-Station switch has four red LEDs, four green LEDs and four pushbuttons in addition to the power switch on the front panel. The red LEDs indicate which PCs are powered up, and the green LEDs indicate which PC you are currently controlling.

Changing the current PC can be done through the keyboard or by pushing one of the buttons on the front panel. The keyboard sequence for switching PCs in the ProConnect manual is incorrect. The correct sequence is to press and release the **ALT** key, press and release the **CTRL** key, press and release the **SHIFT** key, then press and release the number (1-4) of the PC you want to use, and finally, press the **ENTER** key.

The ProConnect switch can also scan through all the PCs connected to it. To start scanning, enter the keyboard sequence above, but press 0 for the computer number. After the **ENTER** key is pressed, the ProConnect switch will cycle through all the running PCs. To stop this automatic scanning, press the space bar.

ProConnect CPU switches can also be connected to other ProConnect CPU switches to provide control to many PCs with one keyboard, mouse and monitor. The instruction manual provides the proper DIP switch settings and cable connections to use for cascading.

If you're using the CPU switch to control Windows PCs, make sure the Windows PC option is selected on the CPU switch while it is booting up, or its screen won't have the proper resolution when you change to it.

I have used my Linksys ProConnect switch for several months now, and am entirely satisfied with it. In addition to freeing up valuable desk space, the ProConnect switch makes it easy to use multiple PCs without the hassle of finding the right mouse and keyboard.

Ralph Krause is a freelance computer consultant in Michigan. His current career goal is to stay out of automotive paint shops. He can be reached via e-mail at rkrause@netperson.net.

Archive Index Issue Table of Contents

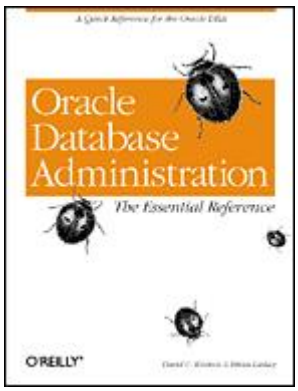
Advanced search

Oracle Database Administration

Charles Curley

Issue #71, March 2000

Oracle Database Administration is not your beginner's book on SQL or Oracle.



- Authors: David C. Kreines and Brian Laskey
- Publisher: O'Reilly & Associates
- E-mail: info@ora.com
- URL: http://www.ora.com/
- Price: $36.95 US
- ISBN: 1-56592-516-5
- Reviewer: Charles Curley

Oracle Database Administration is not your beginner's book on SQL or Oracle. If you want a beginner's book, get the third edition of *The Practical SQL Handbook* by Judith Bowman, et al. (Addison Wesley, 1998). This is also not an introduction to Oracle database administration. It is a handbook to have nearby if you are an Oracle database administrator (DBA) or are about to become one. Given the release of Oracle8 (covered by this book) for Linux, you may be in danger of the latter. The authors assume you already know more or less what you want to do, and they have made it easy to look up exactly how to do it. They are both Oracle DBAs of long standing, and both are active in Oracle user groups. The authors clearly write from experience.

The book is divided into two sections: the first, DBA Tasks; the second, a reference work for DBAs.

The first part's organization somewhat reflects the life of a database. In their introduction, the authors cover some theoretical ground, which you will find useful. The introduction ends with configuration planning. This section is one you probably won't need very often, but if you are ever in the position of setting up a new database or porting one to a new server, you should read this chapter. There is much good advice on a task you don't do often, but which is essential to making the database work correctly within your budget. The book then covers installation and performance tuning. Again, these two chapters cover things you don't do often, but which are important. Performance tuning is not the same as query optimization, covered in a later chapter.

Following this is a chapter on backups, essential for data integrity. The authors discuss the various types of backups one can do with Oracle or the underlying operating system, their advantages and disadvantages. Step-by-step procedures for several different types of backups and restorations are then given. As the authors say in the chapter, “The time to plan for recovery is before you need to recover data.” The authors recommend you walk through several of the backup/restoration procedures to practice and test the ones you will use.

Next are chapters on networking and security. The final chapter covers a number of Oracle tools and has a tantalizingly short section on how to find the many third-party tools for Oracle. Much of the rest of the book consists of reference pages for commands the Oracle DBA will find useful.

When you need more information on Oracle and Oracle administration, refer to the appendix, “Resources for the DBA”. This covers books, magazines, organizations (and their web sites), newsgroups and list servers. It is an excellent starting point, and in typical WWW fashion, will head you onward from resource to resource.

Finally, the index includes entries for the various Oracle-specific SQL commands and parameters. At a minimum, the index will be very useful for reverse-engineering some of those cryptic SQL statements that inadvertently get left around any database shop.

The language in the book is terse and concise, almost laconic, and very much to the point. On any question, the authors provide rules of thumb or points to consider, but leave the final decision up to the reader. Recommendations are also sprinkled throughout the first part of the book. The authors explain their recommendation, which means the reader learns the “why” as well as learning a good practice.

The first edition was printed in April 1999. By June, four errata were posted on the web page, and a second, corrected printing was due by the end of that month. Clearly, O'Reilly takes the professionalism and accuracy of their books seriously. O'Reilly's web site is one of the best around. Looking at a web page for a book is never as good as holding the dead trees in your hands in a bookstore, but O'Reilly's web site comes close. If you are not sure whether this book is for you, check out its entry at the web site. If you are an Oracle DBA or are about to become one, I highly recommend it.

Charles Curley (ccurley@trib.com) has been working with computers for twenty years. His present and former clients include Rockwell International, the Jet Propulsion Lab and Hughes Aircraft. He wrote this review using Emacs and ispell on Red Hat Linux 5.2. No windows were crashed in its production.

Archive Index Issue Table of Contents

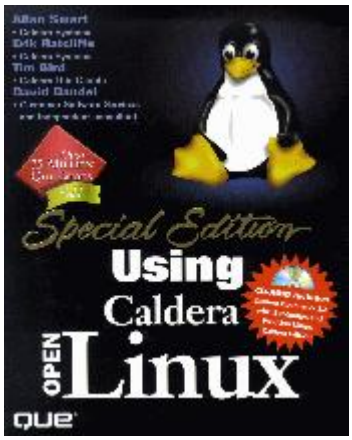
Advanced search

Using Caldera OpenLinux, Special Edition

Ben Crowder

Issue #71, March 2000

This 1200-page book covers almost everything you would need to know about Linux.



- Author: Allan Smart, Erik Ratcliffe, Tim Bird, David Bandel
- Publisher: Que
- E-mail: orders@mcp.com
- URL: http://www.mcp.com/publishers/que/
- Price: $39.99 US
- ISBN: 0-78-972058-2
- Reviewer Ben Crowder

If you've ever wanted a thick, massive reference to Linux—particularly for OpenLinux—then *Using Caldera OpenLinux* is the book for you. This 1200-page book covers almost everything you would need to know about Linux. Granted, in only 1200 pages there are *some* things that aren't covered, but this book does an excellent job on the things it *does* go over.

The first couple of chapters introduce Linux and OpenLinux and explain how to install OpenLinux (the book comes with an OpenLinux 2.2 CD). There are

distribution comparisons, an explanation of the Linux Standard Base project and a complete guide to LIZARD (the Caldera installation program).

The second section, “Using OpenLinux”, introduces KDE, shows you how to navigate the desktop and tweak KDE to your tastes, explains KDM and how to get it set up, briefly covers the hordes of applications that come with KDE and has a chapter on KOffice. In case you were wondering, there is virtually nothing on GNOME—but that makes sense, since Caldera's default desktop is KDE (and when you already have 1200 pages, you don't want yet another chapter). This is a marvelous introduction to KDE, one I would suggest for any KDE user.

“OpenLinux System Administration”, the third section, explains the Linux file-system structure, users, groups and permissions, DOSEMU, the boot process (inittab and friends), how to customize your shell environment (with a few pages on shell programming), printing, RPMs and other types of package management and how to build your own RPMs. Other topics include building your own kernel and the kernel modules, partitioning your hard drive, mounting and unmounting file systems and LILO. This section is for the most part true for all Linux distributions, not just OpenLinux. For example, take the chapters on recompiling your kernel; these apply to any Linux box.

Section four, “Networking with OpenLinux”, is the meat of the book. Three hundred pages are devoted to networking and rightly so, considering Linux is basically a networking operating system. There are chapters on TCP/IP fundamentals, network administration, IP aliasing, PPP, e-mail, BIND and DNS, FTP, Apache, IP masquerading and firewalling, TCP wrappers, NFS, NetWare, Samba and other Windows connectivity tools. If you want to learn Linux networking, you should definitely read this section. Even if you aren't using OpenLinux (like section three, this section applies to most Linux distributions), you'll find the information in these chapters highly relevant and useful.

There are a hundred pages on X: setting it up, the beautiful XF86Config file, customizing X and X resources. The final section is for miscellaneous topics, with two chapters on encryption and multimedia. The appendices include a list of commonly used commands, a hardware compatibility list, Linux module information and other Linux resources.

So is this book a must-buy? Yes, yes, yes, ten times over. I was very impressed with it, even though I'm not running OpenLinux (most of what I read, I was able to use on my Red Hat machines). This is one of the best Linux books—in fact, make that *computer* books—I've read in a long time. It's clear and concise, and (perhaps most importantly) humorous at appropriate points. It's geared more toward intermediate and advanced users, but beginners can learn much from it as well.

Ben Crowder is a young Linux aficionado living in Utah. In addition to fiddling with the insides of computers, Ben enjoys reading, writing and music. He can be reached at mlcrowd@enol.com.

Archive Index Issue Table of Contents

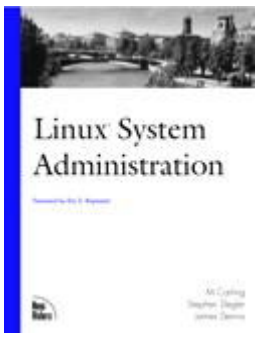
Advanced search

Linux System Administration

Paul Almquist

Issue #71, March 2000

Although it addresses some of the how-to's, it is more of a “why-to”.



- Authors: M. Carling, Stephen Degler, James T. Dennis
- Publisher: New Riders Publishing
- URL: http://www.mcp.com/publishers/new_riders/
- Price: $29.99 US
- ISBN: 0-56205-934-3
- Reviewer: Paul Almquist

Linux System Administration is not the average “how-to” for the system administrator. Although it addresses some of the how-to's, it is more of a “why-to”. The book is divided into 3 parts: Principles, Practicum and Services. Throughout, administration principles are emphasized, followed by practical suggestions.

Part I, Principles (chapters 1-7), starts out with a discussion of “The Prime Directive: to provide and maintain access to system resources.” In other words, why we are doing this job; not just because it's fun, although it should be most of the time. It goes back to the basic reason for the existence of computers—to provide a service to the users. “Requirements Analysis” is then examined to put some legs on the prime directive.

If you are responsible for your own Linux box at home or at work, you might let “Recovery Planning” slip, but when you are supporting the work of others in the enterprise, a recovery plan is essential. Computers will fail. Users will make mistakes. How will you respond? The authors present the interesting concept of a rescue partition and an auto-rescue configuration to automate recovery from some disasters, although they missed the opportunity to identify just what constitutes “a full suite of repair tools”. The experienced system administrator will know some, but the beginners need more guidance. Numerous recovery issues are discussed, including administrator error.

Too often we get caught up in a putting-out-files mode. The chapter on capacity planning gives guidance in preventing fires. Principles ends with a discussion of the Help Desk, Network Computing and Security Policy.

Part II, Practicum (chapters 8-16), begins with the topic of enforcing security. Certainly not an exhaustive treatment, considering another recently published book has several hundred pages on just this one subject, but fundamental topics are addressed. Exploits and their consequences should have you thinking of your recovery planning again. This chapter and most of the following chapters end with a reading list citing relevant HOW-TOs and many other books. Booting and shutdown reviews LILO, MILO and SILO and the System V startup/shutdown rc scripts. If you need to build a custom kernel, the chapter entitled “Configuring and Building Kernels” gives some helpful hints. Everyone finds some tasks repetitive. The chapter on scripting offers ideas in several scripting languages.

There are two kinds of computer users: those who do backup, and those who will. The Backup chapter discusses practical solutions to the problem. Most computers are on a network today. Networking Technologies discusses IPv4, IPv6 and related protocols. This chapter has an extensive reading list.

Do you have several systems to support? Wish they all had the same configuration? System Profiles gives good guidance here. This chapter has the best discussion of disk partitioning theory I have read—not just a set of recommended sizes, but why various branches ought to be separated into their own partitions. Once you have a partitioning plan, the OS has to be installed on all machines and upgraded as necessary. Automated Installation and Upgrade discusses some how-to guidelines, including Red Hat's **kickstart**. User and Group Administration appropriately follows.

Part III, Services (chapters 17-21), has 40 pages reviewing network services, mail, printing and faxing, working with NFS and managing NIS. Little nuggets of information are sprinkled throughout.

Appendix A is something we hope we never need, but know we must deal with: emergencies. It starts with the lost root password and ends with some miscellaneous performance problems. Appendix A is followed by a glossary, worth reading in that it's more of a mini-encyclopedia than a dictionary. The index is huge for a book this size (41 pages long), making it a valuable tool for using this book as a reference.

Eric S. Raymond, writer of the book's Forward, says in a review on its back cover, “This book is a reference for the experienced Linux administrators and a guide for new ones.” I heartily agree and recommend a thorough reading.

Paul Almquist (almquipf@cis.chippewa.tec.wi.us) has been teaching all aspects of software-side computer technology for over 30 years. He uses Linux on the desktop, laptop and server at work and home. With discarded parts, he built his own mini-Beowulf system with 8-486/66 CPUs. In addition to teaching, Paul administers all of the UNIX/Linux systems and is webmaster for the Computer Information Systems Department at Chippewa Valley Technical College.

Archive Index Issue Table of Contents

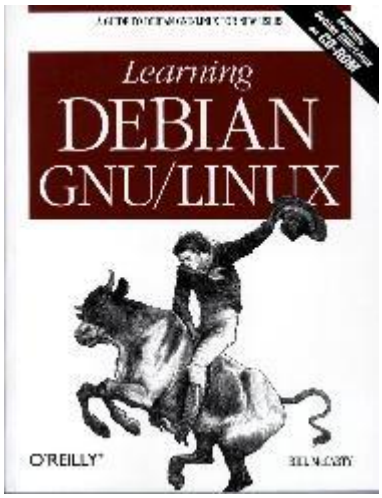
Advanced search

Learning Debian GNU/Linux

Marjorie Richardson

Issue #71, March 2000

This book contains a goodly amount of information, and not all of it is for the newbie.



- Author: Bill McCarty
- Publisher: O'Reilly & Associates
- E-mail: info@ora.com
- URL: http://www.ora.com/
- Price: $34.95 US
- ISBN: 1-56592-705-2
- Reviewer: Marjorie Richardson

Like most O'Reilly books, *Learning Debian GNU/Linux* is well-organized and well-written. The author, Bill McCarty, is an associate professor of computer science and obviously knows his stuff. The layout is well-done, and not filled with cutesy graphics to distract the eye.

Many books geared for the Linux newbie have come out in the past year—most of them try to stay vendor-neutral. This book follows the recent trend of

focusing on one distribution, in this case, Debian. I actually like this trend, as many newbies are confused when presented with variations in the different distributions, so sticking to one should make discussion clearer and shorten the learning curve for some. Also, Debian is not the easiest of distributions to install and use, so a book devoted to it is deserved.

This book contains a goodly amount of information, and not all of it is for the newbie. It is truly a guide to the entire system, not just an introduction. While introductory material such as a history of Linux, installing, configuring, definitions, etc. is presented, much more complex operations, such as setting up both a local (LAN) and a wide-area (WAN) network, are also described.

The material is presented in a straightforward manner, using clear, easy-to-understand language. However, I did feel the book was written with the computer-literate reader in mind. Even though he defines basics such as “what is an operating system”, it still reads as if he expects you to already know. Indeed, he tells you how to get information you will need to install Linux by using your MS Windows menus.

Mr. McCarty devotes entire chapters to user administration and X configuration, as well as networking. His desktop of choice is GNOME, and it too receives a chapter. Also included are chapters on games and connecting to the Internet. The appendices include a Linux directory tree, the principal Linux files, Debian utilities and a Linux Command Quick Reference. A CD-ROM containing the Debian 2.1 distribution is bound into the book.

The book is not without its faults. One amusing mishap is in Chapter 4. An introduction to useful Linux programs states that you will learn the simple text editor **pico**, then the author proceeds to teach you the simple text editor **ae**. Guess he changed his mind.

Overall, *Learning Debian GNU/Linux* is a complete reference guide to this distribution, and well worth adding to your collection of Linux books.

Marjorie Richardson is Editor in Chief of *Linux Journal*. She loves her job and the fact that she gets free t-shirts to wear to work. She can be reached via e-mail at info@linuxjournal.com.com.

Archive Index Issue Table of Contents

Advanced search

Shell Functions and Path Variables, Part 1

Stephen Collyer

Issue #71, March 2000

Shell utilities can simplify the maintenance of your path variables.

Few UNIX users give much consideration to their path variables. They are typically used in a “set and forget” fashion, and consequently, they often end up like straggly weeds, overgrown and unlovely. Take a look at this mess:

```
$ echo $PATH
/opt/kde/bin:/localbin:/usr/local/bin:/bin:/usr/bin:
/usr/X11R6/bin:/home/stephen
/scripts:/home/stephen/bin:/opt/CC/test/bin:/usr/sbin:
/usr/bin/X11:/ora01/app
/oracle/product/7.3.2/bin:/scripts:/opt/CC/bin:/bin:/usr/bin
```

Given this undifferentiated stream of characters, how long will it take you to:

- List all the bin directories in PATH? (**grep** won't help—try it)
- Swap the order of /bin and /usr/bin?
- Remove that pesky /opt/CC/test/bin directory?
- Get rid of the duplicate directories?

Despite their apparent simplicity, path variables can be tricky beasts to manipulate. It's all too easy to end up with duplicate entries in a path, and even the act of checking the contents is not straightforward. Adding a new directory to PATH is easy enough, but even then you may end up with a duplicate, because your eyes don't parse a colon-separated list efficiently.

A path variable is any shell or environment variable comprised of textual elements separated by colons. You are almost certainly familiar with the so-called search path, **PATH**, which your shell uses to find executable files, but there are other standard paths, such as **MANPATH**, which the man program uses to locate man pages, and **LD_LIBRARY_PATH**, which the dynamic loader can use to find shared libraries.

Path variables consist of textual elements separated by colons, and the (admittedly non-standard) term I use for these is “path element” or simply “pathel”. (You'll also see the term “path prefix” used, but not by me.) I'll also abbreviate “path variable” to “pathvar”.

All the utilities I describe here assume the bash shell (though there are Korn shell versions available as well), and they have been tested using bash 1.14.7 and bash 2.03.4.

I assume you know how to set and access variables in a shell and have used (or seen) shell control constructs (if, for and while). I also assume you are not necessarily clear about shell variables versus environment variables, or shell scripts versus shell functions, and specifically, that you have no idea what **eval** does.

Utilities

Here's a brief description of some path-variable utilities:

- **addpath**: adds a pathel to a pathvar only if the pathel cannot be found on the pathvar (e.g., **addpath -p NEWP*/abc/***).
- **delpath**: removes pathels from a pathvar (e.g., **delpath -p NEWP */abc/***).
- **edpath**: allows editing, and thus arbitrary modifications, of a pathvar.
- **listpath**: echoes the pathels of a pathvar on separate lines; the output can then be filtered using grep, for example.
- **uniqpath**: removes duplicate pathels from a pathvar.

A good shell utility should provide some guidance to the user, and accordingly, each pathvar utility has a **-h** option, which writes usage information to standard output. Furthermore, a good utility should not be fragile; it should check its arguments for sanity, as far as possible. This is doubly important when an important variable such as PATH is being altered. The path utilities share common option-handling code to simplify this sanity checking.

Taming Options and Arguments

Traditionally, shell scripts have handled their options in a somewhat ad hoc manner. The option-handling code in a script will often comprise a hand-crafted loop around **getopts** (which I'll describe later); this loop sets variables and issues error statements, as appropriate to the requested options. While this approach is common, it requires duplication of code in every script that is written. This is tedious and error-prone.

Listing 1

Option-handling code generally performs a small set of functions (i.e., setting variables and issuing messages), so we can usefully write a shell function to standardize this behaviour. Take a look at Listing 1, a shell script called **testoptions**.

To run this script, we could make the file executable by the owner (**chmod u+x testoptions**) and type its name. If you do that, you should see something like this:

```
$ testoptions
./testoptions: options: command not found
```

This occurs because line 3 of the script refers to **options**, a shell function, which we haven't told the shell about yet. When we do so, we can run testoptions again, this time with some arguments:

```
$ testoptions -a -b fred -d
opt_a=1
opt_b=fred
opt_c=
options_missing_arg=
options_unknown_option=d
options_num_args_left=0
```

Now, the shell function options has looked at its first argument ("ab:c"), a coded specification of the name and type of the expected options. It uses this to interpret its remaining arguments, which in this case are all those originally passed to testoptions (i.e., **-a -b**, **fred** and **-d** because **$@** is converted into a quoted list of all arguments to the script).

The argument specification (ab:c) is in the form expected by the **getopts** command and means "we take three options, **-a**, **-b** and **-c**, and **-b** requires an argument". The fact that **-b** requires an argument is indicated by the colon.

Each time the options function sees one of the allowed options in its argument list, it creates a new shell variable indicating the argument was present. So, for example, when the second argument (**-a**) is examined, options creates a variable called **opt_a** and sets its value to 1. Similarly, if an illegal option is passed, options creates a variable called **options_unknown_option** and sets its value to the name of the illegal option. As you can see from the output shown above, if an option requires an argument, the supplied argument is used as the value of the new variable. (Perl scripters will recognize this behaviour from the Getopts modules, which were, in fact, the inspiration for options.)

Listing 2

The fundamental problem is that options can't know in advance which variable names it will have to create, so they can't simply be hard coded in some way (at

least not efficiently). Listing 2 is the code for options. The first couple of lines inform the shell that what follows is a shell function. A shell function is a collection of commands in a file that can be run by typing that name (i.e., typing **options** in a shell runs the commands in that script) and that run in the context of the shell calling the function. This last part is important; that is, when a shell runs a function, its commands take effect in that shell, in the same way as commands typed on the command line of an interactive shell. You should compare this to the effect of commands executed in a shell script, where a *new* shell is created to run the commands. For example, if you execute the command **cd** in a shell function, the current directory will be altered; in a shell script, the cd will take effect only in the new shell created by running the script. When the script has finished, you'll be in the same directory as before you ran it. Shell functions also have numbered arguments (i.e., **$1**, **$2**, etc.) just like scripts.

The next part of options performs some initializations. The first six executable lines declare variables. Since code in a function executes as if it was run in the calling shell, if we create a variable in a function, it will exist in the shell at the end of the function. If we don't want this behavior, we can make a variable local to a function by preceding it with the reserved word **typeset**. (In bash, you can use **local** instead, but **typeset** works in ksh, too.) Thus, the variable **opts** will not exist at the end of options, but **options_shift_val** will, for example.

After checking the number of arguments, we set **opts** to the value of the argument **spec**, with an additional leading colon. So, with our testoptions values, **opts** would contain **:ab:c**. The leading colon prevents getopts from issuing spurious error messages. The first argument is then shifted away by the **shift** command. This means the argument that was **$2** becomes **$1**, **$3** becomes **$2**, and so on. This is a common trick in shell scripting, used when an argument is no longer needed.

The meat of the function begins with the line **OPTERR=0**. This code section does the work of examining the options and creating the variables. We delegate the option examination to getopts and create variables using eval.

The shell command **getopts** examines the positional parameters (**$1**, **$2**, etc.). When you call it the first time, it examines **$1**; the next time **$2**, and so on. When called in a while loop as in Listing 2, it will look at all positional parameters and return **false** when finished, thus terminating the loop. Remember, options expects its first argument to be the getopts specification, and the remaining arguments to be positional parameters. However, we shifted the getopts specification away, so **$1**, **$2** and so on are indeed the positional parameters when getopts examines them. The **$opts** argument to getopts tells it the legal set of arguments, as described above.

If getopts sees a legal option, it stores it without the leading **-** in the **argname** variable, and if that option takes an argument, it stores that argument in a variable called **OPTARG**. If an incorrect option is seen, getopts stores an error code in **argname** and the name of the incorrect option in **OPTARG**. There are two sorts of incorrect options:

- An option in which the name is not listed in the getopts specification. In this case, getopts stores **?** in **argname**.
- An option requiring an argument, but where the argument is missing; getopts stores **:** in **argname** if this occurs.

bash getopts has a bug: it stores **?** in both these cases. Listing 2 contains a workaround. **ksh** does not have this problem.

If neither of these problems occurs, we have a valid option and can go on to create a variable. This is done in the final **if** statement in the loop. The **then** branch handles the case when the option has an argument and the **else** branch handles the case when there is no argument; both use eval. Let's look at one of these:

```
eval opt_$argname=$OPTARG  # set option name
```

Suppose we're processing the **-b** option with an argument of **fred**: **argname** will contain **b** and **OPTARG** will contain **fred**. We want the shell to run this code:

```
opt_b=fred
```

Our first attempt is likely to be:

```
opt_$argname=$OPTARG
```

reasoning that the shell will replace **$argname** by **b**, **$OPTARG** by **fred**, and we're done. Good try, but it doesn't work. If you're sitting in front of a bash shell prompt now, try this:

```
$ argname=b
$ OPTARG=fred
$ opt_$argname=$OPTARG
```

You should see this message: **bash: opt_b=fred: command not found**.

Which command is not found? The shell did indeed expand the variables. The problem is that, although the shell has generated the string **opt_b=fred**, it considers its work on the line to be finished and tries to execute a program called “opt_b=fred”. Although the line after processing looks like a shell command, the shell won't notice this, because it processes each line only once. To fix this problem, we need to instruct the shell to expand the variables in this line and then start over again, processing the line as if it is the first time. That is

precisely what the eval at the start of the line accomplishes. When the shell processes an expanded line, it will recognize eval as a command to create a shell variable and will create one.

Remember, these variables are being created in a function and will continue to exist after the function has terminated. Thus, we can call the options function from a script (or indeed another function) and use the variables it has created in any way we like.

In the interest of space, I have not described all the steps the shell performs when it expands a command line; for the gory details, consult *Learning the Bash Shell* from O'Reilly & Associates.

To ensure the shell knows about a function, there is one option in bash and two in ksh. In bash, you must “source” the file containing the function in one of your start-up scripts such as .bash_profile (or equivalently, include the code directly in the start-up script). In ksh, too, you can source a file in a start-up script, or alternatively, put your function files in a directory (perhaps called $HOME/functions) and add this directory to the **FPATH** environment variable. When you type the name of a command unknown to ksh, it looks in the directories in **FPATH** to see if there is a function file with that name. If so, it reads the file, remembers the function definition and executes it.

Stephen Collyer (stephen@twocats.demon.co.uk) is a freelance software developer working in the UK. His interests include scripting languages, distributed and thread-based systems, and finding out why upgrading to Red Hat 6.0 broke pppd on his Linux box. In his spare time, he campaigns against the British Government's IR35 proposals. Occasionally, he finds the time to talk to his wife and two remarkably attractive and highly intelligent children.

Archive Index Issue Table of Contents

Advanced search

Sending Files by E-mail

Marjorie Richardson

Issue #71, March 2000

When it's time to send those source files to your project leader, Linux has the commands you need.

Often, we have files we wish to send colleagues and friends by e-mail. Depending on the type of file you wish to send, there is a Linux command to make the process easy for you.

Plaintext Files

The shell archive file command **shar** is very useful for combining plaintext files into one file, which can then be sent by e-mail. **shar** writes shell commands for unpacking the output file inside the file itself, as well as code to verify the content using an MD5 checksum. It is much easier to use than the archive command **tar**, whose complexities can trip up even an expert. All you have to do to run shar is type:

```
shar input_filenames > filename.shar
```

It will take all the input files and write them to the output file, embedding the unpacking commands as it goes. The extension conventionally used with shar output files is .shar. To unpack the file, you don't need to know the embedded commands or retrieve them from the file yourself. Instead, use the command **sh**. The first argument to uuencode is the input file name; the second is the name to put in the output file header. We will use the same name for both. Type:

```
sh filename.shar
```

The sh command will do the work for you by reading the file, extracting the commands and executing them to unpack the archive.

It is permissible to use wild cards in naming your shar input files, so you can easily pack all the files in one or more directories. Just remember, the files must all be plaintext—no binaries.

The purpose of the shar command is quite similar to that of tar. However, the command and its format are much easier to remember, and since the output is plaintext instead of binary, it does not need to be encoded to send it as e-mail.

Binaries and Graphics

Non-plaintext files such as binaries and graphics must be encoded before e-mailing them. First, pack and compress your files using tar, which outputs a binary file. We won't go into tar in detail here, since it has been discussed before: see “Tar and Taper for Linux” by Yusuf Nagree, *Linux Journal*, February 1996 (http://www.linuxjournal.com/lj-issues/issue22/1216.html). A simple command to pack and compress all the files in the current directory would look like this:

```
tar -cvzf tar_files.tgz *
```

Now, the tar file can be encoded using the command **uuencode** by typing:

```
uuencode tar_file.tgz tar_file.tgz > mail_file
```

The output file can now be put into e-mail and sent off into the ether.

The recipient can save the e-mail (perhaps as mail.save) and decode the file easily—no need to even remove mail headers. Simply use the command **uudecode** by typing:

```
uudecode mail.save
```

uudecode will do just what its name says: decode the mail file, leaving a file called *tar_file*.tgz in the current directory. To unpack the tar file, type:

```
tar -xvzf tar_file.tgz
```

A single graphics file can be encoded and decoded in the same way, skipping the tar step since an archive file is not needed.

Conclusion

No matter what type of files you have, sending them to others is not a chore using these simple Linux commands.

Marjorie Richardson is Editor in Chief of *Linux Journal*. She has been a programmer for many years and enjoys her role as Linux advocate very much.

Archive Index Issue Table of Contents

Advanced search

Account Administration for K-12 School Systems

Steve Tonnesen

Issue #71, March 2000

Taking care of computers in schools can present special problems; this program solves them.

K12Admin is an account administration system designed specifically for K-12 school systems. It can be used to administer accounts on Linux servers in individual schools from a central location. The interface to the system is web-based. A teacher in a school who has been granted account-administration rights will be able to create student and staff accounts, delete accounts, create/delete local groups, modify the membership of local and global groups and change passwords within that one school.

History

K12Admin was developed at Coast Mountains School District in British Columbia, Canada. It was originally used to administer the accounts in an NT domain. We needed a method for allowing staff in one school to create accounts for their school without accidentally messing with accounts from other schools. The web interface and database ran on a Linux machine, and account changes were made on the NT Primary Domain Controller using a TCP/IP connection. Soon after this, we set up a Linux server in each town to use as an authenticating proxy server. The Squid logs on each of these servers were copied to the main account administration computer and analyzed for potentially inappropriate use by scanning for a list of keywords in the URL.

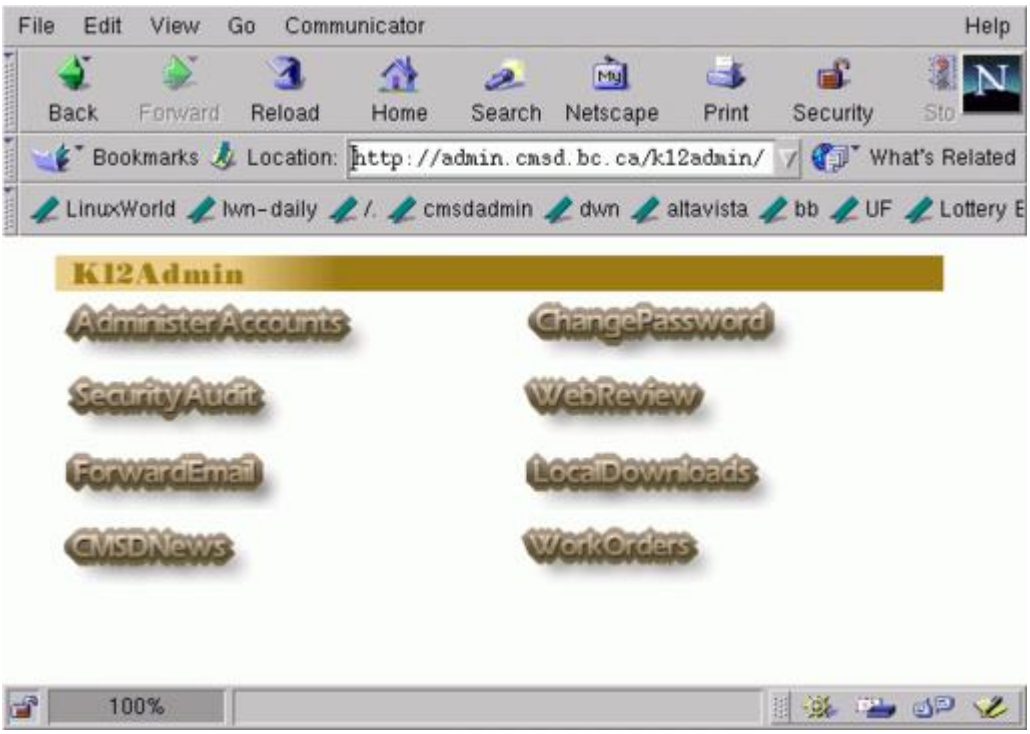


Figure 1. Main K12Admin Menu

Many accounts had already been created on the NT system before we set up the Linux server for account administration, so we did not have UNIX passwords for these accounts. To solve this problem, we modified the source for the Squid proxy server so that it redirected users to a page where they could “register” their account if proxy authentication failed. The user name and password they entered here were checked against a POP server running on an NT server before being added to the password database on the Linux servers.

Once we had UNIX passwords for all users, it became possible to use Linux servers in the schools.

Creating Accounts

When accounts are created, a list of first and last names is typed or pasted into a text box. Unique user IDs are created using the first and last names of the users. There are currently two different user ID creation schemes in K12Admin. One uses part of the first name and all of the last name. Under this scheme, my user ID might be stonnesen, sttonnesen, stetonnesen, etc. until a unique user ID was found. The second scheme uses the first name and the initial letter of the last name. Under this scheme, my user ID might be stevet, stevet2 or stevet3 until a unique user ID was found. This second scheme makes for a more anonymous user ID if protection of children's privacy is a concern in your district.

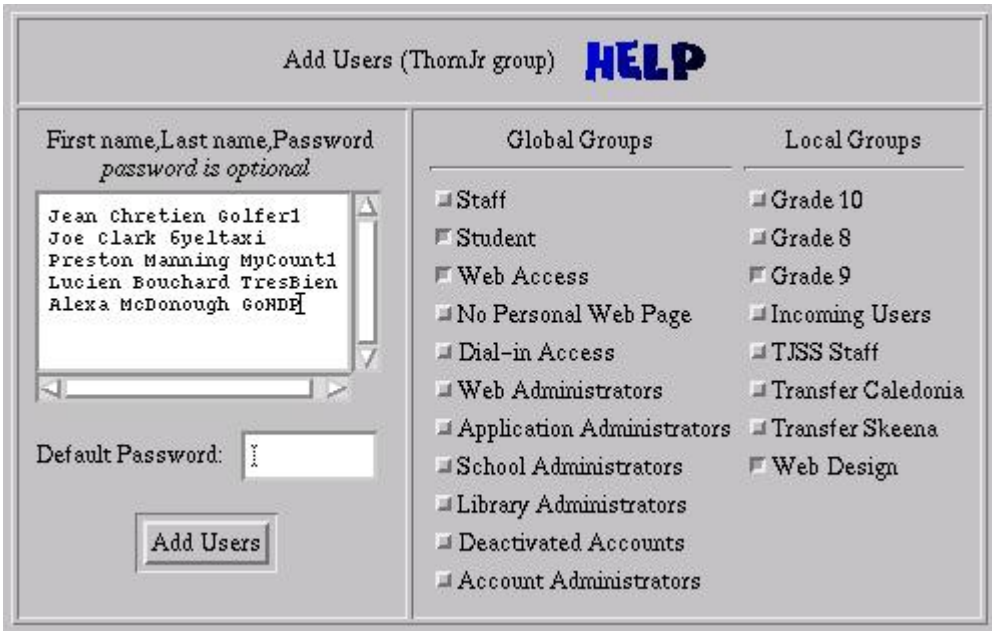


Figure 2. Form for Creating New Accounts

When accounts are created, each user ID can be assigned an individual password by the administrator, all accounts being created can be assigned the same default password, or random passwords can be generated for each account. Currently, the random passwords generated consist of a three-letter word, a single digit, plus another three-letter word (e.g., far6yet). With the word list I'm using, this results in a password space of just under 2,000,000 passwords. Not exactly the best security you could ask for, especially since my list of three-letter words is now quite public. I am considering adding an option to generate truly random passwords (such as “a5Tr43Zp”), but these are quite difficult for students to remember. It might be better just to instruct students (and staff) on good password-selection techniques and get them to change their assigned passwords right away. K12Admin has system-wide global groups that grant special-access privileges to users:

- *Web Users*: members can authenticate with a proxy server to access the Web.
- *Dial-in Users*: members can access dial-up pools.
- *Account Administrators*: members can administer accounts in their school.
- *Application Administrators*: members have read/write access to application shares on their school server.
- *Library Administrators*: members have read/write access to a Library software share on their school server.
- *School Administrators*: members have access to a share for school administrative software.

Figure 3. Account Administration Menu As Seen by School Administrator

- *Web Administrators*: members have access to the root directory of the school's web server.
- *Domain Administrators*: members have access to all schools in the system and can set system-wide configuration options.

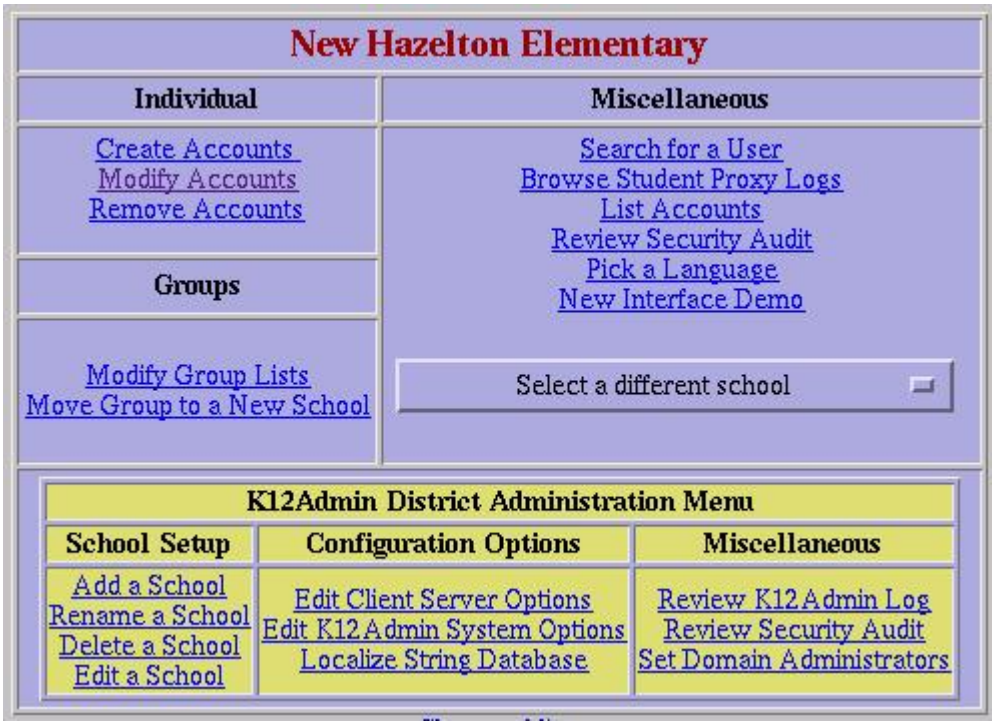


Figure 4. Account administration Menu As Seen by District Administrator

In addition to the global groups, local groups can be created which are unique to each school. These can be used to create class groups (Grade 7, Grade 4/5, Division 14, etc.) or other logical groupings (Yearbook Club, Computer Club, etc.).

Modifying Accounts

Accounts can be modified by account administrators or by users. The users can change their passwords, review a security audit of their account, or review potentially inappropriate web use by their account. Account administrators can change passwords and add/remove users to/from groups.

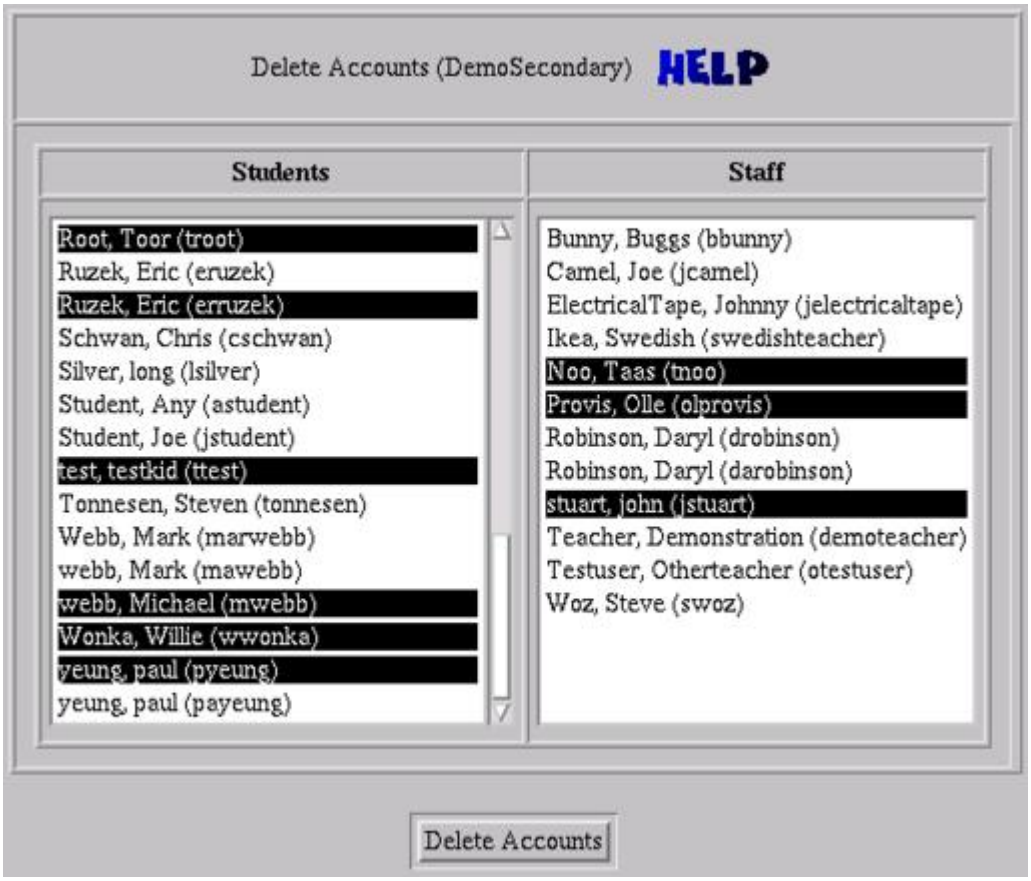


Figure 5. Form for Removing Accounts

Updating Clients

In the rest of this document, I will refer to three different *types* of computers. The *k12admin-server* will be the central account administration computer for the district. The *k12admin-client* computers will typically be Linux servers in the schools that update their account information from the k12admin-server computer. These computers will likely be used as web proxies, file servers, web servers, e-mail servers, etc. Lastly, I will refer to the computers our users sit at as *workstations*.

K12admin-client computers run updates once an hour. The updates are done using **rsync** and **ssh** to securely and quickly copy files from the k12admin-server and run a script, called “setup”, which is part of these files. Password files (passwd/group, shadow/gshadow, smbpasswd, squid.auth, apache.auth) on the client are updated at this time. The hourly update also runs some audits of the school server. Specifically, it scans the log files and generates a synopsis of which accounts have been used and when. This information is sent back to the k12admin-server computer, where it is inserted into an audit database which can be queried by the user. This allows a user to determine if their account is being used inappropriately.

Security Audit for 'demoteacher'

Site	Service	Most Recent Use	Number of Uses
copper-mtn	Apple Login	Mon Jun 14 15:17:25 1999	43
cormorant	Windows Login	Mon Jun 14 10:31:16 1999	14
demoserver	Apple Login	Mon Jun 14 13:00:06 1999	136
demoserver	Windows Login	Mon Jun 7 14:54:36 1999	4
hazlinux	Apple Login	Thu Apr 1 11:25:32 1999	4
hazlinux	Windows Login	Tue Jun 15 08:31:47 1999	58
hazlinux	Web Proxy	Sat May 29 22:44:17 1999	195

Figure 6. Audit of Account Usage for User "demoteacher"

The Squid proxy log files on the k12admin-client computer are also scanned for potential inappropriate use. The log files are scanned for one of a list of keywords. These keywords can be grouped into different categories (porn, chat rooms, gaming, executable downloads) and enabled or disabled on a per-school or per-server basis. Again, this information is transferred to a database on the k12admin-server where it can be reviewed by school administrative staff at any time. A synopsis is also e-mailed to school administrative staff weekly.

Individual k12admin-client servers can be configured from the web page with a variety of options. They can be attached to one or more schools. You can also have more than one server attached to the same school. This allows you to have one server for two small schools connected by Ethernet, or have several servers in a large school. Servers can be configured to update passwd/group files, update smbpasswd files, update Squid-authentication files, update Apache-authentication files, create home directories for new accounts, and clean up old home directories, either by deleting them outright or moving them to a temporary holding area.

Planned Improvements

The K12Admin system is quite functional as it stands. However, my goal is to create a server appliance which can be placed in a school and administered entirely through the K12Admin interface. What I see happening in schools now is that a teacher is given "release" time to maintain the network in the school. Attempts to "homogenize" the network are difficult when teachers in individual schools have invested significant amounts of time in learning how to run their chosen network operating system and setting up their network. In order to make K12Admin a viable alternative, I see a few obstacles to overcome:

- Server administration must be easy to learn. Since K12Admin is geared specifically to a K-12 setting, it should have an advantage over a vanilla, out-of-the-box, network operating system.

- The servers must be flexible enough to serve the needs of all users in a K-12 setting. Since I have experience with only one school district's way of doing things, I need feedback from other users to determine which options are required to meet their needs.
- When problems do occur, they should be easy to diagnose, both for technical staff and the users in the school. I have some problem-diagnosis tools in place for monitoring disk space, swap space and “stuck” printer spools. These tools need to be expanded. I was hoping to incorporate the “Big Brother” network monitor to take care of some of this, but the license is too restrictive.

Here is a list of planned additions to K12Admin:

- Creation of default shares on the client servers (exported by Samba, Netatalk, NFS, Coda, etc.)
- Applications: general applications used by client computers, be they Macintosh, Windows, Linux or whatever. There will, of course, be separate shares for each type of client computer.
- Library: share for library administration software.
- Administration: share for school administration software.
- Localgroups: share holds a directory for each local group created within the school, accessible only by members of that group. This share will contain a class folder for which teachers have write access, a drop box, and a public folder to which everyone can write. This makes it easy to share files with the members of your class or some other logical group.
- HTML: web server root directory.
- CD: a share where CDs can be copied in order to be shared back to client computers.
- Default: skeleton files for creating new user home directories are placed here.
- Rebuild: share for storing images of client workstations for rebuilding purposes.
- More server configuration options, such as the ability to configure the services which should run on a particular server (i.e., dhcpd, Samba, Netatalk, Apache, Squid, etc.).
- Monitor the status of services that have been configured to run on each server. Possibly integrate the “Big Brother” network monitoring tool, if its license allows.
- Integration with Bruno Vernier's EDUML standard.
- Possible integration with the Roster project. Roster is a server configuration system designed for college/university applications. It contains methods for updating server types other than Linux.

- Modularize (OOP!) the data layer (Roster has this already and might be usable).
- Extend the Squid proxy scan to scan the HTML files in the Squid cache, matching files to URLs using the Squid logs.
- Ability to add users to multiple schools. Useful for staff members who teach part time in two schools, or district staff who may work in all schools in the district (this latter case is a special one that should probably be handled differently).
- Generic configuration of system files such as Samba configuration, Squid configuration, Netatalk configuration, network configuration (dhcpd, IP masquerading, etc., using private IP addresses).
- Support for having "backup" k12admin-server machines that synchronize their databases with the main k12admin-server and can be used for automatic failover protection.

Visit http://k12admin.cmsd.bc.ca/ for more information on the K12Admin system or to download and give it a try. A mailing list has been set up for discussion of the issues surrounding K12Admin. To subscribe, send a message to majordomo@terlinux.cmsd.bc.ca with "subscribe k12linux-admin" in the body of the message.

Installation Instructions

Packages are provided for installing on Debian and Red Hat systems. There is also a tar archive file for installing on other systems.

There are two packages: k12admin-server and k12admin-client. The k12admin-server package should be installed only on the computer that will be the main account administration computer for your district. The k12admin-client package should be installed on every machine that will be updated from the k12admin-server machine.

Installing k12admin-server

The k12admin-server package contains the files necessary to use a machine as the central account administration server. Typically, you will want k12admin-server installed on only one machine in your district. It is possible to install the k12admin-client package on the same machine as the k12admin-server package, although for security reasons this may not be desirable. It would be better if the accounts on the server computer were administered manually, so that students and staff members do not have accounts on the server and are less likely to try mucking around there.

Install the k12admin-server package by running

```
dpkg -i k12admin-server*deb
```

or

```
rpm -i k12admin-server*rpm
```

(use **-U** in place of **-i** if this is an upgrade) in the directory where you have stored the package. If this is a first install, you will be told to run **k12admin-server.setup** as root to create the MySQL database. This file was placed in the /usr/bin/ directory when the package was installed, so it should be in your path.

If you are installing the package from a tar file, unpack the archive, go (**cd**) to the k12admin-server* directory and run **make install** to install the package.

In order to create the MySQL database, you will need to enter the root password of your MySQL server. Note that the MySQL root password is *not* the same as your normal root password. You should have been prompted to enter a password for your MySQL server when that package was installed. The script will allow you to keep trying passwords until it succeeds. Just press **ENTER** if your MySQL root password is blank. If this is the case, the k12admin-server package will prompt you for a new password, as it is a grave security risk to have a blank MySQL password.

The second password asked for by the k12admin-server.setup is one for the k12 MySQL user. This user is used by the scripts to connect to the MySQL database. You do not need to memorize this password, as it is stored in the /etc/k12admin.MySQL.pass file which is readable by the k12admin and www-data users only. You can change this password at any time by rerunning the k12admin-server.setup program.

Now you must configure Apache so it knows where the k12admin files are located. Once Apache is configured and reloaded, you should be able to access the account administration system at `http://yourservermachine/k12admin/` from any web browser. Log in as user *demoteacher* with *demopass* as the password.

Apache Configuration

I strongly recommend using **apache-ssl** (http://www.apache-ssl.org/) in order to encrypt packets between your web browser and the k12admin-server. This is especially true if there is potential for someone to be sniffing packets that are being transmitted. The basic authentication that is part of the HTTP standard is not encrypted, and your password can be grabbed easily off every outgoing web request if you are not using a secure server.

You must edit the Apache configuration files to enable the account-administration system. Add the following lines to the bottom of Apache's access.conf file:

```
Alias /k12admin/ /var/k12admin/web/
     ScriptAlias /k12admin-cgi/ /var/k12admin/webscripts/
     <Directory /var/k12admin/>
     AllowOverride AuthConfig
     </Directory>
```

You might also have to change the user and group of the Apache web server process. It may be set to “nobody” by default. The web server process must have access to the database containing sensitive account information. The password for accessing the database is stored in /etc/k12admin.MySQL.pass and is readable by only the www-data user and the k12admin group. It is, therefore, necessary to have the Apache process running as www-data. This account was created when k12admin-server was installed, if it didn't already exist.

To set the user and group of the Apache process, change the following lines in Apache's httpd.conf file:

```
User www-data
Group www-data
```

Reload Apache after making these changes. On Debian systems, reload Apache by running

```
/etc/init.d/apache reload
```

or

```
/etc/init.d/apache-ssl reload
```

On Red Hat, run

```
/etc/rc.d/init.d/httpd reload
```

Installing k12admin-client

The k12admin-client package contains the files necessary to update a machine using information from the k12admin-server machine.

Install the k12admin-client package by running:

```
dpkg -i k12admin-server*deb
```

or

```
rpm -i k12admin-server*deb
```

(use **-U** instead of **-i** if you are upgrading) in the directory where you have the package. If this is an initial install, you will be asked to run **k12admin-client.setup** as root.

If you are installing the package from a tar file, unpack the archive, go (**cd**) to the k12admin-client* directory and run **make install** to install the package.

You will be asked for the name of the k12admin-server and the root password of that machine. The name of the k12admin-server can be an IP address or a host name (as long as the host name can be resolved to an IP address, of course). Once you supply the server's host name, the post-installation script will try to connect to the server using ssh to inform the server that it has a new client and copy the client's ssh key to the server. The ssh program will ask you for the root password of the server machine. The root password is not stored anywhere by the post-installation script. If you ever wish to connect your k12admin-client computer to a different k12admin-server machine, simply rerun the **k12admin-client.setup** program.

After the post-installation script completes, the server will take up to five minutes to notice the new client and register the new ssh key. Configure the new client by using a web browser to access the account administration system. From the main menu, select “Domain Tools”, then “Edit Servers”. Your new client should appear in the list of servers. Select your new client server and click on the *Edit Server* button, then configure appropriately.

Updating Files for Clients

Every five minutes, the data in the MySQL database on the server is converted to plaintext files which can be copied to the clients and used to update their local files. This process is accomplished by the /var/k12admin/scripts/server/k12.updateserver script. This script is run by **cron** every five minutes as the k12admin user. It can also be run manually, but it must be run by the k12admin user and not by root. When this script is run, it generates several files in the /var/k12admin/serverdata/ directory that are used by the clients. This script also uses a lock file in the /var/k12admin/lock/ directory to prevent multiple instances of **k12.updateserver** from running simultaneously and to prevent clients from performing their updates while the files are still being generated.

There is another key file in the /var/k12admin/serverdata/ directory, called “setup”. This file contains all the code used for updating and auditing the clients. This file is copied to the clients before the clients are updated, so do not edit this file on the clients if you are trying to add new features—edit it on the server.

On the client machines, updates are done once an hour. The script that does the updating is /usr/bin/k12.updateclient. Since this script must modify key system files, it is run as root from the system crontab file.

If run interactively, this script executes /var/k12admin/clientdata/setup with no command-line arguments. To avoid any delay in the menu coming up, the rsync is not performed. One option on the menu is to perform the rsync, if the latest data from the server is desired.

If run with command-line arguments, the script first runs rsync (using ssh as the transport agent) to sync all the files from the /var/k12admin/serverdata/ directory on the server to the /var/k12admin/clientdata/ directory on the client, then executes /var/k12admin/clientdata/setup with the same command-line arguments it received.

If the script is run without arguments, it will enter an interactive mode and present a menu of options from which to pick. It will also show progress information that is not displayed in the batch (non-interactive) mode. In batch mode, the following arguments can be supplied:

- **k12.updateclient auto**: performs all client update routines.
- **k12.updateclient passwd**: updates all password files (passwd/group/ shadow/gshadow/smbpasswd/ squid.auth/apache.auth).
- **k12.updateclient audit**: performs security audit on the client. The generated information is copied to the server, where it is inserted into the audit table in the database.
- **k12.updateclient homedirs**: creates new home directories and cleans up old home directories.

The **setup** program contains the following subroutines:

- **auto**: runs the passwd, homedirs and audit subroutines. This is the subroutine that is called from the crontab file.
- **rsync**: syncs the data from the /var/k12admin/serverdata/ directory on the server to the /var/k12admin/clientdata/ directory on the client. When setup is run interactively, this syncing is not done before the menu is brought up, so if you need the latest data from the server, choose this option first.
- **passwd**: updates the passwd, group, shadow, gshadow, smbpasswd, squid.auth and apache.auth files, depending on the configuration of the server.
- **homedirs**: creates new home directories and cleans up old ones. Old home directories can either be left alone, deleted or moved to a

temporary holding area (/var/k12admin/oldhomedirs/) where they are deleted when they are 30 days old.

- **audit**: run a security audit on the local server. This goes through various log files and determines which accounts have been used. This information is copied to the server, where it is added to the audit table in the database, so any user can review where and when their account has been used.
- **atalktable**: generates a mapping of AppleTalk nodes to AppleTalk computer names.
- **squid**: audit of the Squid proxy authentication files.
- **apache**: audit of any authenticated access to an Apache web server.
- **netatalk**: audit of access to AppleTalk services (logins from a Macintosh computer).
- **samba**: audit of access to Samba services (logins from a Windows workstation).
- **ssh**: audit of ssh (secure shell) connections.
- **squidmonitor**: not really a security audit, this searches through the proxy logs for any potentially inappropriate web browsing and also sends this information back to the server computer for review by school staff.

No changes will be made on the client computers until they have been configured through the web interface. Default settings are all “off”. For example, passwords are not updated and home directories are not created. The k12admin-client servers are configured from the Domain Tools Edit Servers menu option.

> Steve Tonnesen (tonnesen@cmsd.bc.ca) works at Coast Mountains School District in Hazelton, BC, Canada. Coast Mountains School District has 30 schools and 5,000 students.

email: tonnesen@cmsd.bc.ca

Archive Index Issue Table of Contents

Advanced search

The Linux Process Model

Moshe Bar

Issue #71, March 2000

A look at the fundamental building blocks of the Linux kernel.

This month, we begin looking at Linux internals. We will travel the innards of the Linux kernels of the 2.0.*x*, 2.2.*x* and the new 2.4.*x* series. Although many articles are written every week on how best to use Linux, very few have reviewed the internals of the kernel. Why is it necessary to know how the kernel works?

For one thing, understanding your kernel better will enable you to prevent problems before they occur. If you are using Linux as a server, most problems will start to appear under stress. This is exactly when it becomes essential to know your way around the kernel to assess the nature of the problems.

If you ever need to check back with the kernel source, you can either install the source from your distribution's CD or go to http://lxr.linux.no/source/ to navigate through all the source code on-line.

The Linux Process Model

UNIX systems have a fundamental building block: the process, including the thread and lightweight processes. Under Linux, the process model has evolved considerably with each new version.

The fundamental data structure within the kernel controlling all processes is the process structure, which grows and shrinks dynamically as processes are forked and finished or killed.

The process structure (called task_struct in the kernel source code) is about 1KB in size. You can get the exact size with this program:

```
#define __KERNEL__
#include <linux/sched.h>
```

```
main()
{
printf("sizeof(struct task_struct) - %d\n",
   sizeof(struct task_struct));
}
```

On Intel 386 machines, it is exactly 960 bytes. Please note, however, that unlike other UNIX systems, this process structure does not occupy space in the true sense of the word.

Since 2.2.*x*, the task_struct is allocated at the bottom of the kernel stack. We can overlap the task_struct on the kernel stack because the task_struct is a per-task structure exactly as the kernel stack.

The kernel stack has a *fixed* size of 8192 bytes on the Intel x86. If the kernel will recurse on the stack for 8192-960=7232 bytes, then the task_struct will be overwritten and therefore corrupted, causing the kernel to crash.

Basically, the kernel decreases the size of the *usable* kernel stack to around 7232 bytes by allocating the task structure at the bottom of the stack. It is done this way, because 7KB are more than enough for the kernel stack and the rest is used for the task_struct. These are the advantages of this order:

- The kernel doesn't have to access memory to get its kernel structure.
- Memory usage is reduced.
- An additional dynamic allocation is avoided at task creation time.
- The task_struct will always start on a PAGE_SIZE boundary, so the cache line is always aligned on most hardware in the market.

Once Linux is in kernel mode, you can get the address of the task_struct at any time with this very fast pseudo-code:

```
task_struct = (struct task_struct *) STACK_POINTER & 0xffffe000;
```

This is exactly how the above pseudo-code is implemented in C under Linux:

```
/* cut-and-pasted from
linux/include/asm-i386/current.h */
static inline struct task_struct * get_current(void)
{
        struct task_struct *current;
        __asm__(-andl %%esp,%0;
        -:-=- (current) : "0" (~8191UL));
        return current;
        }
```

For example, on a Pentium II, recalculating the task_struct beginning from the stack pointer is much faster than passing the task_struct address through the stack across function calls, as is done in some other operating systems, e.g., Solaris 7. That is, the kernel can derive the address of the task_struct by checking only the value of the stack pointer (no memory accesses at all). This is

a big performance booster and shows once again that fine engineering can be found in free software. The code to this was written by Ingo Molnar, a Hungarian kernel hacker. The kernel stack is set by the CPU automatically when entering kernel mode by loading the kernel stack pointer address from the CPU Task Segment State that is set at fork time.

The layout of the x86 kernel stack looks like this:

```
----- 0xXXXX0000 (bottom of the stack and address
                  of the task struct)
TASK_STRUCT
----- 0xXXXX03C0 (last byte usable from the kernel
                   as real kernel stack)
KERNEL_STACK
----- 0xXXXX2000 (top of the stack, first byte
                   used as kernel stack)
```

Note that today, the size of the task_struct is exactly 960 bytes. It is going to change across kernel revisions, because every variable removed or inserted to the task_struct will change the size. In turn, the upper limit of the kernel stack will change with the size of task_struct.

The memory for the process data structure is allocated dynamically during execution of the Linux kernel. More precisely, the kernel doesn't allocate the task_struct at all, only the two-pages-wide kernel stack of which task_struct will be a part.

In many UNIX systems, there is a maximum processes parameter for the kernel. In commercial operating systems like Solaris, it is a self-tuned parameter. In other words, it adjusts according to the amount of RAM found at boot time. However, in Solaris, you can still adjust this parameter in /etc/system.

What about Linux?

In Linux 2.3.*x* (and in the future, 2.4.0), it is a run-time tunable parameter as well. On 2.2.*x*, it's a compile-time tunable parameter. To change it in 2.2.*x*, you need to change the NR_TASKS preprocessor define in Linux/include/linux/tasks.h:

```
#define NR_TASKS 512 /* On x86 Max 4092 or 4090
                        with APM configured. */
```

Increase this number up to 4090 to increase the maximum limit of concurrent tasks.

In 2.3.*x*, it is a tunable parameter which defaults to *size-of-memory-in-the-system* / *kernel-stack-size* / 2. Suppose you have 512MB of RAM; then, the default upper limit of available processes will be 512*1024*1024 / 8192 / 2 =

32768. Now, 32768 processes might sound like a lot, but for an enterprise-wide Linux server with a database and many connections from a LAN or the Internet, it is a very reasonable number. I have personally seen UNIX boxes with a higher number of active processes. It might make sense to adjust this parameter in your installation. In 2.3.*x*, you can also increase the maximum number of tasks via a **sysctl** at runtime. Suppose the administrator wants to increase the number of concurrent tasks to 40,000. He will have to do only this (as root):

```
echo 40000 > /proc/sys/kernel/threads-max
```

Processes and Threads

In the last 10 years or so, there has been a general move from heavyweight processes to a threaded model. The reason is clear: the creation and maintenance of a full process with its own address space takes up a lot of time in terms of milliseconds. Threads run within the same address space as the parent process, and therefore require much less time in creation.

What's the difference between process and thread under Linux? And, more importantly, what is the difference from a scheduler point of view? In short—nothing.

The only worthwhile difference between a thread and a process is that threads share the same address space completely. All the threads run in the same address space, so a context switch is basically just a jump from one code location to another.

A simple check to avoid the TLB (translation lookaside buffer, the mechanism within the CPU that translates virtual memory addresses to actual RAM addresses) flush and the memory manager context switch is this:

```
/* cut from linux/arch/i386/kernel/process.c */
/* Re-load page tables */
{
        unsigned long new_cr3 = next->tss.cr3;
        if (new_cr3 !=3D prev->tss.cr3)
        asm volatile("movl %0,%%cr3": :"r" (new_cr3));
        }
```

The above check is in the core of the Linux kernel context switch. It simply checks that the page-directory address of the current process and the one of the to-be-scheduled process are the same. If they are, then they share the same address space (i.e., they are two threads), and nothing will be written to the %%cr3 register, which would cause the user-space page tables to be invalidated. That is, putting any value into the %%cr3 register automatically invalidates the TLB; in fact, this is actually how you force a TLB flush. Since two tasks in the same address space never switch the address space, the TLB will never be invalidated.

With the above two-line check, Linux makes a distinction between a kernel-process switch and a kernel-thread switch. This is the *only* noteworthy difference.

Since there is no difference at all between threads and processes, the Linux scheduler is very clean code. Only a few places related to signal handling make a distinction between threads and processes.

In Solaris, the process is greatly disadvantaged compared to the thread and lightweight processes (LWP). Here is a measurement I did on my Solaris server, an Ultra 2 desktop, 167MHz processor, running Solaris 2.6:

```
hirame> ftime
Completed 100 forks
Avg Fork Time: 1.137 milliseconds
hirame> ttime
Completed 100 Thread Creates
Avg Thread Time: 0.017 milliseconds
```

I executed 100 forks and measured the time elapsed. As you can see, the average fork took 1.137 milliseconds, while the average thread create took .017 milliseconds (17 microseconds). In this example, thread creates were about 67 times faster. Also, my test case for threads did not include flags in the thread create call to tell the kernel to create a new LWP with the thread and bind the thread to the LWP. This would have added additional weight to the call, bringing it closer to the fork time.

Even if LWP creation closes the gap in creation times between processes (forks) and threads, user threads still offer advantages in resource utilization and scheduling.

Of course, the Linux SMP (and even uniprocessor) scheduler is clever enough to optimize the scheduling of the threads on the same CPU. This happens because by rescheduling a thread, there won't be a TLB flush and basically no context switch at all—the virtual memory addressing won't change. A thread switch is very lightweight compared to a process switch, and the scheduler is aware of that. The only things Linux does while switching between two threads (not in strict order) are:

- Enter schedule().
- Restore all registers of the new thread (stack pointer and floating point included).
- Update the task structure with the data of the new thread.
- Jump to the old entry point of the new thread.

Nothing more is done. The TLB is not touched, and the address space and all the page tables remain the same. Here, the big advantage of Linux is that it does the above very fast.

Other UNIX systems are bloated by SMP locks, so the kernel loses time getting to the task switch point. If that weren't true, the Solaris kernel threads wouldn't be slower than the user-space kernel threads. Of course, the kernel-based threads will scale the load across multiple CPUs, but operating systems like Solaris pay a big fixed cost on systems with few CPUs for the benefit of scaling well with many CPUs. Basically, there is no technical reason why Solaris kernel threads should be lighter than Linux kernel threads. Linux is just doing the minimum possible operations in the context switch path, and it's doing them fast.

Threaded Kernel

Linux kernel threading has constantly improved. Let's look at the different versions again:

- 2.0.*x* had no kernel threading.
- 2.2.*x* has kernel threading added.
- 2.3.*x* is very SMP-threaded.

In 2.2.*x*, many places are still single-threaded, but 2.2.*x* kernels actually scale well only on two-way SMPs. In 2.2.*x*, the IRQ/timer handling (for example) is *completely* SMP-threaded, and the IRQ load is distributed across multiple CPUs.

In 2.3.*x*, most worthwhile code sections within the kernel are being rewritten for SMP threading. For example, all of the VM (virtual memory) is SMP-threaded. The most interesting paths now have a much finer granularity and scale very well.

Performance Limitations

For the sake of system stability, a kernel has to react well in stress situations. It must, for instance, reduce priorities and resources to processes that misbehave.

How does the scheduler handle a poorly written program looping tightly and forking at each turn of the loop (thereby forking off thousands of processes in a few seconds)? Obviously, the scheduler can't limit the creation of processes time-wise, e.g., a process every 0.5 seconds or similar.

After a fork, however, the “runtime priority” of the process is divided between the parent and the child. This means the parent/child will be penalized compared to the other tasks, and the other tasks will continue to run fine up to the first recalculation of the priorities. This keeps the system from stalling during a fork flooding. The code for this is the concerned code section in linux/kernel/fork.c:

```
  /*
   "share" dynamic priority between parent
   * and child, thus the total amount of dynamic
   * priorities in the system doesn't change, more
   * scheduling fairness. This is only important
   * in the first time slice, in the long run the
   * scheduling behaviour is unchanged.
   */
  current->counter >>= 1;
  p->counter = current->counter;
```

Additionally, there is a per-*user* limit of threads that can be set from **init** before spawning the first user process. It can be set with **ulimit -u** in bash. You can tell it that **user moshe** can run a maximum ten concurrent tasks (the count includes the shell and every process run by the user).

In Linux, the root user always retains some spare tasks for himself. So, if a user spawns tasks in a loop, the administrator can just log in and use the **killall** command to remove all tasks of the offending user. Due to the fact that the “runtime priority” of the task is divided between the parent and the child, the kernel reacts smoothly enough to handle this type of situation.

If you wanted to amend the kernel to allow only one fork per processor tick (usually one every 1/100th second; however, this parameter is tunable), called a jiffie, you would have to patch the kernel like this:

```
  --- 2.3.26/kernel/fork.c        Thu Oct 28 22:30:51 1999
  +++ /tmp/fork.c Tue Nov  9 01:34:36 1999
  @@ -591,6 +591,14 @@
          int retval = -ENOMEM;
          struct task_struct *p;
          DECLARE_MUTEX_LOCKED(sem);
  +       static long last_fork;
  +
  +       while (time_after(last_fork+1, jiffies))
  +       {
  +               __set_current_state(TASK_INTERRUPTIBLE);
  +               schedule_timeout(1);
  +       }
  +       last_fork = jiffies;
          if (clone_flags & CLONE_PID) {
  /* This is only allowed from the boot up thread */
```

This is the beauty of open source. If you don't like something, just change it!

Here ends the first part of our tour through the Linux kernel. In the next installment, we will take a more detailed look at how the scheduler works. I can promise you some surprising discoveries. Some of these discoveries caused me

to revalue completely the probable impact of Linux on the corporate server market. Stay tuned.

Resources

email: moshe@moelabs.com

Moshe Bar (moshe@moelabs.com) is an Israeli system administrator and OS researcher, who started learning UNIX on a PDP-11 with AT&T UNIX Release 6 back in 1981. He holds an M.Sc. in computer science. Visit Moshe's web site at http://www.moelabs.com/.

Archive Index Issue Table of Contents

Advanced search

Fun Educational Morsels, Linux-Style

Marcel Gagné

Issue #71, March 2000

In order to make the experience of logging in as simple as possible for the whole family, I recommend that you use one of the graphical login programs.

Bonjour, mes amis! Welcome back to my restaurant, “Chez Marcel”. Please, sit down. In honor of this month's special educational focus, François has prepared your tables (school desks, actually) by putting them all in a big circle. But of course, he will also be bringing your wine.

François! Du vin. Vite!

One of the questions I ask visitors to my restaurant is this: sure, you will cook with Linux at the office, but what will it take for me to convince you to try some recipes at home? As you can imagine, I get many answers, but one I hear so often is, “Sure, Linux is tasty, but what will my kids play with? I have all this wonderful educational software available on that other OS.”

Sacre bleu! Is that all? Well, pull your desks a bit closer, and together we will cook up an education and entertainment center for the kids (of all ages, non?) using Linux and open-source software. And yes, I have tried them all.

In order to keep things lively, I have asked François to hold off giving you the URL for all these fine recipes until the very end. You may take your souvenir menu as you leave the restaurant.

What's that? You think Chef Marcel has been sampling a little too much wine because he suggests turning the power of Linux over to a child? I assure you, mes amis, I have been drinking only as much as is necessary to ensure quality control. As for children, letting them have access to your system only makes sense.

Linux is a multi-user system by nature. Although a number of us instinctively log in to the root account when we start up, this is actually a very bad idea. The root login is there for you to perform administrative functions. While this is the first login we see during an install, for some users it is also the last. This is not only dangerous, but limits the reach of your Linux system. While you are logged in as root, create a user login for your child. Mon Dieu! Create one for everybody in the house! Since François is a *child at heart*, I will use him for our example.

```
adduser francois
passwd francois
```

In order to make the experience of logging in as simple as possible for the whole family, I recommend that you use one of the graphical login programs. For KDE, this is **kdm**, while GNOME uses **gdm**. That old favourite **xdm** is still available, but the less Linux-savvy may find the default X Window System environment a little less than friendly. On my Red Hat system, I set my preference for kdm by modifying my rc.sysinit script. You'll find that in the /etc/rc.d directory. The section you want looks like this:

```
# Set preferred X display manager link
preferred=kdm
if [ -f /etc/sysconfig/desktop ]; then
if [ -n "`grep GNOME /etc/sysconfig/desktop`" ]; then
        preferred=gdm
   elif [ -n "`grep KDE /etc/sysconfig/desktop`" ]; then
        preferred=kdm
   elif [ -n "`grep AnotherLevel /etc/sysconfig/desktop`" ];
then
        preferred=xdm
   fi
fi
```

By default, the **preferred=** variable is not set. I changed mine so that it reads **preferred=kdm**. Another way to do the same thing is to edit the /etc/sysconfig/desktop file and simply put in the name of your favourite desktop environment. A couple of lines further on (in the next paragraph of the rc.sysinit), the system sets the link for **prefdm** which is then started in your /etc/inittab file.

```
ln -snf ../..`which kdm` /etc/X11/prefdm
```

During your install, you may have been given the opportunity to let Linux boot into a graphical desktop. If you chose not to do this, you can still change your mind. This is the tail-end section of my /etc/inittab file:

```
# Run xdm in runlevel
# xdm is now a separate service
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Simply add or edit the last line to the inittab, then run the following command to activate it for next time and each successive boot. You will want to do this when your X environment is not running.

```
init q
```

The reason for that overview is to get you used to the idea of logging in and creating different profiles for all the users in your family, each with their own **protected** environment. Now, from your own login, you can start adding games. I created a folder for François, added and built various games, then copied these to his desktop. In KDE, when you create a folder on your desktop (or a link of some sort) you can find those items in a folder called “Desktop”. For François to have the same folder when he logs in, I simply copy my folder to his desktop. Since I broke my own rules and am running as root, I will also change the permissions on his new folder so he is allowed to open it himself.

```
cd Desktop
find "Educational Games" -print |
cpio -pduvm
/home/francois/Desktop
cd /home/francois/Desktop
chown -R francois "Educational Games"
# chgrp -R francois "Educational Games"
```

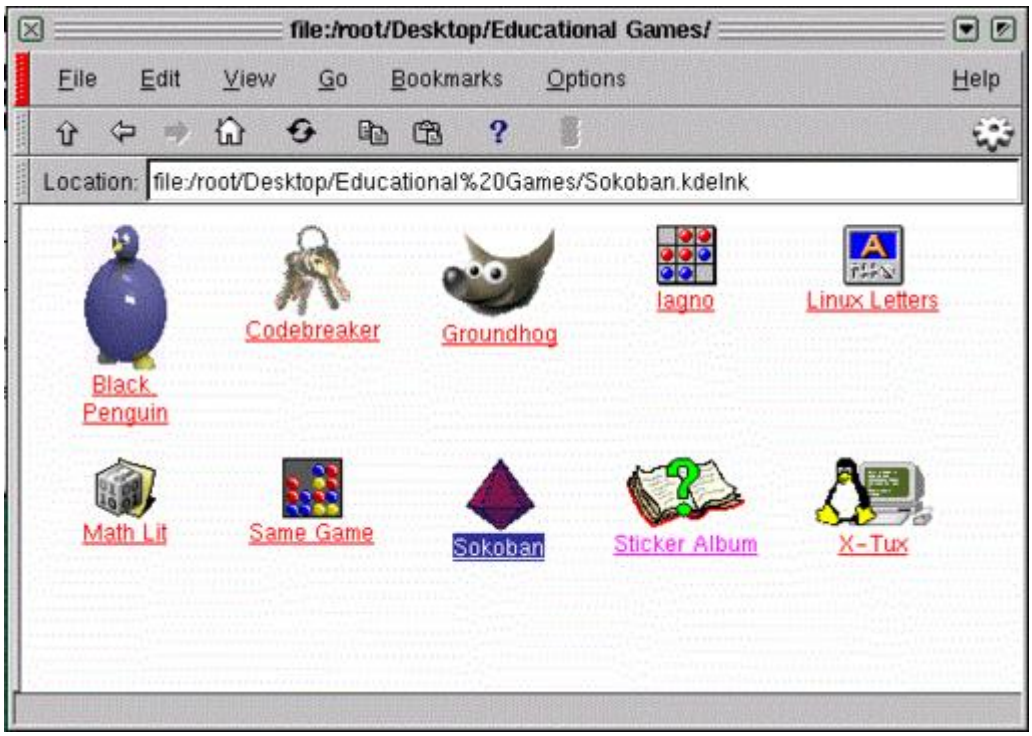


Figure Games Menu

When he logged in, his desktop had a nice little folder with all the appropriate links and icons. Incidentally, this is a great way to maintain (and rebuild if need be) another user's desktop. If you want my little folder (as an example), you can download it from the link at the end of the article. I've also included a web page with links to all the items we will sample, if you don't want to go hunting for everything on your own. (See Figure)

When starting your educational games collection for the kids, look no further than your desktop environment. Both KDE and GNOME come with several games (as does X), some of which fit the bill very nicely. (Non. That “bill” comes

later.) Some of the KDE games will have a counterpart in GNOME and vice versa. Your quest is to find the one you like best.

For instance,. . . **Reversi** is a classic strategy game that requires some planning. In this case, the GNOME version (known as **Iagno**) is more appealing to me for its animation and graphics. The only drawback (and the KDE version's plus) is the lack of board lines. For the younger set, it might require a few tries to get the hang of token placement.

In terms of strategy and logic, I am still a big fan of **Sokoban**, the Japanese "warehouse keeper" game. You will find this one among your KDE games. The object is to move large gems into their proper storage in an increasingly crowded warehouse. Sokoban takes patience, but does wonders for the analytical process.

Eventually, you will want to try something not included with your Linux distribution. While the excitement of Linux hasn't yet generated the vast amount of educational titles stocking the shelves for that other operating system, the selection is growing and will continue to do so. If you want to track what is happening in the world of children's games specifically, try the **Linux for Kids** web site. For the bigger kids, keep an eye on the **Linux Game Tome** and **Linuxberg** to name just a couple. Move your desks a little closer together, and sample what I found in my initial search.

To not only challenge, but twist and turn a young mind, I recommend **Groundhog**. This is a wonderful little puzzle where you manipulate tubes (or groundhog tunnels, I presume) to move little colored balls into their corresponding little colored pockets. So far, I have managed the two-ball by two-ball level, but I think I will graduate soon.

Codebreaker is a GPL'ed version of Mastermind, an old favorite of mine. This "guess the color code" game is still a great exercise in logic. For those who consider themselves experts at the game, I suggest trying it with eight different colors.

There are times when I really don't need to learn much. I just want to save the world from an evil empire, say a large multinational software empire bent on world domination. Enter **XTux**, a great little game that teaches some interesting lessons. For starters, CD-ROMs make great weapons with which to destroy mindless droids, and coffee is all you really need to keep you going. Several characters are provided to help you get the job done: Tux, Gown, BSD Daemon, Emacs and Vi. What can I tell you? It's silly, but fun.

Probably the most fun I had while putting this together was a little something called the **Sticker Book**. The concept is so simple, it's brilliant. Like the sticker books you had as a kid, this one comes with a variety of images and a handful of different backgrounds. You pick up the stickers by clicking on the various images, then simply paste them onto your background with another click. You can also increase the size of images, shrink them and rotate them. If you don't like the way it is turning out, you can erase everything and start over. In the process of writing this article, I had a couple of adults watching over my shoulder, and they were captivated. The only downside (and I hope the developers read this) is that the tabs which let you flip between series of stickers are a bit small for the large arrow pointer they supply. If it wasn't for the adult response, I would've said this was a game for the very young. Try this one. It's deceptively addictive.

Another selection for the very young is **Linux Letters and Numbers**. Essentially, this is a board with large tiles representing the numbers 0 through 9 and the letters of the alphabet. Clicking on a tile brings up a picture of something starting with that letter, or a collection of little pictures representing the number (4 apples, for instance). Letters and Numbers comes with a collection of tiles, but you can create your own when the kids start anticipating the image that follows the letter. “P” is for Penguin!

For a slightly older audience, try **lolit Math Literature**. This is a math training program that helps teach addition, subtraction, multiplication and division. The interface is fairly slick-looking and the questions go from very simple to fairly complex. Just what is 87+54, anyway?

Well, mes amis, it is once again closing time. I sincerely hope I have whetted your appetite for turning over your Linux system to the kids (at least every once in a while). Do come back soon. You are always welcome here at *Chez Marcel*.

Bon Appétit!

Resources

Marcel Gagné lives in Mississauga, Ontario. In real life, he is president of Salmar Consulting Inc, a systems integration and network consulting firm. He is also a pilot, writes science fiction and fantasy, and edits TransVersions, a science

fiction, fantasy and horror magazine. He loves Linux and all flavors of UNIX and will even admit it in public. He can be reached via e-mail at mggagne@salmar.com. You can discover lots of other things from his web site at http://www.salmar.com/.

Archive Index Issue Table of Contents

Advanced search

LINUX JOURNAL

Consumer Rankings

Reuven M. Lerner

Issue #71, March 2000

How to use CGI programs to allow list subscribers to enter and rank their favorite products or services.

On-line booksellers have changed the way I decide which books to buy. Not only have they made it possible for me to get most books within two to three days, but they have also given me a means of comparison shopping previously unavailable to me. If I am thinking of buying a book, I immediately look to see what others have written about it. In most cases, these evaluations clinch the deal, convincing me to buy a certain book instead of its competition.

I have been thinking more about such systems since my recent move to Modi'in, a new Israeli city halfway between Jerusalem and Tel Aviv. Just before I moved to Modi'in, I was asked to take over a small mailing list for residents of Modi'in. The list gives people a chance to share local announcements and ideas related to life in Modi'in.

Soon after becoming the list administrator, I realized that subscribers often asked for recommendations, from doctors to lawn services to after-school activities. Normally, an e-mail list administrator who sees such constant repetitions will prepare a FAQ, a list of common questions and their answers. But recommendations are extremely subjective, and one person's favorite barber may be someone else's nightmare.

This month, we will look at a set of CGI programs I wrote to allow list subscribers to enter and rank their favorite products and services in town. Because my web space provider does not offer **mod_perl**, I had to use the CGI standard for writing my programs.

This “ranking” system, as I call it, consists of three CGI programs written in Perl, which use a relational database for back-end storage. (These listings are not printed here due to space considerations, but can be downloaded from the *LJ*

FTP site. See Resources.) I have used MySQL, but there is no reason another relational database, such as PostgreSQL or Oracle, could not be substituted for it. Some of the SQL syntax might have to be changed in order to fit another database server, but the majority should remain the same.

While the ranking system presented this month is not as sophisticated as the one used by Amazon.com nor as flexible as that used by Epinions.com, it does serve a simple purpose. Moreover, it demonstrates how to produce a simple ranking system, which could easily be extended to produce a rough version of Epinions.com.

Creating the Tables

As always, the first step in creating a database/web application is to consider how we want to store the information, and then to create the tables in our relational database. In this particular case, we will keep things simple, dividing ranked items into categories, but without any hierarchy representing the categories. We will thus be able to keep all restaurants in the same category, but without any distinction between Italian restaurants and French restaurants. Alternatively, we can create two separate categories for Italian and French restaurants, but then the system will see them as unrelated as barbers and vacuum-cleaner salesmen.

We will also associate a name and e-mail address with each ranking. It might be more elegant to place user names in a separate table and refer to them with a numeric key. However, we are less interested in tracking users than in making it possible to find useful consumer information.

Given all this, I decided to implement the ranking system with three tables: RankCategories, RankItems and Rankings. RankCategories, as its name implies, contains the categories into which the items are sorted, and can be defined as follows:

```
CREATE TABLE RankCategories (
category_id MEDIUMINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
category_name VARCHAR(25) NOT NULL,
category_description TEXT NULL,
UNIQUE(category_name)
);
```

Notice how each category will have a unique **category_id**, allowing us to refer to it by number alone. By using the **AUTO_INCREMENT** feature, MySQL can automatically set this number for us, ensuring that it is unique. For a small web site, a **MEDIUMINT** is large enough, ranging from 0 to 16,777,215. A large site might eventually have more than 17 million rankings, in which case a larger size, such as **INT** or **BIGINT**, might be a good idea.

We also want to ensure that no two categories can have the same name, and thus add a **unique** constraint to the **category_name** column as well. Each category can then have some descriptive text associated with it, which is placed in **category_description**.

The items to be ranked are placed in a similar table, RankItems:

```
CREATE TABLE RankItems (
item_id MEDIUMINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
category_id MEDIUMINT UNSIGNED NOT NULL,
item_name VARCHAR(25) NOT NULL,
item_description TEXT NULL,
UNIQUE(item_name)
);
```

Once again, each item has a unique value in **item_id**, and the item name is guaranteed to be unique, thanks to the **UNIQUE** constraint. However, RankItems adds a **category_id** column, identifying the category in which this item sits.

RankItems defines the items available to be ranked, but does not store the grades. That role is assigned to the Rankings table, defined as follows:

```
CREATE TABLE Rankings (
ranking_id MEDIUMINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
item_id MEDIUMINT UNSIGNED NOT NULL,
ranker_name VARCHAR(30) NOT NULL,
email VARCHAR(40) NOT NULL,
entry_datetime TIMESTAMP(8) NOT NULL,
comments TEXT NULL,
rank TINYINT UNSIGNED NOT NULL,
UNIQUE(item_id, email)
);
```

Like the other tables, Rankings gives each ranking a unique primary key, **ranking_id**. None of our applications use **ranking_id**, and it is possible that its inclusion in the table definition wastes some space on disk and in memory. However, such a primary key will make it easier to refer to items in Rankings if and when we build additional applications for the ranking system.

Rankings then contains an **item_id** column, which refers back to the primary key of the same name in RankItems. Because each item is in a single category and RankItems contains a **category_id** column, there is no need to name the category in Rankings as well.

We ask each ranking user to provide his or her full name and e-mail address. This information is displayed next to a ranking, in order to give it a slightly more human touch. In addition, some people's opinions will carry more weight than others, especially in a small community of users, so it is worthwhile to identify opinions by name.

The ranking consists of a numeric rank between 0 and 10, stored in a **TINYINT**. It is accompanied by optional (**NULL**) comments, stored in a **TEXT** column, where the user can elaborate on his or her ideas.

To ensure each user can rank each product only once, we have MySQL require the combination of item ID and e-mail address to be unique in the Rankings table. By combining the two in this way, the database itself will reject any attempt to enter two rows in which the combination of e-mail address and **item_id** are identical.

Adding Categories

Now that we have seen how the data will be stored, we will write several programs that allow users to enter information into the tables. The interface presented here might seem a bit primitive, but this doesn't really matter. Since the information is stored in a database, we can always write new programs to improve or change the interface.

Since each RankItem must be placed in a category, we must first write a program that adds new categories to the system. Listing 1, rank-category.pl (see Resources), is a simple example of such a program.

When invoked with the **GET** method, such as when a user enters a URL into a browser, rank-category.pl produces an HTML form into which details about a site can be entered. The form is submitted to rank-category.pl which handles the data submitted with POST, adding a new row into the RankCategories table. After the new row is added to the database, rank-category.pl displays the HTML form once again. This makes it relatively easy to add multiple categories.

rank-category.pl, as well as the other programs presented this month, use **CGI.pm**, the standard Perl module (written by Lincoln Stein) for working with CGI programs. We also import a number of subroutines into the current name space (grouped together with the **:standard** tag), making it easier to produce HTML output. The **p** function, for example, surrounds text with <P> and </P> tags. Thus the following:

```
print p("Hello");
```

is the same as

```
print "<P>Hello</P>";
```

but is more readable. In addition, CGI.pm's HTML subroutines can work on multiple strings and can be nested:

```
print p("This will be in", b("bold"), "type");
```

CGI.pm automatically inserts whitespace between arguments to these subroutines, so there is no need to put spaces at the end or beginning of quoted strings.

To set HTML tag attributes, pass a hash reference to the subroutine as the first argument. For example, the following code comes from rank-category.pl (Listing 1):

```
print p("Now go ",
  a({-href=>'/cgi-bin/rank-something.pl'},
      "rank something"), "!");
```

The above code puts the text “rank-something” inside of a hyperlink pointing to /cgi-bin/rank-something.pl, producing the same HTML as the following, without having to worry about quotes or whitespace:

```
print qq{<P>Now go
<a href="/cgi-bin/rank-something.pl">
rank something</a>!</P>
```

By default, CGI.pm does not insert whitespace between HTML tags generated using these routines. This might be more efficient for the computers, but makes it difficult to examine and debug the HTML output. We thus use **CGI::Pretty**, a subclass of CGI.pm which inserts appropriate whitespace between the tags. (If your system does not support CGI::Pretty, use CGI.pm instead. The output will not be formatted as nicely, but will still work.)

Regardless of how it is invoked, rank-category.pl always produces an HTML form consisting of two text fields, **new_category_name** and **new_category_description**. These are submitted back to rank-category.pl, which inserts the information into the database.

The program's connection to the database is managed by DBI, the Perl database interface available from CPAN (the comprehensive Perl archive network, at http://www.cpan.org/). DBI provides a generic API to a relational database, in conjunction with a specific driver (DBD) for each type of database server. DBDs are available for most popular brands of database server and make it possible to port Perl programs across platforms with a minimum of difficulty.

Assuming that **new_category_name** and **new_category_description** contain valid data, we insert a new row into the RankCategories table by building up an SQL query. Here is the Perl code that performs this task:

```
my $sql  = "INSERT INTO RankCategories ";
 $sql .= "(category_name, category_description) ";
 $sql .= "VALUES (?,?) ";
```

```
my $sth = $dbh->prepare($sql)
   || die "Cannot prepare: $DBI::errstr";
my $success =
   $sth">execute($new_category_name,
       $new_category_description);
```

Notice how **$sql** contains question marks (**?**) rather than actual data. These values are set in **$sth->execute**, with the first argument (**$new_category_name**) being assigned to the first place holder and the second argument (**$new_category_description**) assigned to the second.

rank-category.pl does not benefit in an obvious way from the use of place holders, which speed up multiple invocations of the same SQL query, with minor variations. However, using place holders also makes it possible to avoid problems that can arise when interpolated variable values contain **'** and **"** characters. DBI handles and quotes them automatically, making the program more readable and eliminating the possibility of these sorts of errors.

Creating the Ranking Form

Once one or more categories have been added to RankCategories, users can begin to rank individual items. This is accomplished with Listing 2, rank-something.pl (see Resources), which produces an HTML form so that users can rank an item. Users have the option of ranking something already in the database, or adding a new item inside an existing category.

The form must be created by a program, rather than written as a static document, because it displays pop-up menus of catalogs and items from the database tables. When it is first invoked, rank-something.pl retrieves the items and categories from RankItems and RankCategories, respectively, putting them into hashes for easy retrieval.

DBI makes it relatively easy to retrieve a table into a hash with its **fetchrow_hashref** method. However, the documentation explicitly states that fetchrow_hashref is not very efficient, so we use **fetchrow_arrayref**, retrieving each row as its own array reference and storing the information in a hash:

```
$sql  = "SELECT category_id, category_name ";
$sql .= "FROM RankCategories ";
$sth = $dbh->prepare($sql)
   || die "Cannot prepare: $DBI::errstr";
$result = $sth->execute
   || die "Cannot execute: $DBI::errstr";
my %categories = ();
while (my $row = $sth->fetchrow_arrayref)
   {
   my ($id, $name) = @$row;
   $categories{$id} = $name;
   }
```

The “existing item” and “new item” sections of the form both use the **popup_menu** function provided by CGI.pm for the creation of **<select>** lists. A

pop-up menu has a name and a set of potential values, each of which can be optionally associated with descriptive text. For example:

```
<select name="number">
   <option value="1">one
   <option value="2" selected> two
   <option value="3">three
</select>
```

The above HTML creates a three-element pop-up menu named “number”. This menu can pass one of three options (**1**, **2** or **3**) to a CGI program. However, the user never sees the values; instead, the menu is labeled with English words corresponding to the values. While the user can select and submit any of the menu's values, the **two** element is selected by default.

Given that the existing items are placed in the **%items** hash (just as the above code places existing categories in the **%categories** hash), we can create a pop-up menu with the following code:

```
popup_menu(-name => "existing_item_id",
   -values => [0, (sort {
   $items{$a} cmp $items{$b}} keys %items)],
      -labels => {0 => "Choose one",
      map {($_, $items{$_})} keys %items},
   -default => $existing_item_id,
   -override => 1),
```

popup_menu returns a text string, appropriate for passing to **print** or saving to disk (if the program is creating a file of HTML-formatted text). The **name** parameter sets the **name** attribute, and the **default** parameter indicates which attribute will be selected by default.

The **values** parameter takes an array reference as an argument. In this case, the values should be the keys of **%items** (i.e., the primary keys from RankItems), but sorted in the order of each key's **item_name** value. The solution is to sort **keys %items** by value, rather than by key. Adding a 0-value element to the front of the **values** parameter means the first option will always have a value of 0. MySQL does not use 0 in **AUTO_INCREMENT** columns, and Perl sees 0 as false—so setting the default value to 0 will never collide with an actual value of **item_id**, and can easily be identified when passed to our program via **POST**.

By default, popup_menu will set the descriptive text for each **<option>** to the value itself. However, the **labels** parameter makes it possible to assign custom labels to some or all values. The **labels** parameter takes a hash reference as input. Since a hash in list context turns into a comma-separated list of keys and values, we can stick **%items** into the hash reference, preceded by a mapping from 0 to the text “**Choose one**”.

Unfortunately, HTML does not support hierarchical pop-up menus. It would be easiest and best for the user if we could use popup_menu to provide a menu of

categories that would lead to a list of associated items. Given the choice between requiring a second program invocation (as we did with view-ranking.pl, described below) and producing a flat list of items, I chose the latter. Another approach is to use the method pioneered by Yahoo! and used by Epinions, in which each category hyperlink is followed by links to the most popular individual items. Implementing such an approach would require some changes to rank-something.pl (and presumably view-ranking.pl), but not to the underlying database.

Inserting a New Ranking

When rank-something.pl is invoked with the POST method, it looks for a new item that it will need to insert into RankItems. The actual insertion is rather straightforward, using DBI's **prepare** and **execute** methods, as we have seen before:

```
$sql  = "INSERT INTO RankItems ";
$sql .= "(category_id, item_name, item_description) ";
$sql .= "VALUES (?,?,?) ";
$sth = $dbh->prepare($sql)
   || die "Cannot prepare: $DBI::errstr";
$success =
   $sth->execute($item_category, $item_name,
   $item_description);
```

If the execution is unsuccessful, **$success** will be set to a false value. We can trap errors this way, producing an error message like the following:

```
unless ($success)
 {
 print h1("Error inserting new item");
 print p(
 "There was an error inserting the item:".
 $DBI::errstr");
 print p(
 "Perhaps this item already exists in the".
 database?");
 exit;
 }
```

We can now be sure the item to be ranked is in RankItems. If the item was already in RankItems, then we knew its primary key from the **<select>** list in the HTML form. However, how can we retrieve the primary key of the new item just inserted? The DBI driver for MySQL provides an attribute, called **mysql_insertid**, which returns the primary key of the most recently inserted row. We can retrieve this value and store it in **$item_id**, which otherwise would get its value from the pop-up menu:

```
$item_id = $dbh->{"mysql_insertid"};
```

As we saw earlier, each ranking consists of an integer between 0 and 10 (from worst to best), as well as user comments about the item. In order to avoid potential formatting problems, I decided to remove all HTML tags from the comments. Using Perl's non-greedy regular expressions, this is an easy task:

```
$comments =~ s|<.*?>||g;
```

The above substitutes all occurrences of <, followed by zero or more characters, followed by >, with the empty string. The |g modifier at the end of the s||| operator performs this operation globally.

Next, we handle paragraph separators. We want people to be able to enter more than one paragraph, but cannot let them use <P> and </P> tags to do so. The solution is to treat every occurrence of more than one \r (carriage return) or \n (newline) character as a paragraph separator. UNIX machines rarely produce \r characters in text input, but DOS/Windows systems end lines with a combination of \r\n, and Macintoshes use a single \r. The following turns any two (or more) of these into a <br> tag, followed by two newlines:

```
$comments =~ s|[\r\n]{2,}|<br>\n\n|g;
```

Finally, we insert the new ranking into the Rankings table:

```
$sql  = "INSERT INTO Rankings ";
$sql .= "(item_id, ranker_name, email, comments, rank) ";
$sql .= "VALUES (?,?,?,?,?) ";
$sth = $dbh->prepare($sql)
   || die "Cannot prepare: $DBI::errstr";
$success =
   $sth->execute($item_id, $ranker_name, $email,
   $comments, $rank);
```

Once again, we use question marks (**?**) as place holders, potentially speeding up multiple queries and removing the need for us to explicitly quote the individual items.

If the INSERT is successful, the user is given a short message:

```
print p("Your ranking was successfully entered.");
```

Following this message, the user is given the chance to rank another item in the database or view the current rankings database by clicking on a hyperlink.

Retrieving Data

Finally, once data has been entered into Rankings, we can write Listing 3, view-ranking.pl (see Resources), a program that lets us look through the rankings and read them. There are many ways to present the information, and I took the easy way in this version of the program, forcing users to go through two menus (one of categories and a second of items in that category) before seeing the list of rankings for a particular item. Once again, we use the GET/POST trick to write a program that both creates a form and accepts its data.

If view-ranking.pl is invoked with GET, it creates a simple HTML form with the categories in a pop-up menu:

```
print $query->start_html(-title =>
   "Choose a category");
print h1("Choose a category");
print startform(-method => "POST",
   -action => $query->url);
print p("Select a category to view:",
   popup_menu(-name => 'category_id',
   -values =>
   [sort {$categories{$a} cmp $categories{$b}}
          keys %categories],
   -override => 1,
   -labels => \%categories));
print submit(-value =>
   'View items in this category');
print endform;
```

If view-ranking.pl is invoked with POST, it checks to see whether **category_id** was set. If so, then it assumes the invoking form was the above, and displays a list of items within that category:

```
$sql  = "SELECT C.category_name, I.item_name, ";
$sql .= "       I.item_description, AVG(R.rank)";
$sql .= "FROM RankItems I, RankCategories C, ";
$sql .= "     Rankings R ";
$sql .= "WHERE I.category_id = C.category_id ";
$sql .= "AND   I.item_id = $item_id ";
$sql .= "AND   I.item_id = R.item_id ";
$sql .= "GROUP BY I.item_id = R.item_id ";
$sth = $dbh->prepare($sql)
   || die "Cannot prepare: $DBI::errstr";
$result = $sth->execute
   || die "Cannot execute: $DBI::errstr";
```

This seemingly complex SQL query retrieves information from all three tables, getting the category name from RankCategories and information on this item from RankItems. The AVG function returns the average value from all returned rows, making it possible to get a sense of where the item should truly be ranked. Of course, there is no way to stop a malicious user (or the owner of a ranked business) from trying to skew the scales in a particular direction, so it is just as important for users to read the comments and the individual rankings as the average score.

Once we have retrieved general information on the item, we perform a second **SELECT**, requesting all rows for this item in chronological order:

```
$sql  = "SELECT ranker_name, email, comments, rank ";
$sql .= "FROM Rankings ";
$sql .= "WHERE item_id = $item_id ";
$sql .= "ORDER BY entry_datetime ";
```

The results of this query are then printed for the user:

```
while (my $row_ref = $sth->fetchrow_arrayref)
  {
  my ($name, $email, $comments, $rank) =
      @$row_ref;
print p(dt(a({href => "mailto:$email"},
```

```
    $name), "*" x $rank, " ($rank)"),
  dd($comments));
  }
```

HTML's <dt> and <dd> tags are perfect for formatting these sorts of comments, handling the indentation automatically. Perl's **x** operator, which multiplies a text string, makes it simple to produce the correct number of stars associated with a particular review.

Conclusion

The ranking software I presented this month is only in its earliest stages and will undoubtedly have improved by the time this issue of *Linux Journal* reaches your hands.

Along with improvements in the user interface and the possible addition of a hierarchical category system, there are several related applications I would like to add. One would return the highest-ranked items within a category, allowing a user to find the best choice without having to wade through dozens or hundreds of reviews. Hard-core users of this system might want to read only those reviews which have appeared since the last time they visited the site. Some additional personalization, including the use of cookies to remember the user's name and e-mail address, would reduce the amount of typing someone would have to do. An editing facility for system administrators will undoubtedly prove useful as the system grows. Finally, it is always nice to provide a search function, just in case an Italian restaurant is accidentally classified as a café.

The programs we examine this month demonstrate that it is not particularly difficult to produce a simple user ranking system. This is especially true if we take advantage of the power a relational database affords us in storing and retrieving data. Best of all, subscribers to my e-mail list can now spend their time trading information, leaving the recommendations to a set of CGI programs.

Resources

Reuven M. Lerner , an Internet and Web consultant, moved to Modi'in, Israel following his November marriage to Shira Friedman-Lerner. His book Core Perl

will be published by Prentice-Hall in the spring. Reuven can be reached at reuven@lerner.co.il. The ATF home page, including archives, source code and discussion forums, is at http://www.lerner.co.il/atf/.

Archive Index Issue Table of Contents

Advanced search

Linux Goes to Algebra Class

Gail Fultz

Issue #71, March 2000

Learning algebra can be fun using a computer and software from Learning Logic Labs.

I remember a summer day that was as wonderful as a carefree day can be, but it slipped away to produce probably the worst day of my life. For on that day, I received my eighth-grade algebra schedule. How was it possible that I had been placed in Mr. Johnston's class? He, who was capable of striking fear in the hearts of even the boldest of my classmates. Didn't he slam students against the lockers? Didn't he yell at them in class? How was I, a quiet and shy thirteen-year-old, to survive the year? I felt there just had to be some mistake. It wasn't fair that this was happening to me, considering math was usually my favorite class. But no, there wasn't any mistake or any getting out of it. Mr. Johnston's algebra class became a reality for the next nine months of my young life. I discovered he was even more fearsome than I had heard. I decided my goal for the year wasn't to get an “A”, but to survive until June! Even though Mr. Johnston is tough, I soon learned first-hand that he is an excellent teacher who cares deeply about his students. I made this discovery shortly after the third week of school. I finally began to breathe easier, relax a bit in class and enjoy the work.

By the beginning of the second six weeks, I was getting bored. Could it be because I understood how to solve the equation the first time he showed us how? Why couldn't he just let me go ahead and solve the equations once I understood? I didn't need his help after the first time. I was getting frustrated! Does this sound familiar? No doubt some of you have been in the same situation. You've had to endure a class in which the instructor was teaching at a rate geared for the average student rather than your faster rate. Or, perhaps the opposite is more accurate. You feel totally lost, everyone else is catching on quickly, and the teacher is leaving you behind.

Wouldn't it be nice if all learners could move at their own rate, rather than be forced to move at the teacher's pace? Students would have the best of both worlds if material were fully explained on a computer screen, so the classroom teacher would be free to work individually with the student if a problem occurred. Think of the advantages to having all material randomized, so that students on the same screen would be presented a different set of problems. This would mean students couldn't copy answers from each other on homework or tests. Imagine having homework assignments automatically printed with the student's name, and later checked by the computer for immediate feedback.

How about automatically generated tests, presented to cover specific material the student has mastered since the last test? Consider the increased teacher productivity that could result from all students working independently on material specific to their needs, and grades being managed by a gradebook feature. How about the luxury of being able to schedule students assigned to Pre-algebra, Algebra I and Algebra II into one computer lab during the same period of time? What about the advantage for students mastering a section of material before they move on, so as to avoid of misunderstanding showing up in a later course? Consider the convenience of having the program automatically download enhancements and additional courseware as they become available. Imagine all this, along with a mechanism for teachers and students to communicate directly with program developers.

Do these features sound like a math teacher's dream? They're a reality with the development of Learning Logic (L2). A self-paced, computerized algebra program, Learning Logic is currently being used in algebra classrooms in twelve U.S. states and runs on a Red Hat 5.2 Linux platform.

The average Learning Logic Lab consists of a single Linux server. Students work on X terminals running in 1024x768 mode connected to the server via a 10- or 100-base twisted-pair network. A PostScript-capable printer automatically prints student homework and tests. A classroom modem allows the NSCF to connect remotely to the server. A PPP connection to the server can also be set up to assist the NSCF Support Team in diagnosing any possible problems. The programs Maple and Reduce are used as an artificial intelligence engine for evaluating a wide range of possible student answers.

Why Linux? For starters, it's a free operating system! The source is freely available, so if there is a need to fix something, it can be done without external assistance. It's also much easier to debug. Linux has message and error logs to indicate specific problems, and if that doesn't work, one can always read the core files. It comes with networking built in, so there is no need for third-party

software. Compilers and development libraries already exist, so there is no need to buy expensive libraries.

Drivers evolve quickly since many people worldwide are working in Linux to develop software. Many commercially available UNIX systems don't have the driver support, due mostly to a limited staff. Often, one has to buy extra driver packs.

With Linux, most drivers are free. In addition, Linux does not install a new package or replace system libraries without letting users know. Window programs often store things in the main system directories when a new program is installed. This can cause a problem if the libraries are different. Linux updates are quick, and mail and network services are usually included in any commercially available release. They usually include the source code.

Linux is particularly well-suited for a school environment for several reasons. Firewalling and other forms of network security are built into the OS, and Linux supports them at the kernel level. As with most types of UNIX, viruses are virtually nonexistent in Linux. There are also a number of drivers available, and Linux does not throw away the *obsolete* drivers just because someone *thinks* no one uses them any longer. Schools are generally on limited budgets, so they often have to make do with older hardware. Most importantly, with Linux a standard PC can be used to run an entire classroom of 25-30 workstations.

Why Red Hat? When project development began, Slackware was used before migrating to Red Hat Linux 4.1. Even though it was not free, at the time Red Hat was the easiest to install and had the most professional installation interface. Essentially, it had everything needed right out of the box. RPMs make it easier to install updates and fixes, and it provides a common database to search through to determine which version of a specific package is installed. Learning Logic was created at the National Science Center Foundation (NSCF), a not-for-profit foundation located in Augusta, GA. The NSCF has no salespeople and Learning Logic is spread mainly by word of mouth. Teachers are attracted to Learning Logic's flexibility, its ability to be tailored to meet the needs of the individual student and its variety of teacher management and reporting tools such as the gradebook feature.

Some schools report that students who use Learning Logic Algebra I are faring better than their peers when both groups reach Algebra II. Data indicate that L2 students in these schools maintain a lower Algebra II failure rate. Teachers also report that L2 students tend to make more As and Bs in Algebra II, compared to students who are not exposed to Learning Logic Algebra I. In the fall of 2000, Learning Logic will celebrate its tenth anniversary. At its inception in 1989, a Motorola 88K multiprocessor system was used in Learning Logic classrooms.

SunOS, AIX, SCO and OSF/1 have each been used as the operating system at some point. Now, faster PCs and Linux result in improved capabilities and speed as well as a wider range of hardware choices.

At the time Learning Logic was first introduced into schools during its beta test period, UNIX had very few followers in the secondary school arena. However, since Linux has gone to algebra class in the form of Learning Logic, it has become more of a household word in Learning Logic schools.

Gail Fultz was a secondary school mathematics and Spanish teacher for over fifteen years and has worked in the UNIX environment for more than ten years. She is currently Director of Educational Programs for the National Science Center Foundation in Augusta, GA. She welcomes your comments sent to gail@nscf.org.

Archive Index Issue Table of Contents

Advanced search

Moving to SMP

Michael S. Keller

Issue #71, March 2000

Wondering about multiprocessing? Think it might be fun? For one man's experience with setting up SMP, read on.

For some, SMP, symmetric multi-processing, represents everyday work; for others, a hope for performance gains. Windows 9*x* users need not bother, since that platform supports only one CPU per host. Windows NT does not scale as well as other operating systems on multiple processors. The 2.0 Linux kernel series provided some SMP support, but the 2.2 series has much better support. Linux scales well, up to 16 CPUs. Beyond that, Linus Torvalds has not yet decided to commit, because the tradeoffs required to make Linux scale well to more would mean compromising performance for small systems. For large-scale SMP, Sun's Solaris does well at 64 CPUs. SGI and IBM also have large-scale SMP offerings. This article provides an introduction to running SMP under Linux on the x86.

Why SMP?

For some time, I had wanted to experiment with SMP Linux. When the 2.2 kernel appeared and another hardware transaction in my home network left me with a 486 for my desktop machine (my wife got the K6-200 to go with her new keyboard, where the integrated pointer required the only PS/2 mouse port in the house), I decided to replace the 486 with something a bit different.

While looking longingly at multi-CPU Alpha systems and Intel P6 and Pentium II motherboards, I decided to build an SMP machine at a lower cost, but with new parts. Alpha and Ultra-SPARC systems did not have the low price I preferred to pay, and since I had a Baby AT case, Pentium II motherboards would not work with the hardware I could recycle. That left Baby AT-size P5 and P6 motherboards. I did not find any new P6 motherboards at appealing prices, but I did find a few P5 boards that would work.

After conducting more research, including reviewing past discussions on SMP at Slashdot, I chose the Tyan Tomcat IV 1564D, which sports the Intel 430HX chip set. This board can use one or two (preferably matched) Pentium processors from the P5-75 to the P5-233 MMX. It can also hold up to 512MB RAM, spread across eight 72-pin SIMM slots, using parity or non-parity memory devices. It has typical on-board I/O, including two IDE, two serial, one parallel, one PS/2 mouse port and USB. It can also use non-Intel CPUs, but will support only one, since Intel had de facto control of SMP in the x86 world until the AMD Athlon reached market, and the non-Intel offerings that support any SMP do not use the Intel signalling.

I purchased two P5-233 MMX CPUs and the motherboard from Motherboard Express. I added 128MB of fast-page parity memory from Crucial Technology.

SMP Requirements

For proper operation, SMP should combine with a thread-safe libc, such as glibc2. I run Debian GNU/Linux version 2.1, which has all libraries and utilities up to date for SMP. Debian's package set also includes libc5 libraries for software compiled to require libc5. After three years of near-continuous use, I find Debian the most pleasing Linux distribution. I have had no trouble performing upgrades and keeping current with updates. The package manager requires no manual downloads, and retrieves only the packages necessary for staying current. (My previous and parallel experience with Red Hat, through version 5.2, found no such facility built into or near RPM. If it exists, I missed it.)

Some drivers also require updates in order to perform correctly under an SMP kernel, since additional locking must occur to reduce contention for system resources. 4Front Technologies' OSS sound driver comes in uniprocessor and SMP varieties. PCMCIA Card Services may require recompilation. Most other drivers reside in the kernel source tree, so they should work with SMP after compiling a new kernel.

Making It Work

After replacing the 486 motherboard with the new Tyan unit, Linux booted straightaway. I already had the 2.2 kernel running, and so reconfigured it for SMP. (See smp.txt in the Documentation subdirectory of the Linux kernel source for more on how to perform this task.)

The first SMP kernel I compiled did not work correctly. From my reading of the documentation included with 4Front Technologies' OSS sound drivers and in the kernel itself, I realized the dependencies didn't get built correctly. I saved the .config file elsewhere, performed **make mrproper** to clean the kernel source tree, then restored the .config file. After performing **make oldconfig**, I built

again and installed the SMP kernel. On the next boot, I saw additional startup messages to indicate that both CPUs had started running. The 2.2.7 kernel, in conjunction with the utilities shipped with Debian 2.1, report each process's CPU usage as a percentage of the total available. A process consuming all of one CPU will show 50% usage.

Using It

In order to demonstrate the difference in performance provided by a second CPU, I performed benchmarks with Linux kernel compilation, the distributed.net rc5des encryption breaker and with POV-Ray's ray tracer (see Table 1). All take direct advantage of multiple CPUs. POV-Ray can also directly use CPUs spread across a network. All figures represent averages of three runs.

Table 1

Recompilation of the uniprocessor 2.2.7 kernel took 376.91 seconds when running under the same kernel. Recompilation of the SMP 2.2.7 kernel, running under the same SMP kernel, took 395.04 seconds when run on only one CPU, 5 percent longer than the uniprocessor compilation time. When run on two CPUs (**make -j 2 bzImage**), the compilation took 302.77 seconds, 80 percent of the uniprocessor compilation time.

For POV-Ray, I used the benchmark source file, skyvase.pov, available from POV-Ray's web site. I ran it at **xpvmpov**'s default resolution of 320x240. SMP took 72 percent of the time for a uniprocessor run.

The rc5des code cracker performed its benchmark at nearly the same rate under both uniprocessor and SMP kernels. When in actual operation, it will run on as many CPUs as desired or automatically detect the number of CPUs. I believe there were much smaller performance differences between the two kernels because of the optimizations it contains for maximum performance. It most likely runs within the level 1 (L1) cache as much as possible.

SMP may improve performance in other ways. GUI operations may benefit from having the X server run on one CPU while an application runs on another. Anything that runs well on one CPU but can take advantage of another will benefit from using SMP. I now run the SETI@home client on all CPUs I have that run Linux.

Running Even Faster

Both L1 and L2 cache quantity and speed matter. RAM speed matters. The Intel P5-233MMX contains a 32KB L1 cache, distributed as a 16KB code cache and a 16KB data cache. My wife's AMD K6-200MMX contains a 64KB L1 cache,

distributed as a 32KB code cache and a 32KB data cache. For some tasks, it performs faster than one Intel P5-233MMX. Intel Pentium Pro CPUs have both L1 and L2 cache on board, with up to 1MB of L2. Pentium II CPUs have up to 2MB L2 cache on board. New CPUs also run their caches faster. More cache on the CPU means less contention for external cache and main RAM, which means higher performance. The CPUs, through the support chip set, co-operate among themselves to maintain cache coherency, so that they always maintain accurate views of RAM.

Locking a process to one CPU, particularly when that process' code and data fit in the L1 cache, may also improve performance. Linux does not support this as fully as more mature UNIX variants, but it probably will soon.

Conclusion

Do I need SMP for what I do? No. A single 200MHz P5-class processor can adequately perform the tasks I want to perform. As for most tasks, adequate memory, both RAM and cache, contributes more to performance than the number of processors. Do I have fun with it? Oh, yes.

Resources

Michael S. Keller works as a technical analyst with Sprint Paranet, a wholly owned subsidiary of Sprint, a nationwide network services provider based in Houston. He has used UNIX variants for nearly nine years and enjoys communing with cats, motorcycles and the universe. You may reach him at mskeller@sprintparanet.com.

Archive Index Issue Table of Contents

Advanced search

Focus on Software

David A. Bandel

Issue #71, March 2000

note, gnotepad+, WebDownloader for X and more.

By the time you read this, even more restrictions on the export of strong cryptography will have been lifted. I was delighted to hear that PGP was given export permissions, although it still can't go to the T-7 (terrorist seven) countries. I've never understood the contention that cryptography was a munition. The folks (including politicians) who have insisted that the measure prevented terrorists from obtaining it show a complete lack of understanding of how terrorists work. Export of a few more packages (FreeS/WAN and OpenSSH, at least) will result in distributions with adequately secure administration tools. In the meantime, these tools are still available from sites outside the U.S.; just don't forget you can't export the tar files back outside the U.S.

note: www.daemon.de/software.html

note is an easy-to-use utility that allows you to keep notes to yourself in a MySQL database. You can create notes, edit them (using vi, but it will honor your **$EDITOR** environment variable), delete them, list them, search them, etc. This utility would also be the perfect tool for a journal when you need to keep extensive, dated notes over a long period of time because of the MySQL back end. Easily extensible, the front end works from a console or xterm and is written in Perl. This is one of those utilities that causes you to say, “how did I ever get along without it?”--a must-have. It requires Perl, the Mysql-msql module, DBI module and MySQL.

gnotepad+: http://ack.netpedia.net/gnp/

Although the author touts this program as a notepad, he also admits it is much more, thus the “+”. Indeed, **gnotepad+** has all the trappings of an HTML editor. You can customize the button bars to include an HTML set, including only those

buttons you might need. This notepad can create HTML code more easily than most of the HTML editors I've used. If you use GTK and need a good editor, give this one a try. It requires libgtkxmhtml, libgtk, libgdk, libgmodule, libglib, libdl, libXext, libX11, libm, glibc, libXpm, libjpeg, libpng and libz.

WebDownloader for X: http://www.krasu.ru/soft/chuchelo/

This utility gives you complete control over downloads. You can set it up to download http or ftp URLs at night. You can interrupt downloads, then pick them up where they left off. You can limit the download speed from any individual site, or if you're using it while on-line, you can select any of two configurable slower speeds. If you are connected to the Internet via cable modem or DSL and do a lot of downloading, you will want to check this one out. It requires libgtk, libgdk, libgmodule, libglib, libdl, libXext, libX11, libstdc++, libm and glibc.

Yet Another Computer Algebra System: www.xs4all.nl/~apinkus/yacas.html

For all you math geniuses out there, **yacas** provides a convenient, easy-to-use interface for doing math computations. While I don't need much beyond simple addition, subtraction, multiplication and division, I do wish I'd had this 25 years ago while taking calculus. This program can handle Taylor series, Newton method for finding zeroes and several other functions. While the author is still working on improvements (functions with multiple occurrences of variables), this is an excellent program as it is. It requires libstdc++, libm and glibc.

xglobe: http://www.rzstud.rz.uni-karlsruhe.de/~uddn/xglobe/

Well, what can you say about another program that puts a globe on your desktop? I've always been fond of the Earth, and a satellite's eye view of the world is always nice. Shows the planet from the sun's perspective. Complete with names and locations of major cities. It requires libX11, libqt, libstdc++, libm, libXext and glibc.

rubix: http://sed.free.fr/rubix/

It's been a while since I played with a Rubik's cube. What this little toy showed me was that I no longer remember how to solve it. It is not yet the easiest thing to manipulate and certainly not intuitive. Hopefully, the author will work on this aspect. Actually, it's probably more my ineptness at using a mouse than it is the controls. Hours of entertainment for the easily amused (even more for the easily confused). It requires libX11, libm and glibc.

xzgv: ftp://metalab.unc.edu/pub/Linux/apps/graphics/viewers/X/

If you remember **zgv**, you know what **xzgv** is. Since most folks today prefer graphical user interfaces to command-line ones, this provides them with the power of zgv in X. It does an excellent job of rendering most image formats. It requires libgdk_imlib, libgtk, libgdk, libgmodule, libglib, libdl, libXext, libX11, libm and glibc.

tkftp: http://tkftp.firebird.cx/

As GTK continues to proliferate on desktops, it's nice to see Tcl/Tk applications remain. This is especially true for folks with older PCs who want a GUI interface, but don't want to bog down their system with larger applications. This particular Tcl/Tk utility is very handy, easy to use and simple to configure. Connections options are flexible and powerful. This client can handle most, if not all, popular FTP servers with ease. It requires tcl-8.0 and tk-8.0.

David A. Bandel (dbandel@pananix.com) is a Linux/UNIX consultant currently living in the Republic of Panama. He is co-author of Que Special Edition: Using Caldera OpenLinux, and he plans to spend more time writing about Linux while relaxing and enjoying life in the “Crossroads of the World”.

Archive Index Issue Table of Contents

Advanced search

Embedded Linux News Briefs

Rick Lehrbaum

Issue #71, March 2000

The latest news from 3iLinux, TINY Linux, Motoroal Computer Group and more.

3iLinux introduced a low-cost, compact and configurable Linux system that lets OEMs rapidly create Internet devices simply by adding their unique application software. The ready-to-run box includes a 386 CPU, 2x16 LCD, one or two modems, Ethernet, pushbuttons, power supply and boots from DiskOnChip. (http://www.3ilinux.com/)

The latest version of **muLinux** (V7r6) is now available. It's described as a minimalistic—but mostly complete—application-oriented Linux distribution that fits on a single floppy diskette. Minimum system requirements are a 386 PC with 8MB DRAM. Download muLinux from http://sunsite.auc.dk/mulinux/.

TINY Linux (v0.1) has arrived. It's a PC Linux requiring minimal system resources that can be installed easily by Linux beginners. TINY doesn't need a CD for installation, even on stand-alone machines. (http://tiny.seul.org/)

Motorola Computer Group has hired **Linuxcare** to provide Linux services and support for Motorola's customers. Linuxcare will also support Motorola's Linux-related engineering programs, product customization and application development. (www.mcg.mot.com/, www.linuxcare.com)

Moreton Bay announced “the world's tiniest Linux DHCP server”, targeted to embedded environments. The miniscule open-source DHCP server fits in 22KB, vs. 400KB+ for typical DHCP servers. (http://www.moretonbay.com/dhcpd/)

The world's leading developers of real-time and embedded Linux convened in Vienna, Austria for the first annual **Real Time Linux Workshop**. The group reached consensus to develop a standardized real-time Linux Application Programming Interface. (http://www.realtimelinux.org/)

There's a new mailing list devoted to **Linux i386 assembly language** programming. The list address is linux-assembly@egroups.com. To subscribe, send a blank message to linux-assembly-subscribe@egroups.com. List archives are at http://www.egroups.com/lists/linux-assembly/.

Lineo announced that Arriba!, Viosoft's Integrated Development Environment (IDE), will be bundled in Embedix, Lineo's embedded Linux SDK. (http://www.lineo.com/)

Four leading Linux distribution suppliers—**Caldera**, **Red Hat**, **SuSE** and **TurboLinux**—jointly announced an alliance called the Trillian project, which will ensure Linux support of Intel's upcoming Itanium (IA-64) processors when they come to market in the second half of 2000.

WireSpeed Communications announced an Embedded Linux Service which will provide contract development services to companies wanting a rapid way to develop embedded systems based on Linux. (http://www.wirespeed.com/)

After receiving $32.5M in venture capital funding, **Linuxcare** acquired three Linux technology companies including **Prosa**, a well-known Italian consulting firm that specializes in porting Linux to embedded devices. (www.linuxcare.com/, www.prosa.it)

Zentropix began shipping RealTime Linux v2.2, a distribution derived from Red Hat 6.0 and Linux 2.2.10 combined with a choice of either RTAI version 0.7 or NMT RTLinux version 2.0 real-time functionality. The distribution features one-step installation of a full real-time Linux environment. Powerful Zentropix RealTime Linux debuggers are also included. (http://www.zentropix.com/)

LinuxDevices.com launched an on-line automated Embedded Linux Polls center. The polls will monitor trends in the hot emerging market for embedded and real-time Linux. (http://www.linuxdevices.com/polls/)

Unique Broadband Systems unveiled RealLinux, a new real-time Linux distribution that will support non-“x86” processors including PowerPC, 68030, i960 and DSPs. RealLinux will provide real-time extensions along with reduced footprint, allowing execution from ROM or Flash. (http://www.uniquesys.com/)

Amino Communications announced “the World's Smallest Linux System”. The system is implemented on a tiny PC board (2.0 x 4.0 inches) and is optimized for Internet-enabled set-top boxes, web phones and embedded devices. According to Amino, the board costs less than $100 to manufacture. (http://www.aminocom.com/)

Coollogic acquired **ON Channel**, a developer of embedded Linux applications. ON Channel simultaneously announced availability of the E-Pilot, a network appliance it calls “one of the first real-world embedded Linux applications”. The 180MHz MediaGX-based device is compact (11 x 9 x 2 in.) and supports a broad spectrum of communications interface options. (www.coollogic.com, onchannel.com)

An **Embedded Linux Consortium** is being formed. The non-profit multi-vendor association will foster the rapid proliferation of Linux in embedded applications. An organizational meeting is planned for ESC Chicago (Feb. 28 - Mar. 2). Preliminary plans and discussion can be found at http://www.linuxdevices.com/forum/.

Bristol Technology introduced Wind/U for Linux, a set of cross-platform development tools which simplify the process of converting Windows applications into Linux applications. (http://www.bristol.com/)

KYZO released the commercial version of its PizzaBox Linux distribution. It supports Linux file, print and CD thin server functions and boots from an M-Systems DiskOnChip. NASA has used KYZO's open source PizzaBox distribution in two applications: at Goddard Space Flight Center and the Jet Propulsion Laboratories. (http://www.kyzo.com/)

Belgium-based **Lernout & Hauspie** announced Linux SDKs that enable developers to incorporate speech and language offerings in Linux-based applications for hand-held and embedded devices. (http://www.lhsl.com/voicexpress/)

FSMLabs released v2.0 of RTLinux, a popular Linux distribution with the “hard real time” functionality needed for machine control and mission-critical applications. The new version supports POSIX API and device drivers, symmetric multi-processing, and has highly optimized timings. (http://www.fsmlabs.com/)

Lynx Real-Time Systems, renowned for their proprietary UNIX-like LynxOS “hard real-time” OS, unveiled BlueCat Linux, a derivative of Red Hat Linux for embedded applications. Additionally, Lynx said they will offer a future version of LynxOS that will run Linux applications (unmodified). (http://www.lynx.com/)

Rick Lehrbaum (rick@linuxdevices.com) co-founded Ampro Computers, Inc. in 1983, where he served for 16 years in the roles of VP of Engineering, President and Executive VP of Strategic Development. In 1992, Rick formed the PC/104 Consortium and then served as its chairman through January 2000. In October, 1999, Rick turned his attention to embedded software, founding his second

startup: LinuxDevices.com—“the Embedded Linux Portal”. Rick received his BS and MS degrees in Physics from NYU and Northeast Louisiana University, respectively.

Archive Index Issue Table of Contents

Advanced search

Unit Safety

Stan Kelly-Bootle

Issue #71, March 2000

Accidents, by definition, happen in all walks and trips of life....

The recent spate of NASA upsets in the Martian environment reminds us that, in spite of the many “provable” (sez who?) advances in academic risk analysis and software development, our practical trade remains subject to the most bizarre glitches. Accidents, by definition, happen in all walks and trips of life, but once having happened, are revealed by post-hoc scrutiny to have been eminently, nay, trivially “preventable”. Next time, we boast, we'll avoid spilling the salt, we'll bypass the banana skin under the ladder and we'll secure the loose horseshoe that lost the last war. More space-age specifically, we'll watch out for inelastic rubber washers, shun rash global identifiers, and work around the quirks of FORTRAN FOR loops, all of which have proved to be life- and dollar-threatening.

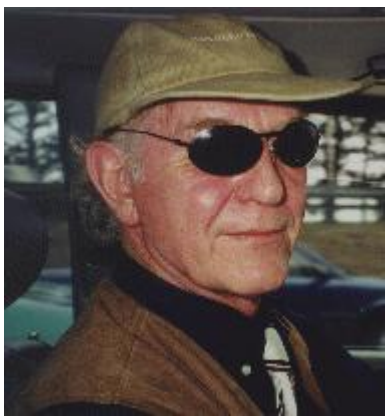
Of course, digging deeper into fashionable post-Darwinian theory, we must acknowledge that “social evolution”, without which this column might have emerged as a mysterious henge in some foreign field, is due to a long sequence of fortuitous accidents. Random memic mutations, some say, guarantee the survival of fitter crania, neural paradigms, languages, algorithms and software engineering methodologies. Whether we consciously “learn” from our mistakes or not, some argue that there is an ongoing natural “self-correctional” process that converges to error-free nirvana. Fat chance! The optimistic concepts of “inevitable improvement” and “human perfectibility”, fashionable when the previous millennium started [see Note], are hard to relinquish, although the alternatives are so damned depressing, leading to a self-fulfilling why-bother apathy.

The challenge for historians is selecting and recording “significant” events. The so-called random elements (Luther's constipation, Napoleon's insomnia, Hitler's rejected watercolors...) seem to have a ridiculous, chaotic impact on the *grand deéroulement*. My favorite historian, Barbara Tuchman, warns against

the *post hoc* contingent narrative, but it's impossible to feign ignorance of what happened next. What if, say, Saint Linus had wealthy parents like Bill Gates? What if he had not grown up in the fractal Swedish-Finnish isogloss?

One of the most expensive, embarrassing lessons from the missing Mars probe mishaps was the lack of “unit safety”. We've spent all these years ensuring type-safety (not to mention the oxymoronic thread-safety), whereby oranges and lemons cannot be added without deliberate, fruitless miscasting, yet it seems that a given floating-point value was interpreted as *Newtons* rather than *pounds*. Or vice versa. Or was it an understandable confusion between parsecs, meters, yards, chains, rods, perches and poles? We'll get it right next time, unless there are bright Martians anxious to resist our inferior technology.

Note



Stan Kelly-Bootle (skb@crl.com) has been computing on and off since his EDSAC I (Cambridge University, UK) days in the 1950s. He has commented on the unchanging DP scene in many columns (“More than the effin' Parthenon”--Meilir Page-Jones) and books, including The Computer Contradictionary (MIT Press) and *UNIX Complete* (Sybex).

Archive Index Issue Table of Contents

Advanced search

Letters

Various

Issue #71, March 2000

Readers sound off.

Going to the Chapel or Shopping?

Eric Raymond's *The Cathedral and the Bazaar* (reviewed by Peter Salus in the January issue) is a fascinating book; I could hardly put it down. Some thirty years ago when I suggested at Boeing that software should be distributed in source-code form, the idea was hooted down and rejected out of hand. Eric's book documents the fundamental validity of that idea and records for us how it took root, and now provides the very direction of software development.

Much as I like Eric's book, it has side tones that could use a filter. A clue to the source is revealed, I think, on page 39 where Eric tells us that in 1992, his attempt to get some code merged into the Emacs Lisp library was rebuffed by the Free Software Foundation (FSF). So Eric has kind of a sour-grapes attitude toward the FSF, and I think it has caused him to miss-read (sic) the real relationship of Richard Stallman's *cathedral* approach to Linus Torvalds' *bazaar* approach to software development. Eric tells us Linus' approach is “much healthier” and “the very opposite of cathedral-building”. But Eric is comparing an apple to an orange.

Linus released his kernel sources into a world brimming with youngsters who were absolute UNIX experts. Some of them probably knew more about UNIX bedrock than Linus did. Linus lifted them over a formidable obstacle. Intel's 386 was equipped with precisely the hardware UNIX requires, but it was kludged up to be backward-compatible all the way to day one, so the hoops you jump through to get it started are just a bit *hairy*. These UNIX experts created quite a different situation for Linus from what Richard Stallman had when he developed his C compiler or the Emacs editor. The UNIX world was not brimming with text editor experts, let alone compiler experts. It doesn't work to

share development with people who know little about what you are trying to accomplish.

When Eric comments that “FSF was not the only game in town”, I think he unnecessarily gives short shrift to the work Stallman has contributed to the free software movement. The reality is, without Stallman's compiler and make director, not to mention Emacs and gdb, there might not be *any* game in town.

Software tools do come first, just as Bob Canup pointed out in these pages a few issues back. Toolmakers, such as Dennis Ritchie and Richard Stallman, have been awarded special status in the programming community, and rightly so. The real relationship of the cathedral to the bazaar is not antagonistic at all, but complementary. As the work of Dennis Ritchie stands in relation to that of Ken Thompson, so the work of Richard Stallman stands in relation to the work of Linus Torvalds. As a poet long ago summarized, they are “...useless each, without the other”. We need both the cathedral and the bazaar.

—Jack Dennon jdennon@seasurf.com

Desktop Wars

I look forward to *Linux Journal* every month. It is one of the few publications I take time to read completely. Please keep up the good work.

I have been following the thread of KDE vs. GNOME, Red Hat vs. others, etc. for the last few months. My personal opinion is that many of the advocates of particular desktops or distributions are to some degree missing the point. I do have my personal preferences, but I believe what is really needed is standardized file formats. That is, if I generate a “document” in Applixware or StarOffice or whatever, it must be readable by the other office suites. Also, enough design has to go into the file formats so that I do not have to purchase an updated version of the software every two years. This rationale applies to spreadsheets, presentation software, etc. It would greatly enhance the portability of documents between the various office suites and/or distributions (even other operating systems). The actual desktop (KDE, GNOME, etc.) would then be the preference of the individual using the system.

I know this may have the short-term effect of limiting some creativity, but it would be a solution to one of the problems created by the other popular OS. Just my two cents worth.

—Dave Underland dunder@earthlink.net

Hurrah for Feathers

Loved the January cover! I watched the “Wallace and Gromit” special on the tube, and cracked up when I saw Feathers McGraw. A bit Tux-like, isn't he? I have also spied a *very* Tux-like fellow during reruns of “Reboot” on the Cartoon Network. This character pops up from time to time and casually strolls by. I wonder if the folks at Mainframe Entertainment are sending out a subliminal message?

World Domination!

—Patrick Murman pmurman@earthlink.net

Wrong Earl

“Distribution Watch” by Jason Kroll in “upFRONT”, January 2000, was a most enlightening article. However, the given URL appeared to be www.khaOS.org, while it's actually www.kha0S.org (note the O/0). The former URL, which I assumed was correct upon reading, directed me to “Diario El Mundo del siglo XXI”, which I was able to infer through my minimal knowledge of the Spanish language was not KhaOS's web site. Though I found the “Diario El Mundo del siglo XXI” most singularly intriguing, it would be appreciated if your URLs were a tad clearer.

—Auknight Colather auknight@postmark.net

Sorry about that. Depending on the font, a capital O and a zero can be difficult to distinguish —Editor

Light Up the Bat Signal

I really enjoyed your interview with Linus in the November issue of *Linux Journal*. How refreshing to read about a famous person who is normal and intelligent! I love the questions you chose to ask, as well as the cover title you chose—very humorous.

—Kimberly Guardino kimberly@zna.com

Lydia Kinata came up with the “Linus is Batman” cover. We like it too —Editor

Thanks for the Magazines

I am writing to you to say thank you for the copies of *Linux Journal* you have sent to me. I am a researcher and computer programmer at the Higher

Politechnical Institute J.A.E. of Havana (ISPJAE) and specialize in writing programs for industrial use (most in Delphi).

We are not familiar with UNIX systems, because these are not very common in this country. In my university, only a few people have installed Linux, and some communication servers run UNIX.

Most people have Windows 95, 98 or NT. In our country, it is easier to find an installation CD for Windows than one for Linux, and most available programs are for the Windows platform. Finally, at the universities and almost all over the country, no one has to register and pay for Windows (I do not know what will happen with this situation in the future). Therefore, people don't have to worry about cost, and Linux being freely available is not a factor in decision making.

Nevertheless, I have installed Linux (not without some hardware problems) on my computer and find it very nice and powerful.

Anyway, thank you again for the journals. They were very interesting.

—Melvin Ayalajre yes@ceis.ispjae.edu.cu

Archive Index Issue Table of Contents

Advanced search

UNIX Systems Diagram

The UNIX Systems Diagram on page 10 of *LJ* Issue #69 is missing a UNIX flavor that can be expected to influence the future of Linux. SGI's IRIX is not included in the diagram, although on page 8 it is reported to run more web sites than AIX or HP-UX, which are included.

What is significant about IRIX is SGI's plan to phase out IRIX over the next few years and move completely to Linux on their hardware. You can bet that they will be contributing more components of IRIX to the Open Source community. They have already started contributing some IRIX code, starting with their XFS high-performance journaling file system. I'm waiting for their 4Dwm to be ported to Linux; their Desks Overview has a better multi-window management tool than others I've tried, such as fvwm2, KDE, or AfterStep.

—Mark Plimley, markp@blueneptune.com

Re: PNG Book review

Thus spoke Ian Clarke, I.Clarke@strs.co.uk

> I was somewhat concerned to notice that a link is provided above your online review of this book to that books page in Amazon's online store. This link seems to inform Amazon that the referral was provided from *Linux Journal*. I know that Amazon do reward people who encourage others to buy books from them, and I must say this makes me somewhat suspicious of your glowing review of this book, and also calls into question *LJ*'s journalistic integrity.

I don't have anything to do with how *LJ* does their layout, either in the printed magazine or online. I also don't get anything from Amazon with respect from the link you mention. I wrote the review for a stock price with no input from *LJ* on what that review should contain. I wrote the review without influence as to content from either Amazon or *LJ*. It's just a damned good book.

> Does *Linux Journal* profit in any way from people buying this book through Amazon having followed the link from your review? If so, how do you justify disguising what is essentially an advertisement as an impartial review?

I don't work for *LJ*. I'm a freelance writer. They ask if I can do a book review. If I say yes, they send me a copy (unless I already have it, which I'd already purchased a copy of this particular book *before* they asked me to do a review of it). They don't tell me what the review should say—it's completely up to me. And it's always been that way. Even before on-line partnerships with resellers like Amazon came along.

Additionally, you assume there is a profit motive in making that link. No one is getting rich off of “associates” programs with Amazon or other book sellers. It's a convenience thing for readers more than anything else. I doubt *LJ* would notice the drop in income if they removed the link.

It's not a disguise. It's a good book. That's why the review is so glowing. If you'd read any other technical books on Linux you'd know this book makes them all look like children had written them. Greg is a very good writer.

> Further, given Amazon's recent exploitation of the American Patent system, something that is anathema to anyone who supports the principals of Open Source software, *LJ*'s relationship with them appears even less well thought-out.

I don't tell *LJ* who they should associate with, and they don't tell me who I should. I don't like patents much, but I'm not boycotting Amazon because of patents. I simply don't buy books from them for the same reason I don't buy computer equipment from Wal-Mart.

If you disagree with the review (i.e. you think it's not a good book), that's one thing. But my involvement with how it's displayed or associated with advertising within *LJ* is non-existent. I suggest you take this up with *Linux Journal* directly.

—Michael J. Hammel, mjhammel@graphics-muse.org

> *LJ* does receive some sort of pittance if someone buys the book clicking through us. But we never tell our reviewers what to say. The link is there on both good and bad reviews.
> —Editor

Linux GUIs

I just read the article on GNOME and Its Future. Some comments:

1) The OS interface must be usable without a mouse. I have not found any GUI interface for Linux that meets this criteria. Even Microsoft has keyboard equivalents in its various Window OSs. People talk about right/left clicks like a keyboard never existed. This may not seem important, but work can be more efficient if the mouse is used sparingly.

2) The abhorrent focus on the Internet has obscured the vision of product developers. Many people do not have Internet access and never will unless it is totally free. Internet-related modules should not be loaded by default nor should they be an integral piece of the OS. For example, Red hat's installation loads “chat room”-related modules/packages by default. Why? Is everyone supposed to use “chat rooms”? I do not think so.

3) Personally, I do not like the Internet-style interface. I do not have Internet access at home and will not until I can get DSL for Linux at an affordable price. I do think the Internet can be a useful part of people's lives, but it needs to alter its direction somewhat.

4) I use a computer for computing not entertainment. I use Linux for computing not entertainment. If I did not care about computing I would use a Windows product. Linux development seems to be emulating Microsoft rather than continuing to build/improve a real OS.

—Burk, Galen G CHPPM-Wash DC, Galen.Burk@NA.AMEDD.ARMY.MIL

Meaningless Survey

As a statistician I am irritated by those surveys you are publishing since several issues and specially by your Readers' choice awards. The method of the survey is completely flawed.

What you are using is accounting the votes of people who go to your site. This has a name in statistics “straw ballots”. The problem with them is that they are very sensitive to the problem of vocal minorities spreading the word and massively voting to the straw ballots. In the 1932 US presidential elections they predicted a crushing victory of the republican candidate while the first Gallup polls predicted Roosevelt's victory and got very close of the percentage of votes he got. Since then no serious statistician uses straw ballots or gives a dime for their results.

The Gallup method consists in _you_ selecting a fair sized sample of your readers and asking them to answer a questionnaire. While militant types will have a higher answer rate and thus the results will have a certain bias towards the software they support this is far less than the bias in straw ballots.

So, please if you can't use Gallup polls at least include a BIG disclaimer warning the reader about the inaccuracy of the results.

—Jean Francois Martinez, jfm2@club-internet.fr

> Actually, we don't use a disclaimer on the Reader's Choice awards, because we feel our readers have the intelligence to figure out the bias of straw polls themselves. Doing the survey on the web is cheaper, easier and more convenient for both our readers and us than doing a mail-out survey.
> —Editor

IBM Ad in your Jan 2000 Issue.

I just received my Jan 2000 issue of *LJ*. I have just one question. Why is Tux made to look like a Chicken in the IBM 3D Ad? The fact that he looks like a chicken makes the “World Domination” theme confusing. Are they talking about “World Domination” for Linux or KFC?

—Jason Lawrence, jason@19.com

Stock Info!

First let me say *LJ* is GREAT!

I have been following the different Linux-based stocks and have been reading the articles on the Linux insiders and startups. I thought; it sure would be nice if somehow *LJ* could cover one or two companies from the perspective stock holders point of view. It would also be good for investors and the companies advertising in *LJ* if there stock symbol was listed in the advertisers address information. For those who are looking for investment capital. It seems that I remember a brewery in New York where the owner went public on his own and did very well. Maybe a story on something like that would help small Linux programmers to get a foot up with there work.

Just some thoughts,

—Fred Walker Jr., fastfred@wil.net

---

re Feathers McGraw

I PROTEST!!!

How dare you make Feathers McGraw an honorary member of the Linux community (“About the Cover” *LJ* 1/2000) on the basis of his goal of World Domination!

Not only do I not recall his goal as being one of world domination (though it may well have been), but he was an out-and-out crook: in Nick Parks' “Wallace & Gromit - The Wrong Trousers”, Feathers McGraw stole Wallace's robotic trousers (with Gromit in them) and then used them to help him steal a big diamond!

Feathers McGraw is a thief, an verifiable criminal (a master criminal according to http://www.aardman.com/wallaceandgromit/homesweethome/feathers/index.shtml), and you want to bestow upon him the great honour of an honorary member of the Linux community?! I am both shocked and horrified!

Feathers McGraw should be seen for what the really is: an attempt by subversive evil forces, such as those found in a certain part of Redmond, Washington, U.S.A., to infiltrate and corrupt the near untouchable ranks

of our great Linuxdom. Let us, therefore, close ranks to keep such potential infiltrators and perpetrators of great mischief out!

Enjoy 2000! Graeme Nelson

P.S. It is good to see the even the eminent Phil Hughes makes typos! In his “Watch Out For The Snakes” letter (*LJ* 1/2000), he calls Python “easy-to-debut”: as things usually only debut once, I guess his finger slipped upward and hit the <t> instead of the <g>. You see, just 2 pages after awarding Feathers McGraw honorary Linux community membership, his is already working his heinous schemes of decay, keeping Phil from declaring Python to “easy-to-debug” and thereby encouraging untold millions away from M$ products. Instead Phil declares it to be “easy-to-debut” and so causes it to be viewed with suspicion, or even worse, causing it to be viewed as mediocre, and therefore equal to or worse than M$ products.

—Graeme Nelson, graeme@cheerful.com

---

jan 2000 cover

Good mag; I'm well pleased with my subscription but... er.... your cover for Jan. Umm.... you do realize that's the evil penguin from Wallace and Grommit! Seriously though, thanks to all for an excellent mag. Keep up the good work.

Cheers

—K Clethero, k.clethero@taranaki.ac.nz

Readers choice awards

Maybe for those of you with Pentium III's mpg123 is THE audio application, but for those of us with more modest; computing power amp (and xamp) work much better. Also I think somewhere mention should be made of the 2 things that make text Linux preferable to GUI namely gpm and command completion

—mraivio, mraivio@oln.com

Linux Journal Letter to Editor: “World Domination in 1900”

So, Mr. Smartipants, unless you are playing a “Ha ha, only serious!” joke on us, I think you have some explaining to do. Your January 2000 issue has “January 1900” footers on some pages (with red-line correction). Embarrassing, after joking “Y2K compliant & certified” on the December issue cover.

Regards,

—Walter Cooke, wcooke@paragon.bm

> The December cover and the follow-up date in January were jokes. We wanted to poke a little fun at all the Y2K uproar and at ourselves.

Why the Y2K explosion was a dud.

On an old PC with Linux, I entered the following New Year command:

```
% clear ; date ;
```

and what appeared on my screen was the following:

```
% 01-01-00
% ? ... ?
% Mon Jan 1 9:49:22 EST 1900
% No electronic computers existed in the year 1900.
% I compute, hence I exist.
% Sat Jan 1 9:49:23 EST 2000
% Descartes ... eat your heart out.
```

All this in less than a second of computing time.
Allan Widom

—Allan Widom, widom@acausal.physics.neu.edu

LinuxPPC vs PowerPC Linux]

In the Forum, page 112, February 2000, Stephane Morvan opens his presentation of PowerPC Linux “flavors” by stating “LinuxPPC, along with MKLinux and NetBSD, are the three flavors of Linux that can be found for Macintosh computers.”

There is a major flaw in this statement that has been overlooked by the author of this article, *Linux Journal*, and the media in general over the past year. “LinuxPPC” was once used in reference to the “Linux for PowerPC” development community. However, with the introduction of the for-profit company LinuxPPC, Inc., a general confusion and lack of separation between the development community and the commercial product resulted. This is compounded by the fact that LinuxPPC, Inc. has failed to release the www.linuxppc.org domain to the PowerPC Linux development community and continues to link to their specific commercial products.

In an effort to regain autonomy, the once “LinuxPPC” development community has migrated to the “PowerPC Linux” development community. This title needs to be recognized and supported in order for every commercial venture offering Linux for PowerPC to be truly competitive, offer quality products to Linux consumers, and most importantly, make certain the development efforts of the community are readily recognized as independent of the commercial ventures.

As Caldera, SuSE, and Turbo Linux have fought an uphill battle to gain recognition from the media who continue to focus, and sometimes use

the term “Red Hat” in reference to Linux in general (or even the reverse where “Linux” refers to “Red Hat”), we simply ask that proper recognition be given for all distributions and development efforts.

—Kai Staats, kstaats@terraplex.com

> We try to catch blanket statements made by authors but sometimes we miss. Sorry about that. In our April Distribution Watch, we will discuss all the various flavors of Linux of the Power PC. Thanks for writing.
> —Editor

Desktops of the Future

I was pleased to see “as we may think” mentioned by Mr. Salus. However Memex, the hypothetical machine described by Dr Bush, was not a general purpose computer. It was a database appliance which would literally search through a file cabinet, locating records by filenames, just as a human file clerk would do. What's interesting about Dr. Bush's paper isn't the machine, which he contrived as an example of what might be possible with 40's ear technology, it's the system of linking documents together with key words and targets thus making it possible to search the database without looking at every file. Now that we've all been exposed to the web for a few years clicking a link to find related documents doesn't seem unusual but in 1945 it was a fairly new and exiting idea.

“As we may think” has been donated to project Gutenberg, and the entire text of the article is displayed on many web sites. It's an important part of our history, I recommend it to anyone who is interested in the how & why of science and technology. http://www.isg.sfu.ca/~duchier/misc/vbush/vbush.shtml

—N.Masters, enormas@gj.net

Grief with LinuxCad

I don't know if this is the right forum for this, but hopefully it will save other people some grief.

Software Forge Inc., which advertises LinuxCad in the *LJ*, falsely states the capability of LinuxCad. On a direct question through email whether LinuxCad supports STL output format, Software Forge Inc. lied and said “yes”. Further, software Forge's web page falsely states that LinuxCad contains equivalent 3-D capability as AutoCad, which it does not. On top of that, many of its existing functions don't work. After many unanswered email and phone calls (request left on a answer machine) I got the the email response “Wait until the next release.”

I have requested a refund.

I have purchased quite a few commercial software for Linux (Applix, Star, VMware, Motif ...) which have all been top-notch. This is the first product that has come close to being a scam.

On another note; I have being using Linux since 1992 with my first copy a TAMU distribution. The *LJ* is a good magazine and does well on presenting to a wide, diverse audience.

Fill free to edit this and do as you wish with it.

Cheers

Dale M. Snider, dsnider@nmia.com

Archive Index Issue Table of Contents

Advanced search

UpFRONT

Various

Issue #71, March 2000

Stop the Presses, *LJ* Index and more.

STRICTLY ON-LINE

An Introduction to Using Linux as a Multipurpose Firewall by Jeff Regan is exactly what its title suggests. Whether you want a firewall at home or office to give you the security you need in order to stop worrying about crackers entering your system, this article tells you just how to set it up. From configuration to locking it down, all the details are here.

Network Monitoring with Linux by Tristan Greaves is another introduction, this time to a freeware software package called NOCOL, designed to keep your system stable without endangering the security of your system. NOCOL does not need to run as root. Complete instructions are given for installation and configuration, as well as final tweaking to get things running smoothly. NOCOL will analyse your system and keep you informed on how it is running.

LUIGUI—Linux/UNIX Independent Group for Usability Information by Randy Jay Yarger has a long and descriptive title. LUIGUI is a new Linux group that has been organized to look at user interfaces and help formulate a standard in an effort to ease the way for Linux to move onto the desktop. Find out all about it and how you can help.

UNIX Shells by Example is a book review by Ben Crowder. Ben describes this book as a "must-have" for those wishing to learn shell programming. Learn why by reading his review.

STUPID PROGRAMMING TRICKS

Welcome again to another sporadic episode of Stupid Programming Tricks! Fate conspires against our would-be monthly column, but we get fired up to do it again. Last month we recklessly forked and killed processes to play midi files,

before burning CPU cycles like venture capital by playing MODs and S3Ms with the MikMod library. If we're quite clever, we can figure out how to put the simple playmidi or MikMod calls into, for example, the scrolltext demo from last December. Well, it *would* look cool. Still, since we've touched on audio already, let's finish up with it before we get into something else exciting.

Digital audio in Linux comes to us by way of /dev/dsp, which shows up as a file but is actually an interface to your sound card. The kernel interface makes dealing with /dev/dsp fairly easy, if a tad latent. You just open it as you would a normal file, set some parameters, and make ioctl calls, a bit like filling address and data registers before calling a library function in assembly code, not that anyone would do *that* anymore... (haha, what was it, a whole year ago you last used asm?) So, we get all the thrills of appearing to do something exceedingly clever, while we're actually just following procedure. The audio half of Linux multimedia *does* exist, and it *is* easy to use; it's just been a tad ignored on account of visual preoccupations.

If you wish to make sound truly from scratch, you must first invent the universe *and* compile your kernel for sound support (or insmod the right module with correct IRQ and DMA values). Hopefully, you already have a universe and sound support (find out with **cat /dev/sndstat**); otherwise, prepare to be frustrated. Compiling your kernel for sound support is a royal pain, so check the Sound-HOWTO and perhaps also the Kernel-HOWTO. For now, let's assume (read: really, really hope) you've already got sound working.

The first thing to do, when you want to use digital audio, is to open your audio device, which is accomplished by using **open** on /dev/dsp. We'll start simply with playing sound, rather than recording and playing back, so we just need to set the **WRITE_BITS** (8 or 16), **WRITE_CHANNELS** (mono or stereo) and **WRITE_RATE** (typically 8000Hz, 22050Hz or 44100Hz). For clear sound quality, having 16 bits is most important (exponential quality improvement for linear CPU cost), followed by sampling rate (linear quality improvement for linear CPU cost), followed by stereo (enables cool effects at double the CPU cost). Obviously, this is a gross generalization and everyone knows we have to balance the elements, so I recommend 16 bits at 22KHz mono for optimizing performance for CPU cost. However, unless you have a computer from the neolithic, you can afford full quality stereo.

The way audio works is rather simple—all sounds are just collections of different frequencies. You can break down essentially any periodic function into a series of sine functions, the technique known as Fourier analysis. Conversely, you can *create* anything out of a series of sine functions. At a very simple level, if you want to hear a pure 220Hz tone, just play a sine function that repeats 220 times per second. To play an octave higher, just play a sine function that

repeats 440 times per second. To play an octave chord, add the two functions together. (If you have a graphing calculator, you can add sines of different periods together and see the results.) This is additive synthesis, a simple, resource-intensive idea that is also the most powerful and flexible synthesis technique. I thought I'd share that with you, since we'll use simple additive synthesis in our demo to generate a wave table to play via /dev/dsp.

How does this work? Your speaker vibrates according to the signals it receives from your sound card, and as anyone who lives with dying appliances knows, vibrations make noise. If the speaker moves forward and back in a perfect sine pattern many times each second, you'll hear a pure tone at the frequency corresponding to the speed of the impulses (440 times each second would be A 440, the most common frequency of tuning forks). So, the values in digital audio are just amplitude data for the speaker, and these values are ultimately just composites of many, many sines. When dumped to the speaker, these generate complex tones, producing familiar sounds like human voices, snare drums and brass ensembles. All digital audio, including CDs and mpegs, works this way.

In our example code, we'll generate an additive wave table of a chord using sine tones. The function for equal-tempered, 12-tone intervals is simply freq*(12th root of 2)<+>*n*<+>, where *n* is how many intervals up you want to go from freq, your starting frequency. For an A major chord (meaning the 1st, 5th, 8th and 13th tones of the 12-tone scale, the 13th tone being an octave on top) starting at 220Hz, these are our values: 220 277 330 440.

We'll generate the wave table by adding the sines together. (Remember, our wave table contains 44100 16-bit values (88200 bytes), which is exactly 1 second of audio data at 44.1KJz 16-bit mono.) Then, we'll open the digital audio, loop for a few seconds while playing our chord, then close up shop and go home. By replacing our wave table with an audio file, we could add a sound effect to a game, such as the “intoxicating” sound that occurs after blasting one of the turrets in Fleuch. The code for our project is in Listing 1.

Compile with

```
gcc -Wall -O2 sound.c -lm -o sound
```

meaning gcc, warnings all, optimization level two, from the source sound.c, linked to the math library, producing executable object named sound.

—Jason Kroll

Listing 1. sound.c

```
#include <math.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <linux/soundcard.h>
int main(void)
{
  unsigned short int wave[44100]; /* our wavetable */
  int c; /* a counter */
  int out; /* audio out */
  /* open /dev/dsp for write only */
  out = open("/dev/dsp",O_WRONLY);
  /* now we use ioctl to set audio quality */
  c=16; /* 16 bits */
  ioctl(out,SOUND_PCM_WRITE_BITS,&c);
  c=1;  /* 1 channel */
  ioctl(out,SOUND_PCM_WRITE_CHANNELS,&c);
  c=44100; /* 44.1KHz */
  ioctl(out,SOUND_PCM_WRITE_RATE,&c);
  /* this generates the wavetable of our sines
   * it's standard trig, so play around with
   * whatever crazy equations you want to hear!
   */
  for (c=0; c<44100; c++) {
    wave[c] =8000*sin(220*2*M_PI*(c+13)/44100);
    wave[c]+=7000*sin(277*2*M_PI*(c+29)/44100);
    wave[c]+=6000*sin(330*2*M_PI*(c+41)/44100);
    wave[c]+=5000*sin(440*2*M_PI*(c+67)/44100);
  }
  /* now we write the wavetable to /dev/dsp
   * as though writing to a simple file
   * we'll loop for 5 seconds of sheer joy
   */
  for (c=0; c<5; c++)
    write(out, wave, sizeof(wave));
  close(out); /* close /dev/dsp ! */
  return 0; /* satisfy gcc */
}
```

This listing is available by anonymous download in the file ftp.linuxjournal.com/pub/lj/listings/issue71/3836.tgz.

GAME FOCUS—>MAME Emulation of 1951 games

Imagine if computers had a use: if you could walk in, turn one on, and use it for something. Case in point: suppose that on your wall lived an enormous, flat panel monitor, and controlling it, a Linux box—very fast, faster even than normal. What to do? Revel in high culture! What's that, you say? Play games, that is, bigger and better than you've ever seen them before. What games? Antique games. Vintage games. Aged rarities. Far more than just games, these are cultural entities, which so many of us grew up with, and which once resided in arcades and pizza parlors when games cost a quarter to play. As is well-known, aging does a lot for software. Unlike people, who grow old and die, computer programs gain in energy with age, and when double-compiled and then stored in silicon casks for several years, the bits reduce and compress, leaving stronger elemental semiconductor properties, while taking out the fire. A fine vintage game is more fun than vi!

Oh, but arcade machines were often 6502-based, or even 68000-based (and a host of others); at any rate, not x86. How can we possibly run these programs, and where *are* we going to get the code? Fortunately, in 1997 a very cool Italian hacker by the name of Nicola Salmoria began a project to write an emulator for old hardware in order to preserve classic games so that they would never be forgotten, but enjoyed and shared around the world. Within a short time, MAME had grown to support over a thousand games, with 1951 at last count.

MAME can be played in X or console mode, and on several different platforms (including DOS, Macintosh and Amiga). You need to have a fairly fast machine in order to get the best performance on complicated games, but the farther back in time you go, the less processor-intensive the games are. If you have anything halfway decent in terms of hardware, you can enjoy the finer elements of computer culture.

The one problem is that MAME is only an emulator; it doesn't include the ROM images. You have to get ROMs, but apparently it is quite illegal (a felony) to duplicate these things. Hence, instead of openly distributing these images, we have to make secret hunts over the Web to find them. Fortunately, it's very easy to find huge ROM repositories, even though Dave's Classics, which used to be the most complete location, was coerced into taking the ROMs off-line a couple of years ago. Major gaming companies are upset with ROMs, probably since they see emulation ultimately encompassing the console games they currently make (or maybe they realize the older games are superior in game play anyway). Either way, it's a major greed issue. Many of them have taken steps to make *emulation* itself completely illegal, which would be such an abuse of government authority that it rather invalidates any arguments against ROMs based on “moral” or “ethical” grounds, since holding civil liberties hostage at corporate or economic whim is probably not ethical, and one who is flagrantly unethical has little validity to impose moral censure on others. But then, we've seen it before when lobbyists tried to make mpegs illegal, simply because the record industry might lose money, and without any regard to the technological implications of outlawing algorithms for being too efficient, let alone the civil libertarian issues involved.

Nevertheless, every now and then some fool makes a big fuss over emulation and how immoral it is, and tries to spoil everyone's fun, so we're forced to distribute games underground. I've always considered copyright law to be illegitimate. I even opted out of becoming a professional programmer because I decided it would be immoral for me to write proprietary software, so you can imagine where I stand on this issue. Nevertheless, the world is swarming with particularly evil people in whom the money drive has taken hostage all forms of reason and compassion, and these people are legally allowed to lock you in an iron cage for five years if you get convicted. In addition to stealing your life, the

feds can abscond with $50,000 (or perhaps more) for every count. How's that for thievery? Maybe “piracy” is a better name for how the government behaves than for the act of duplicating bits. But I digress.

Games, games, games! We can have so much fun as long as no one catches on. Fortunately, even if the Web goes down, so many people have stockpiled ROMs and burned them onto CDs that you can probably get them from anyone. But what can we do to protect our hobby?

One technique that works is cyber-squatting; no, not bad-faith domain name squatting, but merely claiming the unused “property” as our own by using it. People took over abandoned farms after the plague, and no one complained. By mixing their labor (<\#224> la ESR's perennial favorite, John Locke), they made the farms their own (and MAME was certainly a lot of labor). Or we can use the law of the sea, sunken-treasure metaphor, whatever ways we can possibly think of to justify it! If we continue in this vein for long enough, the so-called “property” will fall into public domain on account of disuse. (The squatter tradition ensures that wasted resources are put to good use, and in the case of video games, these wasted treasures could bring so much entertainment to so many.) Get ready for homesteading the arcade-o-sphere.

Another technique is one that was most effective in the DeCSS situation. One Slashdotter called it the “whack the mole” phenomenon, whereby every time one site is closed, two more pop up. (The name comes from a popular carnival game where in poor little plastic moles get bopped.) What with Don Marti's Great International DVD Source Code Distribution Contest, maybe we'll find some truly clever ways to get information passed around. As with DeCSS and mpegs, authoritarians soon found it was impossible to win, and either gave up (with mpegs) or lost (with DeCSS). Commercial forces will fight hard on this one, and the situation is a bit unclear, nowhere near as pure as the GNU/Linux movement. Maybe it will even make the movement look bad if we participate in ROM trafficking, but we as a society have an interest in seeing that these games are preserved. After all, they are part of the culture of a developing technology and relate much about our times.

If, of course, you believe in “intellectual property” and that copyright restrictions have to be respected no matter what, or you just don't want to commit felonies, what should you do? Well, rather than writing hate mail or telling the teacher, just ignore MAME and go develop some completely free wares under the GPL. If you *really* have to say angry things and call people “pirates” and whatnot, there is a place for people like you: Usenet.

For the rest of us, let's dig in (archaeologically/anthropologically speaking) and enjoy! **www.mame.net**

—Jason Kroll

LJ INDEX—March, 2000

1. Position of Procter & Gamble as a "branding" company: **#1**
2. Number of Procter & Gamble brands: **>300**
3. Percentage of U.S. households with at least one P&G brand product: **98**
4. Number of categories where P&G has the #1 or #2 brand: **32**
5. Place of Procter & Gamble in the series of companies that have employed AOL chairman Steve Case: **#1**
6. Year by which AOL's estimated advertising revenues will pass those of ABC or CBS: **2003**
7. AOL advertising revenues in the year ending June 1999: **$1 billion US**
8. Number of web pages that contain the word "brand": **1,699,630**
9. Number of web pages that contain the word "branding": **92,288**
10. Estimated year when Egyptians first branded cattle: **2000 B.C.**
11. Range among estimates of 1999 Internet advertising revenues by 14 research resources: **$839 million-$5 billion US**
12. Estimated amount spent on advertising by Linux companies in 1999: **$15 million US**
13. Date on the fourth day of the current year, according to the I-Advertising and Seinfeld web sites: **January 4, 19100**
14. Same date on the Oldham Chronicle site: **Tuesday, January 04, 100**
15. Same date on the Gigabyte site: **January 4, 2100**
16. Same date on the Case Western Reserve site: **Tue. Jan 04 1900 EST**
17. Number of J Builder for Linux downloads by December 1999: **~100,000**
18. Number of J Builder for Windows downloads by December 1999: **50,000**
19. Building number on the Microsoft campus where Bill Gates works: **8**
20. Number of theses nailed to the Wittenberg Church by Martin Luther in 1517: **95**
21. Number of theses to begin the Cluetrain Manifesto (1999): **95**
22. Year Marx and Engels published the Communist Manifesto: **1848**
23. Months after publication Marx became editor for an industrialist-funded newspaper: **2**
24. Months before publication of the Cluetrain Manifesto Doc Searls became Senior Editor of *LJ*: **7**
25. Number of women named to *Linux Magazine's* Who's Who in Linux: **0**
26. Amount, in stock, RHAT paid to "merge" with HKS, Inc.:**$97,000,000 US**

27. Amount the world has spent since 1977 on licensed *Star Wars* merchandise: **$4,500,000,000 US**
28. Expenditures which the Pentagon could not account for last year: **$22,000,000,000 US**
29. Fee charged by a Pennsylvania cyber-psychologist for on-line treatment of Internet addiction, per minute: **$1.50 US**

Sources

- 1-18: Sloan Brands, Procter & Gamble, Fast Search & Transfer ASA, Inprise, *Fortune*, I-Advertising, *Linux Journal*, *The Register*
- 23, 28, 29, 30: *Harper's*
- 19-22, 24: Jason Schumaker
- 25: http://www.linuxgrrls.org/
- 27: *LinuxToday*

THEY SAID IT…

We think IT managers would be irresponsible not to take a hard look at Linux and consider it as a platform for new applications. The year 2000 will be marked by the rise of Linux and the release of Windows 2000, creating real choice among operating systems. There is a risk, however, that Wall Street greed-mongers could ruin it all by overheating expectations and seeking to turn a leading Linux player, such as Red Hat, into the next software monopoly.

—Editorial in *PC Week*

Amazon has obtained a U.S. patent (5,960,411) on an important and obvious idea for e-commerce: the idea that your command in a web browser to buy a certain item can carry along information about your identity...Amazon has sued to block the use of this simple idea, showing that they truly intend to monopolize it. This is an attack against the World Wide Web and against e-commerce in general.

—Richard Stallman in *Linux Today*

Source code is like manure, if you spread it around things grow. If you hoard it, it just smells bad.

—Zachary Kessin in *Slashdot*

Gates' Law: Every 18 months, the speed of software halves.

—Omer Shenker in *Slashdot*

Communication has changed so rapidly in the last 20 years. E-mail, which now sends data hurtling across vast distances at the speed of light, has replaced primitive forms of communication such as smoke signals, which sent data hurtling across vast distances at the speed of light.

—Steve Martin in *The New York Times*

olestra, olean, olestra, olean, Pringles, Wow chips, Max chips, Frito Lay, Procter & Gamble, anal leakage, diarrhea, gastrointestial problems, carotenoids, cramps, cancer, macular degeneration, Stampfer, Willet, vomit, Michael Jacobson, Jacobson, CSPI, Center for Science in the Public Interest, Nutrition Action Healthletter, food police, olestra, olean, olestra, olean, Pringles, Wow chips, Max chips, Frito Lay, Procter & Gamble, anal leakage, diarrhea, gastorintestial problems, carotenoids, cramps, cancer, macular degeneration, vomit, Michael Jacobson, Jacobson, CSPI, Center for Science in the Public Interest, Nutrition Action Healthletter, food police,olestra, olean, olestra, olean, Pringles, Wow chips, Max chips, Frito Lay, Procter & Gamble, anal leakage, diarrhea, gastorintestial problems, carotenoids, cramps, cancer, macular degeneration, vomit, Michael Jacobson, Walter Willett, Meir Stampfer Mark Donowitz, Mark Hegsted, Ian Greaves, Herbert Needleman, Fernando Treviño, John D. Potter, Johanna Lampe, Jerianne Heimendinger, Cancer Research Center, Norman Krinsky, Ernst J. Schaefer, John S. Bertram, Sheldon Margen, Jacobson, CSPI, Center for Science in the Public Interest, Nutrition Action Healthletter, food police

—HTML metatag for “A Brief History of Olestra”

APPLIX + COSOURCE FACE THE OPEN SOURCE MARKET

Applix has been in business for a generation. Its office suite, Applixware, has been a market leader since the '80s. By extreme contrast, Cosource has been in business for less than a year. But it has (along with its competitor SourceXchange) quickly helped establish a whole new market category: the development marketplace where buyers and sellers of open-source code can meet and do business.

Now Applix and Cosource are one: Applix acquired Cosource in December. *Linux Journal*, which has featured Cosource development news on its web page for much of the past year, was curious about the synergies that brought these two companies together. Doc Searls interviewed Cosource founder (and now Applix executive) Bernie Thompson on January 3, 2000.

Doc: What did Applix see in Cosource? The FAQ seems to say that Cosource is mostly a way to drive Applix development, rather than something for everyone —a marketplace. That's a big conceptual shift.

Bernie: In doing the acquisition, I made sure we would still be able to have say over Cosource and keep it unbiased. So Cosource is and will remain an independent broker of open-source work. Any other path would be self-destructive.

We've been acquired by a great company. Applix—the name means "Applications for UNIX"—is a software company that's been around since 1983. It would like to extend this noble history for another seventeen years. Like most software companies, it has derived much of its revenue from licenses. It saw that, as open source gained market share, the revenue balance would tend to shift away from licenses towards services. Cosource is a web-auction system that encourages open-source work and can thrive in this new environment. By having Cosource in the portfolio, Applix can have a more balanced mix of products and services and be a more healthy company as a result.

For things like seeding the development of open-source applications on our application builder (called SHELF, which we've released under the LGPL), we'll do that right on Cosource.com as a sponsor just like everyone else. If Applix wants to do anything related to their products that requires special logic (which is quite possible—like getting pre-commitments to buy copies of our commercial products), we'll do that by using the Cosource.com logic on another site like Applixware.com, which is specific to our company's products.

Doc: What will happen to Cosource? All kinds of questions come to mind here. In what new directions will you take it? Will you still run it? If not, who?

Bernie: Our goal for this year is to take the cooperative marketplace concept that we—Cosource—helped pioneer to the next level. We've done an okay job so far, but there are a lot of things we want to do better. If you look at our front page, you'll see that we've got a lot of good information there about what feature enhancements are in demand. But we need to do a better job providing customized views of that data. More importantly, we need to enhance our interface to make it easier to express interest in projects and commit money to specific proposals.

From the business perspective, we need to bring more partners into the system, and have them share in the risk and the reward. A site like eBay was able to launch as a largely closed-loop site that doesn't allow for affiliates or competition. It's an island. We doubt that will work anymore. We're looking to do a network approach, where partners like VA Linux, Red Hat and independent Linux consultants can participate side by side and benefit by driving the development of open source.

As far as who's running the show, Cosource.com is still my baby. The great team that built it is still intact. When you create something from scratch, there's always a strong emotional attachment to it. So while I'm responsible for a bunch of other products now, including the Applixware Office Bundle and Applix Anyware (an 800KB Java client to access the office suite running on a server), I'll still be looking over Cosource with the same goals and the same philosophy as before.

Doc: What exactly is your new job with Applix? What are your goals there?

Bernie: My new role is as the President of Applix's Linux Division. Whew! It's quite a challenge, but also a chance to do so much good.

Doc: What kind of changes should we be looking for from Applix over the next year?

Bernie: We're going to be focused on producing great Linux and *BSD applications. Our strength is that our applications were developed on UNIX many years ago. Our focus is on fast, tight, native applications. Toward that end, we're launching our Applixware 5.0 product this spring and summer that uses and integrates with GTK, for closer integration with the Linux desktop. We hope to make it the “most native” of all the office suites, and thus the most comfortable and hassle-free to use. Beyond this one release, it's our intention to keep up and increase the pace of innovation, since that's our primary value to customers.

Doc: Do you plan to open source any of Applix' products?

Bernie: We've already open sourced our application builder platform called SHELF (see http://www.applixware.org/). We've built a complete PalmPilot desktop interface using that tool and released it as open source. And as part of our Applixware 5.0 product, we have developed a graphical interface to the Pine e-mail program which will be released as open source. In each of these cases, we try to use public, well-known licenses such as the LGPL and GPL, rather than custom “Applix” licenses. We're going to be open sourcing more software in the future, as it makes sense. See Eric Raymond's “Magic Cauldron” paper for some guidelines. In particular, we're going to be using more and more open-source infrastructure in our applications—to the extent the licenses allow—and passing our enhancements to these infrastructures back to the community. This is the natural process of community enhancement that licenses like the GPL/LGPL pioneered, and it works great for us.

In short, the goal here is to prove that software product companies can still exist and thrive in this new market that the Internet and open source have

created. They'll look different, but we still need them. We need companies that can pay programmers for their hard work and not lose money every quarter. We need companies that can invest large sums in R&D, with the ability to earn back that risky investment by winning customers who license the product. Once that investment has been recouped, we can eventually shift over to an open-source service model.

These companies will be different in that they will be more open with information—not trying to lock customers in—and more focused on empowering the community around the product. Hopefully, this happy balance can be found between the "give it all away" and the "keep the customer in the dark" camps that divide the open-source and traditional software communities today. That's what we're aiming for: we hope to find that middle ground.

We know as well as anybody that the Open Source movement has done only good things for customers. It has demonstrated the amazing power of the community to do great things by working together. It's up to companies to absorb these lessons and learn how to apply them back to serving our own customers better. This is our challenge and mission in the coming year.

—Doc Searls

VENDOR NEWS

Adobe Systems announced its initial support for Linux. In the first quarter of 2000, Adobe is offering a Linux version of Adobe Acrobat Distiller software. A beta version of Adobe FrameMaker software for Linux can be downloaded from the Adobe.com web site.

The **Jan III Sobieski Hotel** (Poland) announced that its official operating system is Linux and its office suite is StarOffice. The installed software includes various releases from Linux, Red Hat, SuSE, Mandrake, LX router, StarOffice, HS Partner and others.

Sangoma announced in December that it is involved in a reverse takeover of Inlet Devices Corporation, a public company registered on the Canadian Venture Exchange (CSV). Details can be found in the entries for Inlet on the SEDAR web site at http://www.sedar.com/.

Bitstream Inc. and **Corel Corporation** announced that Corel has licensed a Linux font server currently being developed by Bitstream. Corel and **S3 Incorporated**'s Professional Graphics Division announced a partnership to deliver 2-D/3-D graphics to the Linux desktop. Corel and **Creative Technology Ltd.** announced an agreement that will advance the development of Linux applications for high-quality audio and video. Corel also announced it has

acquired an ownership stake in **LinuxForce Inc.** of Philadelphia. LinuxForce delivers a full range of technical services and support for Linux, allowing Corel to deliver end-to-end Linux solutions.

Eicon Technology, a worldwide provider of remote access products, announced the release of Linux drivers for its PCI bus ISDN server adapters, the DIVA Server BRI-2M and the DIVA Server PRI-23M. The new drivers will work with Caldera's OpenLinux 2.3, Red Hat Linux 6.0 and SuSE Linux 6.2.

Development has begun on a free open-source **Linux Firewalling System**. Home page is located at http://www.sinusfirewall.org/. This firewall works with kernels 2.2.*x* and supports NAT. It has a configuration and management graphical interface written in Java.

LinuxBusiness.com aims to build a huge repository of different ways to use Linux in corporate environments. If you use Linux in your corporation or belong to a service company that has deployed Linux solutions for customers, please post your detailed comments at www.linuxbusiness.com/en/bizform.html.

Hewlett-Packard has stepped up efforts to make Linux compatible with its most powerful processors. HP has retained a Linux consultancy, the Puffin Group, to ensure Linux runs on its advanced, 64-bit PA-RISC chips in the first half of this year. A 32-bit version of Linux for the PA-RISC chip architecture is ready now.

StarBurst Software announced that its product OmniCast, content distribution management software, now supports the Linux operating system. StarBurst's OmniCast is like a multicast mimicker, sending content over satellite, terrestrial and the Internet without any changes to the customer's network.

Digital Media Online, a developer of web-based vertical communities for the digital media market, announced the launch of the first Internet community for professional content creators working on Linux-based systems at http://www.CreativeLinux.com/.

A training partnership has been established between **Linuxcare** and **Wave Technologies**. Wave Technologies will now be delivering Authorized Linuxcare training courses to prepare participants for certification.

O'Reilly & Associates is leading a discussion of open source's impact on publishing. O'Reilly editor Andy Oram is hosting a web conference that examines how the Open Source community and professional publishers can use the principles and practices of open-source development to create technical documentation. Join the conference at http://forums.oreilly.com/~publishing/.

QLogic Corp., a provider of Fibre Channel host bus adapters and SCSI connectivity solutions, announced full, optimized support for the Linux operating system on its QLA2100 and QLA2200 series Fibre Channel host bus adapters as well as its Ultra3 and Ultra2 SCSI host bus adapters.

TSCentral has set up an on-line directory of Linux event- and training-related resources. You can find out more by visiting http://www.linux.tscentral.com/.

SuSE announced a beta version of SuSE Linux 6.3 for the Macintosh PowerPC at MacWorld in San Francisco, with sales release planned for spring. Based on the current version of SuSE Linux, this version includes all open-source software found in the Intel version and is identical in use and administration to other SuSE Linux versions. A free test CD can be obtained by contacting SuSE, or software can be downloaded from ftp://ftp.suse.com/.

Inprise Corporation announced it is open sourcing InterBase 6, the new version of its SQL database. InterBase will be released in open-source form for multiple platforms, including Linux, Windows NT and Solaris during the first part of this year.

Simon Phipps, IBM Corporation's chief Java and XML evangelist, will kick off the **O'Reilly Java Conference** on March 27-30, 2000 in Santa Clara, CA. The O'Reilly Java Conference is an intensely technical four-day conference for Java programmers. See http://conferences.oreilly.com/java/speakers/.

IDG World Expo announced that The XFree86 Project, Inc. is the recipient of the February 2000 IDG/Linus Torvalds Community Award. IDG Chairman and Founder Patrick J. McGovern and Linux creator Linus Torvalds will present the $25,000 award at LinuxWorld Conference & Expo, following Torvalds' keynote address on Wednesday, February 2, 2000 at the Jacob Javits Center, New York, NY.

TIMBERRR!

Trees do not grow to the sky, and right now, the Redmond Redwood is about as close as it's going to get. Worse, it may be getting ready for a big fall. At least that's what Eric S. Raymond thinks, although as a gun enthusiast, he tends to favor firearm metaphors. Recently, the alpha hacker and libertarian economist shared some of his latest thinking in a conversation with *Linux Journal* Senior Editor Doc Searls.

Doc Searls: I heard you've been saying some new stuff about Microsoft. What's up?

Eric: There is a new section of my talk that I'm doing these days. It's called “The Seven Bullets Microsoft has to Dodge to Survive the Next Eighteen Months”. I haven't written it down yet. The bottom line is that Microsoft has much bigger problems than either the Department of Justice lawsuit or Linux. It's just that few people have noticed these problems yet.

Doc: To keep this short, what's the biggest bullet?

Eric: The margin crunch problem. Here's how it works: Microsoft's stock price has to rise every quarter. If that doesn't happen, two very bad things occur. First, the employee stock options stop rising in value. When that happens, their talent bails out. All those people stop working eighteen-hour days in Redmond, and go off to build mansions.

Doc: Which is already happening?

Eric: Yes. The other problem is that Microsoft makes more money playing option games with its own stock than it does selling software. Thirty six percent of its income, and that income goes away if the stock price doesn't trend reliably upwards. If stock prices must always go up, so must revenues, quarter to quarter. This is a problem. Every quarter, it becomes harder and harder to find the additional revenues.

Doc: Why?

Eric: They've got 91% of their market. There isn't enough room for them to get the revenue they're accustomed to.

Doc: But the overall market has tended to increase.

Eric: Not fast enough. And we know this without econometric modeling. Microsoft is raising prices on its high-end customers. In the long term, this is suicide because it will only drive customers to competing operating systems. It really only makes sense if they're caught in the short-term scramble for revenues, and absolutely must have the money.

Doc: I've heard from corporate guys in big companies that the real aversion to Microsoft has less to do with software than licensing fees: expensive licensing of NT servers that customers would rather avoid.

Eric: Today, if you buy a Windows NT server, they not only charge you per seat for the number of developers on your site, but they also charge you per seat for the number of simultaneous web accesses you want to support. You see where that's going. So we know market expansion won't work for them. Now, here is where it really starts to bite: the price of hardware is dropping like a

rock. Microsoft makes money from the hardware vendors—the OEMs—they hold captive. These guys are caught in a bind. From one side, Microsoft has to take a bigger piece of their margin every quarter just to stay afloat. From the other side, the price of hardware is falling. If your total system price is $2500 US, it makes sense to pay the $80-$100 Microsoft tax. When your total system price is down around $600, the margin pressure becomes unsustainable. This implies that there is a price level in the PC and appliance market below which you can't make any money dealing with Microsoft. The key point is that this price level is not fixed over time.

Doc: Where does it stand now?

Eric: In appliance territory. Which is why you see companies like Nokia and GTE Sylvania defecting from the Windows CE alliance. They've figured out they can't make any money at that price point, given the license price of Windows CE. Over time, because the price of hardware is dropping, the functional point represented by that price point is going to rise into low-end consumer PC territory. When the price point at which you can't make any money dealing with Microsoft passes the average price point of a consumer desktop PC, the game is over. And I think this is going to happen before the justice department gets its final verdict.

Doc: Will Zachmann also predicted this quite a long time ago. He thought the margin squeeze would hurt their stock and then they would be abandoned by all those optioneers who have been working the long hours, waiting for a stock payout.

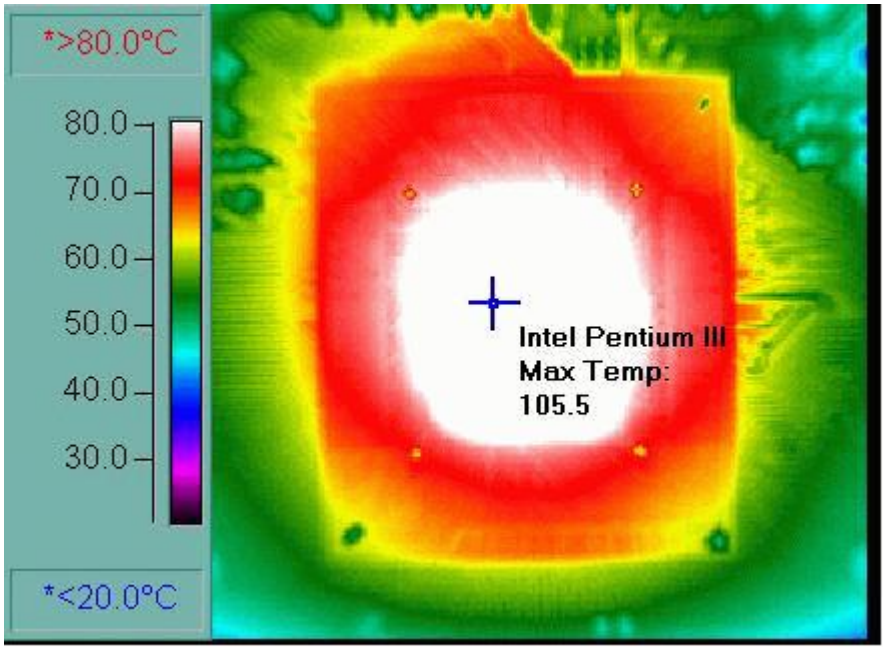
Eric: It's coming. Microsoft simply can't maintain their existing margins. Stepping back and looking at economic history, this is what always happens to monopolies unless the government props them up. They get comfortable at a certain price level, and get that level built into their whole financial structure, then collapse when they price themselves out of their own markets.

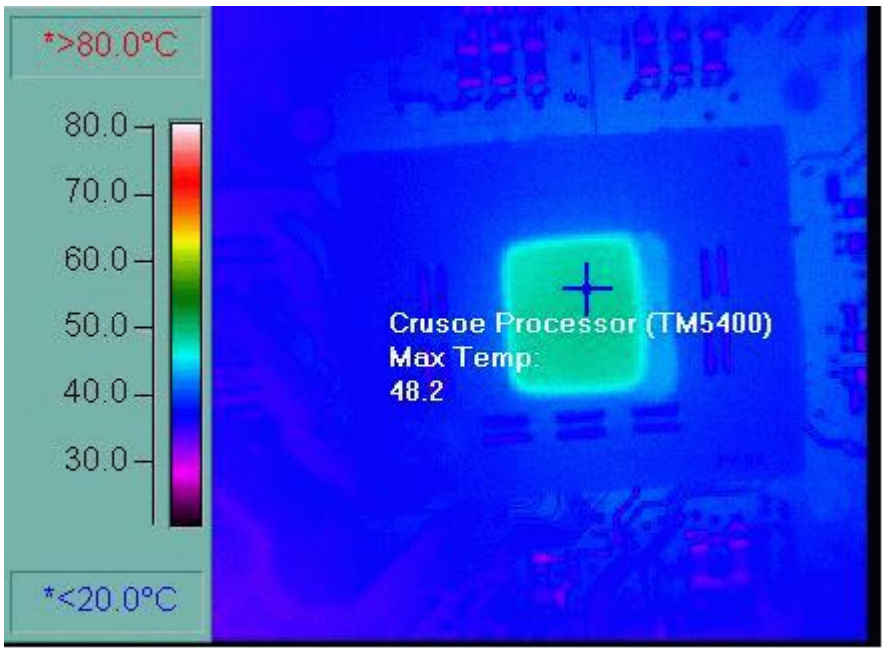
—Doc Searls

STOP THE PRESSES: Cool Chips for a Hot Category

After four and a half years, Transmeta has lifted its veil of secrecy and spilled the beans—or rather the chips. There are two, both built to run x86 code at high clock speeds while delivering unprecedented battery mileage. The larger of the two (TM5400) is targeted to the notebook and sub-notebook Windows laptop market. The smaller one (TM3120) is targeted to the new Mobile Internet Appliance market.

According to Transmeta, that market wants three things: real web browsing, real portability and long battery life. The market, they say, demands Linux, and they are more than equipped to give it to them. “We have a certain amount of expertise with the Linux OS,” says Transmeta co-founder and Chairman David Ditzel. Transmeta is famous as the employer of Linus Torvalds. Mobile Linux is a native x86 version small enough to fit into ROM with enough room left for a browser and other software components. Mobile Linux runs on the smaller and cooler of the two chips, and the system is upgradable.





The figure demonstrates that an Intel and a Transmeta chip are as different as a stove and an ice cube—between 105.5 and 48.2 degrees Celsius. LongRun technology on the TMS400 even regulates clock frequency and voltage to

correspond exactly to the demands of the application, drastically conserving energy and further reducing heat load.

Ditzel, one of the originators of the RISC (reduced instruction set computer) concept and the leader for many years of Sun's SPARC effort, co-founded Transmeta to tackle the challenges of mobile computing. What Transmeta finally delivered not only meets that challenge, but achieves some RISC ideals as well.

In effect, Transmeta designed a whole new breed of microprocessor—one idealized for mobile computing. The Transmeta architecture is utterly unlike anything coming out of Intel or its standard competitors. At its core is a powerful VLIW (very long instruction word) engine, surrounded by a “code morphing” software layer that translates x86 instructions and intelligently caches them while observing the needs of the system's applications, so the chip optimizes execution of translated instructions at extremely high speeds. By replacing millions of transistors with software, the Transmeta chip is small, fast, efficient and extremely undemanding in the way it consumes electricity.

The result is a breakthrough into territory that has remained closed in the absence of designs featuring the kind of long battery life that customers now expect of portable radios, telephones and other popular hand-held devices—the Mobile Internet Computing market.

The big question is: will the market buy it? Linley Gwennap, a microprocessor analyst with The Linley Group, thinks chances are good:

> The fundamental problem is the simple fact that microprocessors are consuming more and more power every generation. And Dave Ditzel is right: we're headed for a hundred watt chip on the desktop. So this isn't a problem in the mobile area. This kind of technology, which gets away from the complexities of doing everything in the chip, and shifting more of it into software, makes a lot of sense. It solves a lot of big problems.

Note: More Transmeta articles can be found on our web site at http://www.linuxjournal.com/articles/misc/013.html and ~/articles/business/030.html and an interview with Linus at ~/articles/conversations/012.html.

THE BUZZ

During the month of December (and the start of January), people were talking about:

- SuSE's debut of the beta version of its 6.3 Linux distribution for the Macintosh at MacWorld Expo in January. The finished release is expected to ship this spring and will be identical to the SuSE we know and love.
- Apple's release of its newest operating system, Mac OS X, later this summer. The kernel, code name Darwin, is said to be “Linux-like, featuring the same FreeBSD UNIX support and open-source model”. We shall see! (from *PRNewsWire*, January 5, 2000)
- Macromedia's announcement that it will be releasing its Flash Player Source Code SDK & Flash File Format (SWF) SDK in mid-January. (*Linux Today*, January 6, 2000)
- Gillian Bonner's (Playboy's Miss April 1996) posting a positively glowing review of Linux (Red Hat and Corel) in which she predicts “that very soon the Linux OS will dramatically change the operating system...and thus the way we work and play on our computers.” (*Linux Today*, January 5, 2000)
- Intel Corporation's announcement that it is developing a family of Intel-branded, Linux-based web appliances expected to debut later this year. The first category of Intel appliances will combine phone and web services, which will not run on Windows!
- The Chinese government's potential ban on governmental use of Windows 2000, as they move toward open-source technology with development of Red Flag Linux! (*The Register*, January 6, 2000)
- RHAT becoming quite chummy with Salon.com, who will provide “award-winning journalism” for Red Hat's Wide Open News web site. Hmmm...didn't Microsoft do something similar with *Slate*? (*Linux Today*, January 06, 2000) —Jason Schumaker

Rumor: Top geeks at Red Hat, Inc. are seriously considering dropping the well recognized brand name in favor of its NASDAQ ticker symbol (RHAT). The change is expected to help push RHAT's new product line: stocks!

Factoid: VA Linux millionaire Larry Augustin tabbed Office, not Windows, as Microsoft's real killer asset. (*Linux Today*, January, 5,2000)

Factoid: The revolution begins another year. With it comes a new title for the Linux faithful: Penguinistas. Xavier Basora coined the term and to that we say, Viva la Penguinistas!

Quote: Every morning when I wake up, I try to remember who I am and where I come from. —Harry S. Truman (The Cluetrain Manifesto)

Factoid: An oil slick near Phillip Island has endangered nearby penguins. Any support would be appreciated (http://www.penguins.org.au/). For off-line donations please send to Phillip Island Nature Park, att: Penguin Hospital Support Fund, P.O. Box 97 Cowes, Phillip Island, Victoria, 3922, AUSTRALIA.

Archive Index Issue Table of Contents

Advanced search

April 2000

Phil Hughes

Issue #71, March 2000

Ask around and see how many geeks you know who bought Andover, Cobalt, Corel, Red Hat or VA Linux stock.

As a geek who became a magazine publisher, it feels strange to write what sounds like a Business and Finance column. I haven't changed, but the Linux community certainly has. If you don't believe this, ask around and see how many geeks you know who bought Andover, Cobalt, Corel, Red Hat or VA Linux stock.

This year has already started out with a bang for the Linux community. First, there is a rumor that the reason Bill Gates stepped down as CEO of Microsoft to become its Chief Software Architect is Linux-related. Okay, maybe “rumor” is too strong a word, but the number-two reason according to *Salon Magazine* is that Bill needed more time to learn Linux.

While I don't believe this is 100% true, things *have* changed for Microsoft. Between the Department of Justice and Linux on the OS level, Microsoft has been forced into a competitive arena. In the applications arena, Sun's purchase of StarOffice and making it freely available offers a difficult-to-match price point. I expect Microsoft will still have a lot of customers, but it is nice to see them having to work harder to attract new users.

The influx of capital into the Linux market is amazing: about $100 million between Turbo Linux, SuSE and Caldera Systems. Also, we see interesting players: some of Caldera's money came from Sun Microsystems.

Turbo Linux's investors make up another interesting list: Dell, Compaq, NEC, SCO, Toshiba, et al. For anyone who has tried to buy a Toshiba computer without MS Windows, this is fairly surprising. And just a year or two ago, SCO was telling us we could trade in our copy of Linux to upgrade to SCO UNIX.

Another interesting company is TiVo, Inc., profiled in last month's *LJ*. The quick summary is they have created a Personal TV Receiver, a Linux-based system that acts like the world's smartest VCR, recording up to 30 hours of TV programs on a hard disk. Why is TiVo hot? First, Phillips is making the systems, so we have a major manufacturer behind this “Linux and your TV” effort. Second, TiVo secured a pact with Blockbuster to do video-on-demand.

There's more. Macromedia will likely open source their flash player. Transmeta (see “Stop the Presses”) has announced their Crusoe chip. Huge investments have been made in Linuxcare. The Caldera IPO is happening.

Linux has grown up. It is finally being recognized as something that solves problems, not just for computer geeks, but also for the mainstream. In the eyes of businesspeople, Linux is finally more than a web server.

My prediction is this is just the beginning. More companies like TiVo will adopt embedded Linux in the mass market. While Internet Appliances (IAs) will be based on other platforms, a majority of the IAs will likely run Linux—it has the functionality and the right price.

The acceptance of Linux in the embedded market adds to Linux's success in the server market. When people access Linux-based web servers from their Linux-based IA, they are getting a strong message that Linux, not Microsoft, is bringing them technology.

I hate to just conclude we won, but it seems like we are there. Linux is marketing itself these days. Geeks believe in it, embedded-systems developers believe in it, and the general public has gone from not knowing it exists to seeing it as a viable answer for their needs. I'm not sure if my next toaster or car will run Linux, but I wouldn't be surprised if it happened.

Oh, and Bill: if that point in Salon Magazine was right, please stop by. I will be happy to give you a copy of our new Linux Command Summary.

email: info@linuxjournal.com

Archive Index Issue Table of Contents

Advanced search

LINUX JOURNAL

Teaching Linux

Peter H. Salus

Issue #71, March 2000

A very large number of potential users see SGI as an Irix company. The result is SGI's Linux University, a traveling roadshow that made many stops this past autumn.

Education means several disparate things. Education re Linux could mean using Linux in the classroom to teach subject matter unrelated to computing or operating systems. It might mean using Linux to teach Linux. It might mean teaching about Linux.

For over a decade, Silicon Graphics has built the highest-quality graphics machines, running Irix (a BSD derivative) on MIPS chips. A few years ago, they began a line of Intel architecture boxes. Now you can get one of those boxes running Linux. (I know there are ports of Linux to the MIPS architecture, too; but SGI isn't supporting them at this time.)

A very large number of potential users see SGI as an Irix company. The result is SGI's Linux University, a traveling roadshow that made many stops this past autumn. Beginning in Washington, DC and ending up weeks later in Los Alamos, NM, Linux University brought information on Linux, about SGI's policy concerning Linux and on SGI's commitment to open source to several thousand users.

While I found much of the content of the talks truly interesting and valuable, the most important revelation may well have been the fact that SGI has contributed its proprietary XFS journaling file system to the community under the GPL. Giving away one of the crown jewels is a true sign of commitment. As Kurt Akeley, CTO at SGI, put it: “In a bacon and eggs breakfast, the chicken is involved, but the pig is committed.” SGI is committed.

What the university tour did was two things: provide a context for SGI's commitment, and supply education about Linux itself—the values, tools and

future. At five of the sites (Huntsville, AL; Dayton, OH; Denver; Albuquerque; and Los Alamos), I supplied context by talking about the origins of UNIX, Linux and Open Source.

The tutorial sessions were very impressive. While they varied from site to site, they comprised XFS and CXFS, Networking and Clustering on Linux, Interoperability in a Heterogeneous Environment (Samba), Security in Linux, Web Serving in a Linux Environment and OpenGL in Linux.

Also impressive was the number of attendees. Most sites had over 200 folks present, with over 900 in Washington, DC. Many of the attendees were knowledgeable about computing, and simply not Linux users. The queries and comments I heard during breaks were at an extremely high level. They might be Linux newbies, but not Irix or BSD newbies.

Perhaps yet more significant, in Huntsville (NASA), Dayton (Wright-Paterson AFB) and Los Alamos (LANL), a large proportion of the attendees were from the U.S. federal government. Linux infiltrating the shops of the feds is a terrific toehold.

Don't forget that Linux was first put on the Net in October 1991. As recently as January 1996, USENIX was offering a single Linux tutorial. The next year, there was a USELINUX track, concerning Linux Application Development and Deployment. Since then, there have been Freenix tracks in both 1998 and 1999. This is important, for it shows exactly why something like SGI's Linux University can have great influence. The first Linux Expo was held in North Carolina in 1996; the first Atlanta Linux Showcase in 1997; and the first Linux Business Expo at Comdex in 1999.

Readers of this magazine may know Linux. But there are many, many programmers and system administrators who know that Linux exists, but have little or no notion of what it can do.

We could sneer at companies like SGI and say they just want to latch onto the business that supporting Linux will bring them. I'm convinced that's not the full story. I've now seen Linux applications running on several different pieces of SGI hardware: a 1400L server, a 540NT workstation, an Irix02 workstation. They all demonstrated to me that high-quality, robust software was running on first-class hardware.

If the Linux University is coming to your area, register for it. I think SGI is doing a lot for the Linux community, and it's teaching thousands of people about the Linux system. Knowledge is power.

Peter H. Salus , the author of *A Quarter Century of UNIX* and *Casting the Net*, is an *LJ* contributing editor. He can be reached at peter@usenix.org.

Archive Index Issue Table of Contents

Advanced search

Canon Fodder

Doc Searls

Issue #71, March 2000

“Of all the practical implements common to the Texas ranchman and cowboy, nothing equaled the branding iron in matters of ownership on the open range. The cattle brand was the key to ownership in a business where ownership was everything.” --Vincent C. D'Amico

What is it with Red Hat? Why, in the world's most open and choice-rich technology market, does Red Hat have Microsoft-grade market shares? If Linux distributions are bottled water, why does Red Hat move so much more digital H<->2<->O than Caldera, Debian, TurboLinux and the rest of them?

The answer, by acclaim, is that Red Hat has *branded* itself better than anyone else. Okay, it helped that Red Hat made installation of software easy with the Red Hat Package Manager, now used by pretty much everybody. And maybe it helped that Red Hat used Python as the scripting language for their system-administration tools. But the prevailing perception is that Red Hat is a branding success.

That success is extremely enviable. In fact, we are witnessing a record tide of brand envy right now. The whole dot-com world is gaga over this ancient and brutal marketing notion, and Red Hat is offered as a huge case in point. Thanks to the success of brands like Red Hat, branding is a huge marketing fad, fueled by apparently limitless billions poured through startups and into the media from swelling vats of venture capital.

This flood of dot-com advertising dollars is driving up the price of traditional media and driving out the traditional advertisers along with it. A billboard on Silicon Valley's Highway 101 now rents for upward of $100,000 per month: more than double the price from just one year ago. Gone are ugly ads for car dealers and office supply stores, replaced by artful ads for WhatEver.com, which is no doubt branding itself.

This begs the question: just what is *branding*, anyway?

Search for the subject at About.com, and you'll find links to nearly three hundred pages, all but one of which is devoted to body art (“branding & tattooing”). The remaining page is a set of marketing links, among which you won't find a word—or a link to a word—about what branding is, where it came from, or why it's so hot in the dot-com world.

The FAST search engine (http://www.alltheweb.com/) finds 1,699,630 pages with the word “brand” and 92,288 with the word “branding”. Oddly, few of those pages are about artful skin burns. Or maybe all of them are. Because both brand and branding come to marketing from the cattle business, by way of the first company to adopt the term “brand”, and the one that still understands it best: Procter & Gamble. From that company's history page:

> P&G's brand management system began to take shape in the late 1920s. In 1931, Neil McElroy, the Company's Promotion Department Manager, created a marketing organization based on competing brands managed by dedicated groups of people. The system provided more specialized marketing strategies for each brand and Procter & Gamble's brand management system was born.

This development coincided with the birth of commercial broadcasting, which P&G did much to fund and define. In 1923, P&G's Crisco was one of the first radio program sponsors. In 1933, P&G's Oxydol sponsored the radio serial “Ma Perkins”. By sponsoring other shows with other detergents, P&G helped create the label “soap opera”.

In 1948, McElroy took over as president of Procter & Gamble. By then, P&G was already what they still call themselves today: the world's top branding company. P&G now has over three hundred brands, dozens of which are #1 or #2 in their categories: Tide, Cheer, Vicks, Pringles, Crest, Pampers, Tampax, Ivory and Clearasil are all P&G brands. What makes them brands is the fact that their identities have been burned into our brains.

Brand management isn't for cowboys. It's a military job. Companies like P&G are in a constant battle for shelf space, often pitting the company's own brands against each other. Because the fundamental nature of that battle has not changed for generations, the principles of branding have long since been proven. Always show the package in the first eight seconds of a TV commercial. Always repeat the name at least five times. Always promise a benefit.

When an advertising campaign breaks the rules, the rules it breaks are the ones P&G still teaches—along with the secrecy, the protectiveness, the paranoia, the

company song (yes, like IBM, they have one) and all the other varieties from the canons of brand management.

While it's safe to say that most dot-com marketing folks don't come from this kind of mindset, it is also untrue. Marketing has always modeled itself on warfare. The verb “market” is a transitive verb. It acts on a direct object. Since markets tend to be competitive and we conceive competition as a kind of warfare, it is no surprise that marketing is thick with military thinking. In “Marketing Warfare”, Al Ries and Jack Trout write,

> The true nature of marketing today is not serving the customer; it is outwitting, out-fighting and out-flanking your competitors. In short, marketing is war where competition is the enemy and the customer is the ground to be won.

No wonder so much marketing treats customers like dirt.

Most of marketing and branding rhetoric today is of a kinder, gentler sort. But it is also kind of vague. “Identity, personality and reputation, while related, are not synonymous,” one correspondent wrote to me. “Branding encompasses parts of all of them.” Buzzwords are like that. They encompass, but they don't define.

If you can't define what you mean, at least use a good metaphor. That's something Red Hat's Bob Young has always done very well. In his book *Under the Radar*, he writes,

> Our goal had always been to become the Heinz ketchup of the Linux world—low-cost, high-quality, reliable and predictable.

Bob Frankel, whose Big Time Branding site (http://www.frankel-anderson.com/) ranks high on most searches for branding, has this definition of the subject:

> Branding is not about getting your prospect to choose you over your competition; it's about getting your prospect to see you as the only solution.

Clearly, Red Hat does that, and they do it in a way that works in the Open Source world, where there are plenty of other equally appetizing varieties of ketchup.

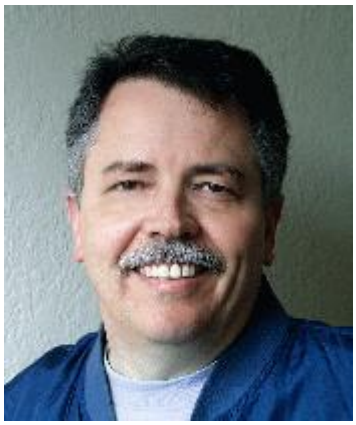
Again, why? I think it has to do with trust. My old partner David Hodskins once defined brand as “the capitalized value of the trust that exists between a company and its customers.” As I write this, the capitalized value of Red Hat is

$14,632,710,582, give or take the few hundred million Red Hat gains or loses on any given day. That's a lot of trust.

It so happens that natives of the Open Source world put a very high value on trust. You earn that trust (as you do anywhere) by being as open and honest as possible. I believe this is manifest in the high market valuations of highly trusted Linux companies. Agreed: the stock market is a very different place than the market for open-source code. But the stock market is also proving to be highly intuitive. It assigned high valuations to Netscape, Yahoo! and other Internet stocks long before the rest of the business world realized that the Net truly was going to goose up the whole economy. And now it is doing the same with Linux stocks.

What do they trust? It's more than just a few Linux brands. It's the explosive potential of what can't be owned, but can be shared.

So maybe Red Hat succeeded because it was on the radar all along.



In addition to serving as Senior Editor of *Linux Journal*, **Doc Searls** (info@linuxjournal.com) is co-author of *The Cluetrain Manifesto*, the web site (http://www.cluetrain.com/) that is now a book by the same name from Perseus Books.

Archive Index Issue Table of Contents

Advanced search

Best of Technical Support

Various

Issue #71, March 2000

Our experts answer your technical questions.

Failure to Connect

My PPP connection is failing while using a modem to dial up to my ISP. My log indicates (after **chat**):

```
pppd:Serial Connection established
pppd:Using interface ppp0
pppd: Connect: ppp0 /dev/ttyS2
pppd: LCP: timeout sending Config-Requests
pppd: Modem Hangup pppd: Connection terminated
pppd: Exit
```

My options file with and without these lines given separately had the same effect:

```
lock passive lcp-echo-failure 10
lcp-echo-interval 400
```

Chat (tried different values):

```
'' 'AT&F' (also used ATZ) 'OK' 'ATDTxxxxxx' 'CONNECT'
'' 'login:' 'xxxxxx' 'password:' 'xxxxxx' 'TIMEOUT' '60'
```

pppd options (also tried 1500 for MTU and MRU:

```
hardflowcontrol=yes defroute=yes MRU=556
```

Do I need to include something more in my chat to make the modem wait longer ? Also, according to my ISP administrator, my machine is “not replying to LCP request”. My modem is a BestData 56K internal modem (#56SF), controller-based. I am using Red Hat 6.0 Intel. —Vijay Nunna, vijay_nunna@hotmail.com

There isn't anything else you should include in your chat script; however, some servers may be slow to start up the LCP communication, and thus your **pppd** daemon will timeout before then. To raise the number of LCP configure-request packets before pppd determines it is not responding, you can add the

lcp-max-configure option to your /etc/ppp/options file, followed by the number of LCPs to send. For example, **lcp-max-configure 30** would raise the number from the default of 10 to 30. You might also want to remove the—Andy Bradford, andyb@calderasystems.com

You need to add **debug** on your pppd command line and look at the /var/log/messages file. You also want to replace chat with **chat -v**. Those two combined will give you lots of debug output and hopefully a better idea of what is failing and where. —Marc Merlin, merlin_bts@valinux.com

Dependency Question

I am trying to install a package that includes a Ghostscript driver for my printer. I'm getting dependency issues. The files required (as printed by **rpm**) are:

```
libc.so.6(GLIBC2.0) libc.so.6(GLIBC2.1) libm.so.6(GLIBC2.0)
libm.so.6(GLIBC2.1)
```

I've upgraded the libc libraries to glibc 2.1, but this seemed to have no effect. Looking at the dependency list, it also seems to need libc.so.6 and libm.so.6 without the GLIBC attached, and it has no problem finding these. I've looked at FTP sites for files with these names and cannot find them. Can anyone tell me what these are and if I need to find them, or whether I can tell rpm to ignore them? —Doug Morgan, dmorgan@cfa.harvard.edu

Often, a dependency on libc6 or libm6 means only that the binaries were compiled to use these versions of the libraries—they can often be recompiled against an older libc/libm without any problems. To do this, grab the SRPM (it should have the extension .src.rpm), do

```
rpm --rebuild
```

and install the resulting binary .rpm. Use this approach with caution, as there are applications that exhibit bugs under the older libc/libm (or won't compile). Still, in my experience, it generally works just fine. —Scott Maxwell, maxwell@ScottMaxwell.org

Although you can install glibc 2.1 on a glibc 2.0 system, it may be better for you to upgrade your system to SuSE 6.3 instead. However, if you do have all the libraries installed, but for some reason the RPM database doesn't agree, you can force the package to install with **rpm --nodeps *package*.i386.rpm**.

Another option is to get the .src.rpm version of your package and rebuild it for your system: **rpm --rebuild *package*.src.rpm**. The resulting package should be in /usr/src/redhat/RPMS/i386/. —Marc Merlin, merlin_bts@valinux.com

Dueling Processors

I am thinking of building a dual-processor computer and wondering if Linux supports this. I have heard Linux is the best OS, and I have heard Windows NT is the only OS that supports multi-processing. —William Cason, fordsho@bellsouth.net

Linux does indeed support SMP. In fact, I use a fast dual-CPU Linux box at work. —Scott Maxwell, maxwell@ScottMaxwell.org

Linux may or may not be “the best”, depending on what you're trying to do and what you're looking for. It does support multi-CPUs on recent kernels (2.2.x), and some distributions like Red Hat support SMP at install time. Some require you to compile your own kernel. —Marc Merlin, merlin_bts@valinux.com

Radio Days

I run Linux Red Hat 6.0 and Netscape 4.7, and I am struggling to install Real Player. I successfully installed (using RPM) Real Player 5.0 (**rvplayer**) and verified using GNOME; RPM shows the program is installed correctly. In the Netscape options, I set up MIME for file RA and RAM to use rvplayer. When I click on an Internet site with radio, nothing happens. There is a message on a Linux newsgroup, saying there seems to be a problem with rvplayer 5.0 and the 2.2. —Yossi HaYored, hayored@mindspring.com

That is correct; you need the beta version of the G2 player. You can get it from proforma.real.com/mario/player/player.html. —Marc Merlin, merlin_bts@valinux.com

I had trouble getting rvplayer 5.0 to work on a 2.2.5 kernel myself, but rvplayer 6.0 worked just fine for me. —Scott Maxwell, maxwell@ScottMaxwell.org

Colorful Monitors Ensue

I have been trying to install Linux and have had one successful installation using a monitor other than my personal one. When I took it home to my monitor, it gave me a blank screen, and the monitor's on light goes from green to maroon, indicating a problem after it entered its graphical interface. When I try to reinstall Linux Mandrake or Caldera 2.2, it gives me a blank screen when it does its graphical interface or when I try to probe for a video card. I have a Compaq V50 monitor and Creative Labs 4MB video card. When I was able to get a successful installation, I was using a Micron monitor. Could there be a compatibility issue, and will there be a way to get Linux installed at all? This monitor configuration works with Windows just fine. —Andy Kissner, thekiss@netzero.net

What's apparently happening is that X is trying to drive your monitor too hard. For this reason, graphical installs are not necessarily going to work. At least with Red Hat, you can ask for a text-mode install when you boot the first floppy. Some distributions, like Debian, will install only in text mode, which is fine for you, and others expect to see a VESA-2-compliant video card and a monitor that can accept suitable refresh rates. Your best bet, once a distribution is installed, is to edit XF86Config (usually in /etc/X11 or /etc) and reduce the value of those two parameters:

```
HorizSync 30-50
VertRefresh 50-70
```

The values given here may work, but you should really put in the ones specified in your monitor's manual. —Marc Merlin, merlin_bts@valinux.com

Zippity-doo-dah

I actually have several questions.

1. How do I access my Zip drive? I suppose I could try mounting and unmounting everything in /dev, but that seems particularly ungraceful.

2. How do I log out? I can switch to superuser and shut down, but letting my kids and wife be superuser defeats at least one reason for preferring Linux over Windows.

3. I tried to set my default shell to tcsh instead of bash. It doesn't seem to have taken. In particular, my .cshrc file (or .tcshrc) is not read at login. —Tim Allison, tallison@logicon.com

1. The answer is very different, depending on whether your Zip is parallel, IDE or SCSI. There are two HOWTOs which you should read: ZIP-Drive and ZIP-Install.

2. In a shell, type **logout**. In X, you have to find the logout option. It depends on which window manager you are using and whether you use XDM (runlevel 5). When in doubt, **CTRL-ALT-BACKSPACE** will kill your X session, but that's the sledgehammer approach.

3. To change your shell, you can use **chsh** or simply edit /etc/passwd and change the last field for root. Make sure that whatever shell you specify exists. To find out which shell you are currently running, try **echo $SHELL**. If you are indeed running tcsh, try doing an **echo** in ~/.tcshrc to see if it is run or not, and consult the tcsh man page which explains which files are being run, depending

on whether your shell is a login shell or not. —Marc Merlin, merlin_bts@valinux.com

Resources

New Products

Ellen M. Dahl

Issue #71, March 2000

AVP for Linux/FreeBSD UNIX, API PowerRAC Chassis 320, ODBC-ODBC Bridge and more.

AVP for Linux/FreeBSD UNIX

Kaspersky Lab announced the public release of the first commercial version of AVP for Linux/FreeBSD UNIX, a comprehensive anti-virus defense system for workstations, file servers and application servers. It includes three main components: AVP anti-virus scanner, anti-virus filter AVP Daemon and background virus interceptor AVP Monitor. AVP Monitor is a client part of AVP Daemon that intercepts all file operations and checks objects for viruses.

Contact: Kaspersky Lab Ltd., 10, Geroyev Panfilovtcev Str., 123363 Moscow, Russia, +7-095-948-56-50, +7-095-948-43-31 (fax), info@avp.ru, http://www.avp.ru/.

API PowerRAC Chassis 320

Alpha Processor, Inc. announced the API PowerRAC Chassis 320, a robust, high-density packaging option for its high-performance UP2000 motherboard. It provides the ability to configure up to 28 fast processors in a standard rack-mountable cabinet and enables customers to pack up to 14 of API's dual-processor UP2000 motherboards into each rack. The PowerRAC Chassis allows the creation of a high-performance computing solution with 42 GFLOPs of peak computing capacity, 28GB of SDRAM, 28 independent PCI buses and numerous gigabytes per second of memory and I/O bandwidth.

Contact: Alpha Processor Inc., 130C Baker Ave. Ext., Concord, MA 01742, 978-318-1100, 978-371-3177 (fax), info@alpha-processor.com, http://www.alpha-processor.com/.

ODBC-ODBC Bridge

Easysoft introduced their ODBC-ODBC Bridge, a new component in the Easysoft Data Access 2000 family of enterprise software components and services. By extending the ODBC API to Linux, the Easysoft ODBC-ODBC Bridge allows access to any ODBC-enabled database using standard Linux development tools and desktop applications. The Bridge is supported within Perl DBI DBD::ODBC, C, Python, Rexx/SQL and PHP and has been fully tested with leading office software such as Applixware and StarOffice.

Contact: Easysoft Ltd, No. 3 The Embankment, Sovereign Street, Leeds, LS1 4BJ, United Kingdom, +44-0-113-222-0400, +44-0-113-222-0500 (fax), sales@easysoft.com, http://www.easysoft.com/.

LinkScan 6.1

Electronic Software Publishing Corporation (Elsop) introduced LinkScan version 6.1. LinkScan is the industrial-strength link-checking and web-site-management tool for UNIX, Linux and other systems. It is accurate, scalable, fast and highly customizable. LinkScan creates two types of publication-quality SiteMaps and is professionally supported. LinkScan 6.1 includes the ability to import a simple list of links for validation, improved support for validating hyperlinks embedded in PDF documents and a new Extraheader command. It has been benchmarked at over 40,000 links checked per hour. Multi-threaded processing can check over 60 links concurrently.

Contact: Electronic Software Publishing Corporation, linkscan@elsop.com, http://www.elsop.com/.

Metro-X Enhanced Server CD

Metro Link's Metro-X Enhanced Server Set on CD-ROM is now available. The new CD release supports both Linux/x86 and FreeBSD/x86 and includes a printed manual. Metro-X 4.3, a robust X11 Release 6.4 server replacement, provides support for several fast, popular graphics cards including AGP cards. Touch-screen and multi-screen support are included at no extra charge.

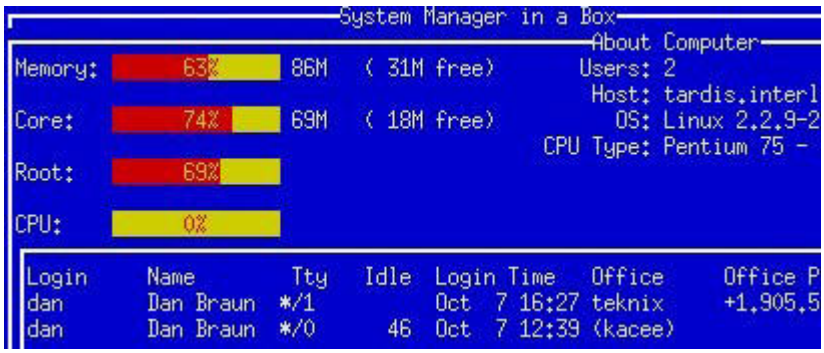
Contact: Metro Link, Inc., 4711 Powerline Rd., Fort Lauderdale, FL 33309, 954-938-0283, 954-938-1982 (fax), sales@metrolink.com, http://www.metrolink.com/.

P-STAT Statistical Software

P-STAT, Inc. announced a free version of the P-STAT software for Intel PCs running Linux. This free Linux version is fully functional and allows a file size of 500 variables and 5000 cases of data-perfect for teaching or real-world applications with limited data requirements. The P-STAT software combines data and file management, data entry and editing with report writing and statistical procedures.

Contact: P-STAT, Inc., 230 Lambertville-Hopewell Rd., Hopewell, NJ 08525-2809, 609-466-9200, 609-466-1688 (fax), sales@pstat.com, www.pstat.com/linux.html.

System Manager in a Box v1.0



PegaSoft announced the release of System Manager in a Box (SMiaB) v1.0. A multipurpose system administration tool, SMiaB offers tuning suggestions, locates configuration problems, monitors system activity and provides job control. Using artificial intelligence techniques, SMiaB can perform more than 2000 system checks—finding configuration mistakes, security loopholes and missing files—then report files which have problems and affected systems. Version 1.0 has been tested for Caldera 2.2, Mandrake 6.0, Red Hat 6.0, SuSE 5.3 and 6.1 and Slackware 4.0.

Contact: PegaSoft Canada, 2631 Honsberger Ave., Jordan Station, ON L0R 1S0, Canada, pegasoft@tiamet.vaxxine.com, http://www.vaxxine.com/pegasoft/.

PGI Workstation 3.1

The Portland Group, Inc. (PGI) announced the availability of PGI Workstation 3.1, the latest release of the company's suite of parallel FORTRAN, C and C++ compilers and tools. PGI Workstation 3.1 is supported on Intel processor-based workstations, servers and clusters running Linux, Solaris86 and NT. Among its included components are High Performance FORTRAN, OpenMP Parallel FORTRAN 77, OpenMP Parallel ANSI C and OpenMP Parallel C++ (Linux/ Solaris86 only).

Contact: PGI, 9150 SW Pioneer Ct., Suite H, Wilsonville, OR 97070, 503-682-2637 (fax), sales@pgroup.com, http://www.pgroup.com/.

Quick Restore 2.6

Workstation Solutions announced Quick Restore 2.6, an enterprise-ready and heterogeneous network backup and recovery solution for Linux servers. Quick Restore servers, clients and devices can be easily added, moved or reconfigured to accommodate changing network needs. Its design makes it easy to install, configure and use. Data stored on machines on the unsecure side of a firewall may be backed up and recovered.

Contact: Workstation Solutions, Inc., Five Overlook Dr., Amherst, NH 03031, 603-672-8600, 603-672-3154 (fax), http://www.worksta.com/.

Threads.h++ and Tools.h++ Professional

Rogue Wave Software announced the availability of Threads.h++ and Tools.h++ Professional on the Linux platform. Threads.h++ provides all the mechanisms needed to add multithreading to existing applications or write new multithreaded applications from scratch. Tools.h++ is a class library with more than 130 fundamental C++ building blocks, such as string, collection, date and time, internationalization and streaming classes, plus an easy-to-use interface to the Standard C++ Library.

Contact: Rogue Wave Software, 5500 Flatiron Pkwy., Boulder, CO 80301, 888-442-9641, 303-447-2568 (fax), websales@roguewave.com, http://www.roguewave.com/.

Scriptics Connect 1.0, 1.1

Scriptics Corporation announced the release of a full-production version of Scriptics Connect, their business-to-business integration server. Scriptics Connect is a complete XML platform, with robust and secure deployment as well as development tools. Version 1.0 includes the Apache web server on UNIX. Scriptics also unveiled a beta release of Scriptics Connect v1.1, which provides support for Red Hat Linux and Netscape Enterprise web servers. Scriptics Connect is designed to enable optimum server-to-server integration across the Internet.

Contact: Scriptics Corporation, 2593 Coast Ave., Mountain View, CA 94043, 650-210-0100, 650-210-0101 (fax), sales@scriptics.com, http://www.scriptics.com/.

TapeWare 6.2 Backup Software

Yosemite Technologies introduced TapeWare 6.2, the latest version of its data storage management software. TapeWare 6.2 now supports Red Hat Linux 5.2/6.x, offering convenience and backup features to make designing a comprehensive backup management program simple, easy to implement and customizable to unique network topologies. Other enhancements to version 6.2 include software compression, remote administration for TapeWare Lite, free auto-loader support on all editions, distributed devices, multiple concurrent devices and a single administration point.

Contact: Yosemite Technologies, Inc., 2750 N. Clovis Ave., Fresno, CA 93727, 559-292-8888, 559-292-8908 (fax), sales@tapeware.com, http://www.tapeware.com/.

DoubleVision for Linux Systems

Tridia released a Linux version of its remote control software, DoubleVision. It permits a remote system to connect to a remote console or another terminal (tty or pseudo-tty) and have full access to its screen and keyboard. Once attached, the screen looks like the screen of the terminal to which it is attached and the keyboard operates the remote terminal or console over a serial connection, modem or network.

Contact: Tridia Corporation, 1000 Cobb Place Blvd., Building 200, Suite 210, Kennesaw, GA 30144, 800-582-9337, 770-428-5009 (fax), sales@tridia.com, http://www.tridia.com/.

Contact Information

Archive Index Issue Table of Contents

Advanced search

Network Monitoring with Linux

Tristan Greaves

Issue #71, March 2000

Are you having trouble keeping your network under control? This article introduced NOCOL: the freeware network monitoring system which will help you keep instability at bay.

NOCOL, Network Operation Center On-Line, enables a designated machine to host a collection of network monitoring agents. These agents can perform a variety of tasks, from checking that a machine is “up” using the ICMP ping method to ensuring that a remote web server is operating as it should by requesting a test page. This allows problems on a network to be diagnosed and reported in a variety of ways, be it by e-mail, web page or dedicated terminal.

The alerting system works via escalation. Normally, any data reported is classed as INFO. However, if a service starts misbehaving, it can be flagged as either WARNING, ERROR or CRITICAL. If a problem is not dealt with, it will escalate (WARNING will move up to ERROR, ERROR will move up to CRITICAL). For example, you may have a machine which has to reboot itself periodically. You would therefore expect NOCOL to complain that the machine stops responding now and then. In this situation, you would class such an event as a WARNING. You will then be kept aware when reboots occur: if the event escalates up to ERROR or beyond, you'll know something has gone seriously wrong.

Most routers and similar equipment today are SNMP (simple network monitoring protocol) compatible, and several of the NOCOL agents have the ability to interrogate such devices.

Security Considerations

NOCOL does *not* need to run as root. The few binaries that do need to be privileged are set SUID root during the installation process. It is recommended that you create a user called “nocol” on your system for all NOCOL-related activities, including using it during installation.

Preparing for Make

NOCOL is available from ftp://ftp.navya.com/pub/. At the time of this writing, the latest stable version was nocol-4.2.tar.gz, which will be used for the purposes of this article.

NOCOL makes extensive use of Perl, so ensure that Perl is installed before continuing. In the unlikely event your Linux system does not already have Perl, obtain it from http://www.perl.com/CPAN/.

Once you have the NOCOL archive safely sitting on your proposed monitoring server (a 486/66DX machine with 32MB of memory sufficed for us), perform the magic:

```
gzip -dc nocol-4.2.tar.gz | tar xvf-
```

We installed NOCOL on a Red Hat 5.2 system, upgraded to allow use of the Linux 2.2.1 kernel. Enter the freshly generated nocol-4.2 directory, and then type:

```
./Configure
```

You will then be asked a few simple questions regarding your system:

- **Enter top-level directory:** The NOCOL tree defaults to being located at /usr/local/nocol, but you may adjust it to suit. Make sure the "nocol" user has permission to write to any directory you specify.
- **Enter location of man pages:** These reside under the main tree by default, but you may prefer them in the more "traditional" location on your system.
- **Enter extension for man pages:** I stuck with the default of **n** for this option.
- **Enter FULLY QUALIFIED name of your log host:** The server I set up for the main NOCOL monitors was also used for logging purposes, and this option does default to the host name of the installation machine. For simplicity, accept the default.
- **Where is your MAIL program located?** For NOCOL's e-mail alerting system to function, it needs access to the **mail** binary. The default of /bin/mail should work with most Linux installations.
- **Where should the operational e-mail go?** This e-mail address is for general NOCOL messages. Set it as appropriate.
- **Where should urgent/critical e-mail go?** Similarly, this e-mail address is for the urgent stuff (e.g., "The web server has exploded!").

- **Which compiler would you like to use?** Parts of the NOCOL system have been coded in C. The default choice of **cc** should suffice.
- **Which compiler options do you want (-DDEBUG)?** This is actually for developers, so accepting the default of **-O** will be fine.
- **Where is Perl located on your system?** Enter the path to your Perl binary here, accepting the default of /usr/bin/perl if that is correct.

Once this is done, you are ready to compile the software.

Compilation and Installation

Predictably, the compilation process can be set in motion by typing:

```
make
```

On our systems, **etherload** (a tool to monitor Ethernet load) fails to compile. **etherload** is not covered here, we hope this problem will be rectified in a future release.

Now install the software:

```
make install
```

Use **su** to log in as root and type:

```
make root
```

Expect another failure due to etherload not compiling.

That completes the installation procedure. Now all that remains is getting NOCOL to do justice to your network.

Configuring the Monitors

Sample configuration files for the monitors are installed in /etc/samples under your proposed NOCOL tree. Take a look at these to become familiar with how it works.

One of the first things you may want to monitor is whether machines on your network are up and running. The traditional way to do this is to see whether they are responding to a ping request.

To deal with UNIX machines on your network (those running an RPC port mapper), create a file called rpcpingmon-confg in the /etc directory, typing something like this:

```
POLLINTERVAL    300
kenny           kenny.your-network.com
kyle            kyle.your-network.com
cartman         123.123.123.123
```

The **POLLINTERVAL** indicates how often NOCOL should “sweep” the network. In our example, it will sweep every 300 seconds (5 minutes). Following that is a list of the machines to monitor: the first column is the “friendly name” and the second column contains the TCP/IP host name or IP address.

For non-UNIX machines (routers, Windows boxes, etc.), you should create a separate file called ippingmon-confg. The format is the same.

NOCOL includes many other monitors (see The NOCOL Suite) which you should investigate and configure to suit your needs. The sample configuration files do a good job of explaining their actions and how to set them up.

The NOCOL Suite

Final Preparations

A few minor scripts must be tweaked before NOCOL can start analyzing your network. Again, these are all located under the directory where you installed NOCOL.

The Perl script bin/keepalive_monitors handles the auto-starting of the monitors. Around line 32, you will find the following two lines (ignore wrapping):

```
PROGRAMS="noclogd etherload ippingmon rpcpingmon nsmon ntpmon
portmon"
PROGRAMS="$PROGRAMS radiusmon hostmon tpmon"
```

Alter these lines to include only the monitors you have actually configured. To match the two discussed here, you could condense them to one line:

```
PROGRAMS="ippingmon rpcpingmon"
```

The script bin/notifier deals with sending warning e-mails to the addresses specified during configuration. By default, it will send a single e-mail when a site has been marked “critical” for more than two hours. If you are feeling confident with Perl, you can specify additional addresses to contact after even more time has elapsed. Specify these addresses in the AFTER*x* lines:

```
AFTER2="
AFTER3="
AFTER5="emergency-team@your-network.com"
```

NOCOL comes with a custom crontab file which will automatically carry out any housekeeping required, such as ensuring all the monitors are running and rotating logs. To install it, enter the /bin directory in your NOCOL tree and type:

```
su nocol
crontab crontab.nocol
```

Starting the Sweep

To finally get NOCOL going, run the **keepalive_monitors** script located in the bin directory. Provided everything has gone well, the monitors will get to work.

If this fails, type **ps aux | grep nocol** (to see if the monitors are running), go back and check that you followed the instructions correctly.

Chances are, you will want to see what NOCOL is reporting. The simplest tool is **netconsole** which can be run either at the console or via a TELNET session. Run it and enter your terminal type when prompted (vt220, for example). The console screen will appear and will most likely be empty. The default is to show only CRITICAL events.

Pressing the **l** key lets you change the viewing mode. Set it to level 4 (INFO), and you will see all the information your configured monitors have gathered. See Listing 1 for an example. Play around with the levels until you find the one that most suits your needs. The **h** key will display a comprehensive help screen.

Listing 1

Linux's Serial Console Support

During the installation here, I found an old ICL DRS-10 serial terminal hiding in a cupboard. This terminal, or an equivalent, can be attached to a Linux box and used as a dedicated monitoring screen.

The exact settings required in /etc/gettydefs depend on the specifications of your terminal. For the DRS-10, we used the following entry for Red Hat:

```
# 9600 baud Dumb Terminal entry
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP \
CLOCAL #@S login: #DT9600
```

Now we need to edit the /etc/inittab file to present a login screen on the terminal. Be careful when playing with this file: it is possible to render your Linux system unbootable. Add the entry:

```
S1:3456:respawn:/sbin/getty ttyS0 DT9600 vt220
```

for a terminal connected to the first serial port (ttyS0).

Finally, force **init** to re-examine its configuration file by typing:

```
/sbin/telinit q
```

If all is well, your terminal should bring up a login prompt. From there, you can bring up netconsole in the usual fashion.

Setting up such a serial terminal is described in more detail in the Text-Terminal HOWTO (www.linuxhq.com/HOWTO/Text-Terminal-HOWTO.html).

Getting More Advanced

NOCOL has a web interface, included in the archive, and instructions for setting it up are found in the INSTALL file. In essence, this is a web version of netconsole which can be customized to look a bit more flashy (see Figure 1).

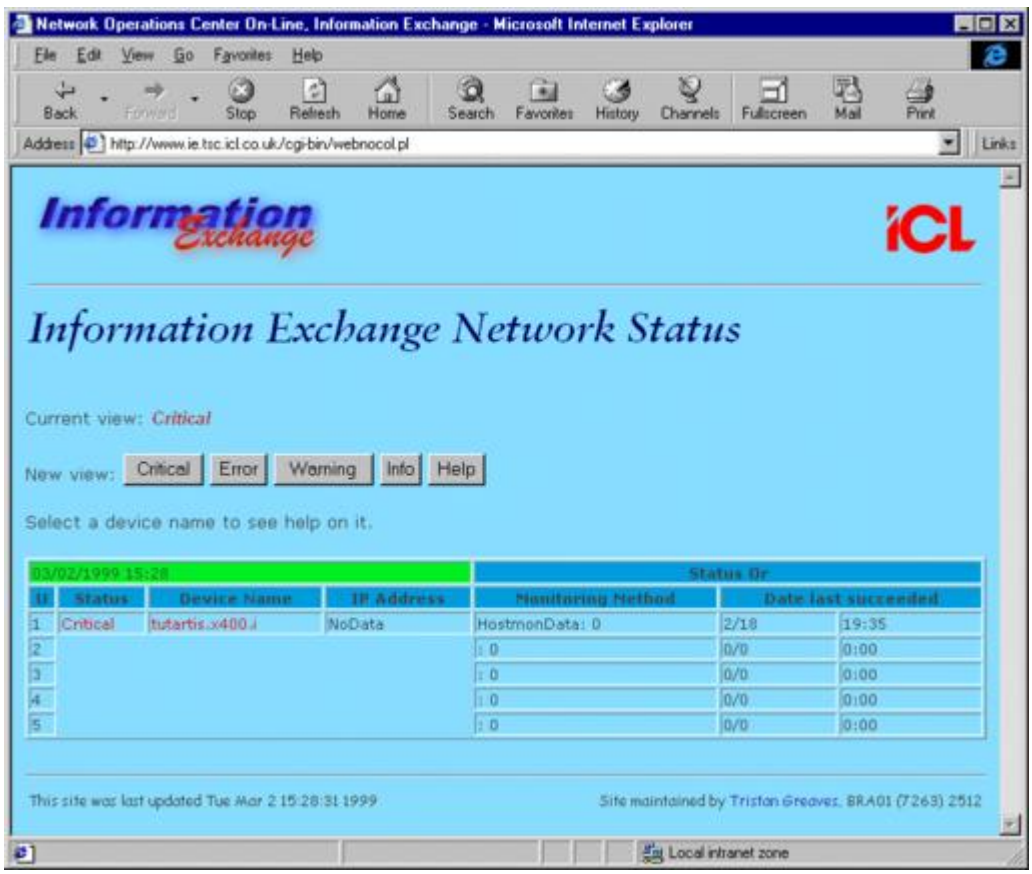


Figure 1. Web Version of Netconsole

The **hostmon** part of NOCOL is also very powerful. It allows you to install a Perl-based client on machines on your network in order to monitor aspects such as available disk space, mail queues, etc. The scripts can be extended to monitor any custom software you may be running. (We added an extension to monitor queues on our X.400/SMTP mail gateway software.)

An API to the system is provided that allows you to script your own monitors in Perl. Because of this, NOCOL has the power to monitor anything.

As an example of NOCOL's flexibility, I coded an extension to the **notifier** tool, which utilized our internal SMS messaging system. This allowed text messages describing CRITICAL problems to be sent to my mobile phone. This was done by coding an e-mail front-end to the SMS gateway, so all **notifer** had to do was fire off an e-mail in the correct format.

To Conclude

In essence, NOCOL has proven itself to be an extremely useful tool. It has alerted us to network problems as soon as they occurred, and the fact that it is freeware (it comes under a “not-quite-GPL” agreement) is just another example of great software under Linux being available for no cost.

Resources

Based in England, **Tristan Greaves** (tristan.greaves@icl.com) works as a Systems Integrator within the Information Exchange at ICL (the IT systems and services company). When not hacking Linux, he can be heard making sweet music with his band “Stash” in the Portsmouth area.

Archive Index Issue Table of Contents

Advanced search

An Introduction to Using Linux as a Multipurpose Firewall

Jeff Regan

Issue #71, March 2000

Feeling insecure? Here's a guide for getting the protection you need.

High-speed Internet connections are becoming more readily available and popular for home computer users. ADSL (Asymmetric Digital Subscriber Line), Nortel's 1MB modem and cable modems all offer connection speeds many times faster than that of a standard 56K POTS (plain old telephone service) modem that most of us know all too well. The other big advantage of these new services is that they are always connected. That is, you don't need to dial your service provider with your modem to start up your Internet connection. When you turn on your computer, the connection is already there, and your operating system will establish a link as it boots up.

Like the standard modem, these connections allow only one computer to connect to the Internet at a time. In some cases, additional IP addresses can be assigned to additional computers, but there is usually a monthly cost involved in providing this service.

By installing Linux on that old 486 you have sitting in the corner collecting dust, you can create a firewall so all the computers on your local LAN can see the Internet, and at the same time, transfer data back and forth between each other, (see Figure 1). You don't even need a dedicated PC. A faster PC can simultaneously be used for other purposes while acting as the firewall; however, there are two main drawbacks with this approach:

- Users on your LAN may experience a slower connection to the Internet.
- You could inadvertently open a security hole, allowing someone on the Internet to get in and play havoc with your system or files.

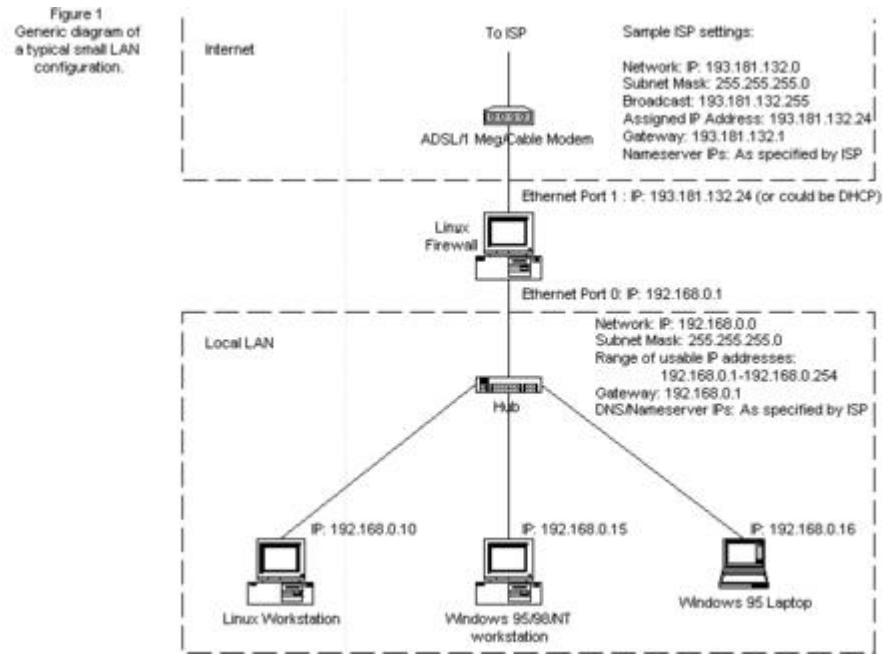


Figure 1. Generic Diagram of Small LAN Configuration

I will be discussing two different types of Linux firewalls. The first type consists of a 486 with 12MB of RAM, and a 200MB hard drive using either Red Hat 6.0 or Slackware 3.6. The second, called the Linux Router Project (LRP), uses a 486, 12MB of RAM, a 1.44MB floppy and no hard drive. Two Ethernet network interface cards (NICs) will be required, regardless of which firewall configuration you install.

Security

Someone is always watching, and people are always on the lookout for computers on the Internet with poor security. Their motivation can be as simple as boredom, or more seriously, a need to find a system to penetrate so they can use it to hide behind while they continue breaking into other systems, leaving evidence that points to you.

If you are running a standard Windows installation, you probably don't have the means to see who is trying to check out your machine. As long as “File and Print Sharing” is turned off inside of Windows, for the most part, you are safe. However, it is possible someone may find a new security hole in your PC and exploit it.

If you have Linux running, you can check out your system logs. Upon doing an informal survey with friends who run Linux firewalls, I found on average five attempts by outsiders each day to use TELNET or FTP to break into their Linux boxes. In the case of a firewall, you can turn off or restrict most services. In general, the strength of your firewall security decreases for each service you open up to the Internet, since each service is an invitation for someone to try and sneak in to your system. For example, if you open TELNET, someone can

use it to break in. A safer alternative is to restrict TELNET to certain incoming IP addresses, such as the IP addresses you might use to access your home system from work. If you have no plans to TELNET or FTP into your firewall from the Internet and all your traffic is originated inside your local LAN, you can lock your firewall fairly tightly. It is always a good idea to stay caught up on new security holes and the fixes for them. Check out http://www.cert.org/ for more information.

Theory of Operation

There are many reasons for having a firewall, some of which I have already mentioned. They include:

- Ensuring that local traffic on your intranet does not spill out to the Internet.
- Allowing the full use of file and print sharing in your LAN without having to worry about unwanted intrusions.
- Providing security for your LAN.
- Allowing yourself and authorized users access to your LAN to read e-mail, listen to MP3s or access file backups.

LAN and Internet Traffic Routing

When you copy a file from another local PC on your LAN using Windows "Network Neighbourhood", or when you FTP a file from another PC on your LAN, that traffic has no reason to go to the Internet. If you had a high-speed modem directly connected to your LAN, it would send out that data, because it has no way of knowing it should not be sent there. By default, it sends all traffic it sees, and although it won't likely get past the next router in the chain, it is sending out data that does not need to be there. This may impact the overall speed of your local LAN. You probably don't want this particular traffic to go out over the Internet, so the firewall prevents it.

One of the TCP/IP settings on our PCs, regardless of the operating system, is the "default route". When the destination IP address cannot be found on our local LAN (this is determined by the subnet mask), then the default route is used. The default route in this example will point to the IP address of the NIC on the local LAN side of the firewall (Ethernet 0 in Figure 1). This IP address usually ends in 1. For example, if you have a local LAN with a network address of 192.168.0.0 and a subnet mask of 255.255.255.0, you have 192.168.0.1 to 192.168.0.254 available for local IP addresses (see Resources for more information on the Linux NET-3-HOWTO). In this case, 192.168.0.1 would normally be assigned to the NIC on the firewall.

Any traffic intended for an IP address outside our local LAN will go into the firewall. The firewall will replace (masquerade) the source address of the PC in the local LAN that originated the packet with the firewall's IP address (assigned by your ISP), so that to the Internet, the traffic looks as though it originates from the firewall and is coming from a valid IP address. Any return packets relating to this originating packet will go through the reverse transformation, so the traffic finds its way back to the originating PC.

Rules can be set up to allow certain packets to make it through the firewall or to be stopped dead. By default, nothing gets passed. A small set of rules are needed to support features such as TELNET, HTTP, IMAP and POP3, and a few extra rules are needed to allow other features such as RealAudio or on-line gaming to function. (Gaming can be a little more difficult to set up, as each game is different.)

Designing the Network

Table 1

In Figure 1, you can see how a typical small LAN/firewall configuration might look. You need to determine how many PCs will be in your network, and how many of them will be connecting to the Internet. The IP addresses chosen for your internal network will be determined by the size of the network. Table 1 shows which groups of IP addresses have been reserved for private LANs, such as the one we are designing. For the most part, a class C network address will be sufficient, as it will allow up to 253 hosts or PCs in our LAN, leaving one IP for the firewall. Table 2 shows an example configuration.

Table 2

The complete firewall will be built over several stages. These include building and configuring the hardware, installing and configuring Linux, configuring the network cards, building a new kernel, establishing routing between the networks, then introducing security and locking down the PC and the local LAN.

Building Up the Hardware

First, you must decide what type of system you want to build. If you want to use your firewall only for firewall/routing purposes, then once it is set up and running, it does not need a keyboard or a monitor. In fact, many systems will run without a video card; however, you might want to keep one handy in the event of a system failure. Software changes can be done by either connecting to the firewall over your local LAN via TELNET, or using a modem program on a laptop (such as Hyperterm) and connecting to the firewall via the serial port. This type of configuration is well-suited to LRP. If you actually want to have a

few users on your machine reading things like e-mail (either locally or via TELNET), you will need a hard drive and RAM sufficient to handle that. A 200MB hard drive and 16MB of RAM will work for this if you don't load unneeded packages, such as the X server or source code, and your users keep space constraints in mind. If you plan on the LAN using your firewall PC for additional functions, you will need to upgrade it appropriately in all respects: memory, hard drive size and processor speed.

You will need at least two network cards in your firewall. One card will face the Internet and the other will face your local network (Figure 1). If you can't afford a hub and you have only a few PCs to connect, you can put multiple cards in your firewall, one for each PC, and wire an Ethernet cable as a “turn-around” cable. ISA network cards can be found inexpensively in some markets these days, and may be less expensive than an Ethernet hub. The use of more than two network cards in your firewall machine will require more rules in your firewall, but that is easily handled.

You will need the DOS configuration disks for your network cards if they are jumperless cards which use non-volatile RAM (NVRAM) to remember their settings (I/O address, IRQ, etc.). The configuration software for most cards can be found on the Internet at the card manufacturer's web page or at some of those helpful Windows driver repository sites.

Make a DOS floppy boot disk, and have the configuration program for each card handy on floppy.

Table 3

Install one network card at a time and boot your PC. Run the configuration software for that card, and set the I/O address and IRQ settings. Make sure you don't configure the card to a setting already in use by some other card. For I/O addresses, the only item you may have trouble with is an old CD-ROM drive with a proprietary controller (see Table 3). Once configured, remove the card, insert the next network card and repeat the procedure. Once you have your network cards individually configured, you can install them all in the firewall. In my firewall, my first network card is set to an I/O address of 300 and an IRQ of 12, while my second network card is set to an I/O address of 320 and an IRQ of 15.

It is now time to install Linux. The sample configuration that follows is based on Linux kernel 2.2.9. If you install a Linux distribution from the Net or from a recent CD, you may find this kernel included. If not, you can get it from http://www.kernel.org/. The more recent the distribution, the less likely it is that you will run into outdated libraries or utilities. One of the utilities we will be using to

control the firewall is called **ipchains**. This program runs only on kernel version 2.2.*x* and higher. If you plan on using an earlier version of the kernel, you will need to find **ipfwadm**. It is always best to use a recent (but not necessarily the most recent) kernel version. Follow the instructions provided with your distribution, and install the distribution. If the default kernel on the CD is not of the 2.2.*x* variety, don't worry; you will need to build a new kernel later anyway. If you are building a small system, you will want to install as little of the distribution as possible. At a minimum, you will need to install the base files and networking support.

Configuring the Network

At one point during the installation, you will be asked to configure the Ethernet interfaces (ports). Generally, you will be able to configure only one of the interfaces during the installation. The remaining interfaces can be configured by editing the configuration files. Alternatively, Red Hat 6 offers a GUI-based application called **netcfg**; however, it requires you to install the X server, something I don't recommend if you are tight on hard drive space or don't plan on leaving a monitor connected to the PC. When you do come across the configuration request for the first interface (generally called eth0), you should enter the information for your local LAN. In our example as per Table 2, we configure this interface as:

```
IP address: 192.168.0.1
Subnet Mask: 255.255.255.0
```

Listing 1

The default gateway of the firewall (not the PCs inside your LAN) is that of the gateway provided by your ISP. If the subnet mask provided by your ISP ends in a .0, your ISP gateway IP address will generally end in .1, for example 193.181.132.1. After the setup and installation of the distribution is complete, you will have to add the additional information on your second Ethernet interface (generally called eth1). We will need to edit or create configuration files for both Slackware (Listing 1) and Red Hat 6.0 (Red Hat sidebar).

Red Hat Configuration Files

If your NIC cards are all of the same type or all use the same driver, you must tell Linux to search for more than one card of that type at boot time. LILO provides a nice way of doing this that works for most Ethernet drivers I have tried. Edit the file /etc/lilo.conf and add the line

```
append="ether=0,0,eth1"
```

anywhere in the LILO global section near the top of the file. If you have more than two Ethernet cards, you would add

```
append="ether=0,0,eth1 ether=0,0,eth2"
```

You can also explicitly define all the cards instead of just telling the system to look for additional cards by using the following on one line:

```
append="ether=irq_card0,io=0xaddress0,eth0 ether=irq_card1,io=0xaddress1,eth1"
```

In the example of my configured cards above, I could then use

```
append="ether=12,0x300,eth0 ether=15,0x320,eth1"
```

Don't forget to type **lilo** after you have finished editing the file so the new **lilo** parameters are read and installed, but, more importantly at this point, so you know you have not created any errors in the LILO configuration file.

Do not reboot yet, as we still need to build a kernel to support our various hardware and firewalling needs.

Building the Kernel

Listing 2

A variety of configurations are required to make the kernel run. Listing 2 shows the settings I have used in my system. If you have never built a kernel before, see “Linux Kernel Installation” by David Bandel in the November 1997, issue 43 of *Linux Journal*. A quick summary is as follows:

```
cd /usr/src/linux
make menuconfig
```

Look at the many screens, read the help and any other reference documentation it points to. This will help you determine which options you need. After you have finished choosing your options, save the kernel, then type:

```
make dep
make bzlilo
make modules
make modules_install
```

The resulting files, the new kernel file called vmlinuz and a new System.map file, will likely be located in the root directory /. You will need to copy the System.map file to the root directory, like this:

```
cp /System.map root
```

Also make sure the file /etc/lilo.conf and the line inside it which reads **image=*IMAGENAME*** (where ***IMAGENAME*** is the name and location of your kernel used at boot time) is correct. If it does not point to the correct location, change

it and re-execute the lilo command to complete the process of setting up the new kernel.

This will build and install the Linux kernel, update LILO to reflect the new kernel and install a variety of modules, such as support for RealAudio which by default is blocked by the firewall.

Listing 2 includes only those options required to make the firewall function. Other options such as processor type are left out, since these are specific to the hardware you are using for this project. As a rule, I put as little in the kernel as required, and I minimize the use of modules. If you are not sure how an option I have shown in the table is used, or where it shows up in the kernel-configuration program, you can match up the item by clicking on help for the items in that section. You will find its configuration file name at the top of the help page. Similarly, if it's not shown in Listing 2 and you don't need it to make your hardware run or support some other feature, then it should be set to off.

If you are building a bare-bones system and are going to compile the kernel elsewhere, be sure to save your kernel on that machine first, and also save that PC's kernel configuration in an alternate file (see the bottom of the kernel configuration program menu). After it is built, you need to copy the files over to the firewall PC via sneaker net or LAN. Don't forget to copy the modules installed in /lib/modules/2.2.9 as well.

It is now time to reboot the PC and cross your fingers. If everything works correctly, both Ethernet cards will be recognized, and they will both be configured. When the system is fully booted, log in as root and type **ifconfig**. It should show detailed information about three interfaces:

- lo0: the local loop interface
- eth0: the Ethernet port pointing to your local LAN
- eth1: the Ethernet port pointing to the Internet

You can also type **route** and see what default routes are up. It should show a default route to the Internet, as well as some information about your local LAN. At this point, the firewall should be able to see both your local LAN and the Internet. If you hook your high-speed modem to the eth1 port at this point, you should be able to **ping** sites on the Internet (e.g., **ping www.linuxjournal.com**) and see an answer coming back about once per second. Press **CTRL-c** to stop the pings. Some high-speed modems need to learn your Ethernet card's MAC address, and only do so each time they are turned on. Therefore, if you are connecting your modem to a different Ethernet card than it was previously connected to, you will have to power off both the modem and your firewall PC,

power the modem on, wait a few seconds, then turn the PC on. If you don't do this, you may find you can't see the Internet at all.

Locking Down the Firewall

We currently have a Linux PC, connected in the middle of two networks. It can see both, and both can see it. The PC is also wide open with all the default ports turned on. We want to restrict this as much as possible. People are always looking for new ways of breaking into systems. The more we lock down this firewall to the outside, the less vulnerable we are to attacks. Nothing is perfect, and the only true way to be sure people are kept out is to unplug your Ethernet connection when you are not there. Since that's undesirable for most of us, this is the next best thing.

What needs to be done now is disabling all services we don't need. If you are making this a true firewall, you can disable almost everything except TELNET and FTP, and these two will be limited to ports from only inside your LAN and trusted outside IP addresses.

Listing 3

The file /etc/inetd.conf, as shown in Listing 3, is where these ports are configured. This file affects traffic terminating at the firewall, not passing through it. Disabling something like POP3 or IMAP is acceptable, since when you go to get your mail from a PC inside your network, this traffic will pass through the firewall (but not stop) on its way to your ISP's POP3 or IMAP mail server.

Remember, the more ports and addresses you choose to leave open, the more closely you will need to watch your firewall for break-in attempts. We have left TELNET and FTP open, so we'll want to restrict the originating IP addresses on both networks to those we want to let in.

Setting Services

This is done by editing the files /etc/hosts.deny and /etc/hosts.allow. By editing these files, you can deny access to everyone except a few specific addresses or range of addresses, or you can allow everyone in by default and disable problem IP addresses down the road when you discover unwanted access from those points. If this is the case, be sure to watch your system logs closely. See the “Setting Services” sidebar for more details. In one sense, we could have left inetd.conf alone and restricted people from those ports via the /etc/hosts.deny table; however, it is always best to lock down ports in multiple ways.

By default, most UNIX systems do not allow root to log in from anywhere but the console. If your system is not set up that way, it should be. You will at least want to slow down someone who might want in your system. If they can't log in directly as root, this is an additional security benefit. Check the file /etc/securetty. In Red Hat 6.0, look for pty1, pty2, etc. entries in the table. In Slackware, look for ttyp0, ttyp1, etc. entries in the table. If these entries are in place, root login is allowed on those TELNET ttys; therefore, remove the entries. The other remaining entries in the table cover your various consoles and serial ports.

Since you can't log in remotely as root and you do not have a console with a monitor and keyboard, it would be best to add a second user to the firewall to ensure you can “su to root” to do work on the firewall.

```
useradd -g 100 -d /home/USER -s /bin/tcsh -c\
 'YOURNAME' USER passwd USER
```

The **-g** controls which group this user will belong to. In this example, 100 was used, as this is the user's group in Red Hat 6.0. If this does not work for you, check out /etc/group to find a suitable group. *YOURNAME* is whatever you want to put in the Name field of the user account, and *USER* is the ID chosen for the user, i.e., I may choose to use jeff as my ID.

Stopping Extra Processes

In a small system, the only processes we want running are ones that pertain to the operation of the firewall. This means disabling processes: all but one or two consoles, Sendmail and anything else you don't need. You can see what is running right now by typing:

```
ps -xa
```

To keep Sendmail from starting next time, you will need to move or edit the file where it starts. Linux usually starts up in runlevel 3. In Red Hat 6.0, you can check that by looking at /etc/inittab and looking for the line that reads **id:3:initdefault:**. The 3 indicates runlevel 3. Therefore, in /etc/rc.d/rc3.d, there is a file called S80sendmail. Move this file to 80sendmail, as follows:

```
mv /etc/rc.d/rc3.d/S80sendmail\
/etc/rc.d/rc3.d/80sendmail
```

Some programs like **elm** require that sendmail be running to operate properly. This opens up a potential hole to to the outside world since it also means port 25 will be open to possible attacks and possibly even mail relaying—allowing others on the Internet to use your firewall to send out spam mail. Turning off

port 25 access is the easiest way to prevent this problem. Other solutions can be found at http://www.sendmail.org/.

In Slackware, edit /etc/rc.d/rc.M and change the line:

```
/usr/sbin/sendmail -bd -q15m
```

to:

```
/usr/sbin/sendmail -bm -q15m
```

In Red Hat 6.0, edit /etc/rc.d/rc3.d/S80sendmail and change the line:

```
daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bd) \
```

to:

```
daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bm) \
```

Creating the Firewall

Currently, we have a reasonably secure PC quite incapable of passing the network traffic from the local LAN to the Internet. It is now time to set up and configure the rules that will make our firewall function. As mentioned earlier, these rules allow acceptable packets to pass through the firewall, while still offering various levels of security to unacceptable packets.

Download (with FTP) the ipchains package from http://www.rustcorp.com/ipchains/. Follow the installation instructions you obtained with the package to install it on your system.

Listing 4

Listing 4 shows the /etc/rc.d/rc.local file which is used to start any process not normally started as part of the distribution's installation. It is here where we set the rules for our firewall. Since our firewall is fairly straightforward, all we need to do is set up forwarding of masqueraded packets. To be able to use the full functionality of FTP, RealAudio, IRC and others, we need to support their port requirements as well. Many of these can be supported using the ipchains command above, but there are loadable modules that will take care of this, such as those shown in the sample rc.local file in Listing 4. See /lib/modules/2.2.9/ipv4 for a list of modules supported in your kernel. This directory should have been created earlier when you built the kernel.

That should do it. You are now ready to test your network firewall. Set one of your PCs inside your local LAN to one of the sample settings shown in Figure 1.

For example, on Windows 95, you will need to enter a local LAN IP (such as 192.168.0.10), a subnet mask of 255.255.255.0, a gateway IP of 192.168.0.1 and DNS entries given to you by the ISP. If the high-speed modem was originally connected to this PC, the DNS entries in the PC should already be set.

To test out your new firewall, try connecting to a web site with one of the PCs on your internal LAN. Try using RealAudio, FTP and other functions you regularly use. If none of these work, try using TELNET to get to the firewall PC. If you can do so, and you can ping a site on the Internet (or get to it via TELNET) from the firewall PC, check your rules in the /etc/rc.d/rc.local file, as you might not have turned on IP forwarding. If web access works, but (say) IRC does not, check to see if you loaded the IRC module correctly. Use the command **lsmod** to show which modules are loaded.

Building a Firewall Using the Linux Router Project

The configuration of LRP I will describe also uses the setup in Figure 1. It was set up on a 486 with 12MB of memory, a 1.44MB floppy drive, two Western Digital ISA network cards and no hard drive. For your system, install and configure the network cards in the same way as for the full firewall build earlier in this article. LRP version 2.9.4 is based on kernel version 2.0.36. This kernel is older than the 2.2.9 used above, and as a result, does not offer some features you may require if you want an advanced firewall. By the time you read this, there will likely be a new version available based on version 2.2.*x* of the kernel. I will describe setting version 2.9.4, and if you need some of the 2.2.*x* features, you have a foundation from which to work.

LRP uses a DOS-formatted floppy, either formatted as a standard 1.44MB disk or larger. (A utility called **2m** can squeeze additional, usable storage space out of a floppy.) During boot time, a RAM disk is created, which is used as the live file system. Various portions of the system are created from compressed archive files (tar) that end in .lrp and are found on the floppy. In general, the floppy can run with write protect on. This means if someone were to find a way in to your firewall, any changes they made would disappear when the system is rebooted.

LRP is available in many forms. The hard way is to create a disk, make it bootable using a program called **syslinux**, and install the kernel and various LRP files required. However, at ftp://ftp.linuxrouter.org/linux-router/dists/2.9.4/, you will find in the download section a file called idiot-image_1440KB_2.9.4. The name might not be flattering, but it is the easiest way to start building an LRP disk. After you get the file via FTP, copy it to the floppy in one of two ways. In DOS, use the **rawrite** utility that came with your Linux distribution. In Linux, type:

```
cp idiot-image_1440KB_2.9.4 /dev/fd0
```

I have assumed /dev/fd0 is your 1.44MB floppy, but if it is not, change fd0 to the correct device name.

Now go to http://www.linuxrouter.org/modmaker/ and make a kernel that includes hardware support for our network cards and includes any modules required to support FTP, RealAudio, etc. This web site is a very nice way to generate a kernel. Click on 2.0.36final and tick off the modules you require. Unless you know you don't want support for one of the few masquerading modules in this list (like IRC), tick off all options that start with ip_masq such as ip_masq_irc and ip_masq_ftp. Then go down the list and find the drivers for your hardware. You may have to do some research as to the driver your NIC cards require. If you don't know which driver to pick, run **make menuconfig** on a working full Linux system and look at the devices under Network Device Support. When you find your card, look at the help and find out its module name. This module name is what you need to tick off on the module maker screen you are looking at. If you want to support serial port access to connect to your firewall using the serial port to do maintenance, select the serial module as well.

Once you have made these choices, click “Create modules.lrp file” and download a copy of modules.lrp and a copy of the kernel. Copy these files to your floppy using either the DOS **copy** command or **mcopy** in Linux. When you copy your downloaded kernel to floppy, name it linux. You will have to overwrite the existing modules.lrp and linux files.

You are now ready to boot the system. It should come up, but may or may not see one or both of your network cards, as we might still have to add some options to the modules setup as discussed below.

Listing 5

Log in as root. You will get a menu with various configuration options available. Select 1 for Network Settings and 1 again for Network Configuration (auto). Listing 5 shows a sample of the file /etc/network.conf. Edit the file on your screen to meet your needs, using mine as an example. I have placed as many details in the example as possible. Save the file by pressing **CTRL-W** and go back to the menu by pressing **CTRL-C**.

Next, edit the /etc/hosts.allow and /etc/hosts.deny files, using menu options 4 and 5. The same rules apply now as they did when the first firewall was built. Refer to the sidebar “Setting Services” for more details.

Quit back to the main configuration menu, then go on to option 2 for the system settings. If you wanted serial port access as mentioned earlier, you will need to edit the /etc/inittab file by selecting option 2 from the menu. You will find the serial port access commented out towards the end of the file. Uncomment the line to make it function.

Listing 6

The last step is configuring the modules to support the network cards, if they were not automatically detected when you booted up. Quit back to the main configuration menu and select 3 for Package Settings, 2 for modules and 1 for Kernel Modules. Find the line that shows the module for your system. Listing 6 is an example of the /etc/modules file. For my WD Ethernet cards, I use the configuration information to tell the WD driver exactly where to find each card.

One last thing to avoid possible problems: delete the /etc/resolv.conf file. If you have one lying around, it will be used even if you turn off DNS support in the configuration menus.

That's about it. Make sure you change the root password, and don't forget to back up your changes (currently, they are changed only on your RAM disk) to floppy via the backup option from the main menu. You should be able to boot up with a basic firewall, and it should function similarly to the one we built above. It is less configurable only because of the lack of a hard drive, but you can build on it. If you quit the menu program and get to the shell prompt, but want to get back into the menu, type **lrcfg** and press **ENTER**.

Using a Dynamic (DHCP) Address

Some service providers do not assign an IP address to your connection, but rather assign you one each time you boot your PC, and at regular intervals afterwards. This is known as DHCP, or Dynamic Host Configuration Protocol. Some providers give you a choice of which you want to use. The decision to use one or the other depends on your situation. DHCP means it would be more difficult for someone to find your PC and break into it over a long period, because your IP address would change from time to time. However, if you wish to use TELNET or FTP to log in from the Internet, you would have to find a way to notify yourself of your firewall's new IP address each time it changed.

Red Hat 6.0 makes it easy to set up one interface for DHCP support. For the NIC facing the Internet, edit the /etc/sysconfig/network-scripts/ifcfg-eth1 file and set it up for DHCP use as per the Red Hat sidebar. If you have an X server available, you can also edit the settings via **netcfg**.

For both Slackware and the LRP, it is not quite as easy. DHCPCD is a program that comes installed on the latest Slackware, but may require configuration. For Slackware, you can start **dhcpcd INTERFACE**, where ***INTERFACE*** in our example would be eth1. Once that is in place, you would then create your routing tables. Therefore, start DHCPCD by placing this line in the /etc/rc.d/ rc.inet1 file as shown in Listing 1. You will have no need to define the second card as we did in the static IP examples discussed previously.

For LRP, you will have to use FTP to get DHCPCD and build it on your system (remember that .lrp files are in fact groups of compressed (zip) files that make up the file system in a tar archive). By the time you read this, there will hopefully be a DHCP.LRP file available for LRP. For more information on DHCP setup, take a look at the mini-FAQ listed in Resources.

Recent Developments

In the past several months, some service providers have rolled out Point-to-Point over Ethernet (PPPOE). As the name suggests, this is somewhat similar the Point-to-Point (PPP) used for most dial-up connections. There are variety of reasons why this is being introduced. Some reasons are the provision for access control as well as billing based on the user's amount of use. Because it is similar to dial-up PPP, you will likely have a different IP address assigned each time you boot your PC, as well as at regular intervals if you never turn your PC off. I have seen IP addresses change anytime after a few days to a week in the one installation I observed. For the service providers trying prevent their customers from running a web page or FTP server on their high-speed connection, the changing IP address makes it easy to discourage such uses.

To be able to communicate with a provider using PPPOE, you will require PPPOE compatible software/drivers on your Firewall.X Some service providers will supply at least the software and installation instructions, if not the full technical support for the installation of the PPPOE software itself. If no software is available from your service provider, look up PPPOE on the web, and you will should find enough information to get you started. The basic firewall configuration does not change, only the configuration of the Ethernet interface facing the ISP.

Summary

What I have presented to you is only a beginning. It may work very well for your application, but you may also want to expand on its abilities. I have added a third network card to my firewall so I can support a second local LAN segment. This works out well because it saves me from having to purchase another hub, and it also means that while I am using a lot of bandwidth, such as copying files on the local LAN, I am not chewing up bandwidth on the other local LAN, so

that user who happens to be the only user on the second LAN segment can access the Internet without any difficulties. The firewall at that level acts as a router, and allows the user to print to my printer or have access to files on this segment, while preventing outsiders from doing so.

My firewall also acts as a DHCP server (using **dhcpd**) for my laptop, so that when I am out on the road or at home, I can leave the PC set for DHCP in both cases.

Other uses you may consider include allowing a small number of outside users to read their mail via a shell account on your firewall (they can IMAP or POP their mail and use something like Elm or Pine to read it). You could also create a small web server for either the internal LAN or the external LAN on your firewall. Shell accounts and Apache require you to enable services that would otherwise have been disabled.

I have also had the opportunity to see some virtual private networking over a Linux firewall. This could be used to allow you to access your internal LAN via the Internet and add encryption and other features. This is handy if you are on the road a lot and might normally call long distance to get into your LAN. Just dial up an ISP wherever you are currently located, and off you go.

You can also increase the overall security by implementing additional rules to prevent IP spoofing in the full firewall. These rules are already included in the LRP.

As you can see, you can do a lot with that old 486 sitting in the closet gathering dust. It can make a wonderful firewall, and allows you some flexibility if you have more than one PC you want to connect to your high-speed modem. Just remember to do what you can to keep up with any new security holes that someone may have found. Check the resources at the end of this article to learn more about where to look for more ongoing security information.

Resources

Jeff Regan is an electrical engineer at a Canadian telecommunications provider. He started using Linux in 1993 with kernel version 0.0.93. He can be reached at jeregan@computer.org.

LUIGUI—Linux/UNIX Independent Group for Usability Information

Randy Jay Yarger

Issue #71, March 2000

A new Linux group has been organized to look at user interfaces and help formulate a standard in an effort to ease the way for Linux to move onto the desktop.

Over the past year, public exposure of Linux has increased beyond anyone's expectations. Company after company has embraced Linux as a superior solution for server applications. Despite all this, one mantra of negativity has continued to dog Linux: “It's not suitable for desktop use.”

Even when extolling the virtues of Linux as a server, the praise is often followed with a few discouraging warnings. Linux is difficult to install. The command line is too cryptic. The X Window System's applications have no consistency. Linux doesn't work with the major Windows and Macintosh applications. Linux is only for engineers who like long strings of numbers. (I've actually heard this last one, believe it or not.) They all boil down to “Linux is too hard for the average user.”

While many people, even Linux users, do believe the above, for each assertion against its ease of use, there is a group of people who know it's not true and can prove it. Anyone who has installed Windows from scratch can testify that it is hardly a simpler process than installing Red Hat or Caldera OpenLinux. Likewise, anyone who has attempted to move a set of files from one machine to another using standard DOS commands can appreciate the sheer usefulness of **tar**, **gzip** and other commands.

In these cases, as well as many others, the simple fact is that other systems seem simpler only because most people don't encounter these situations when using them. When you buy a new PC, Windows is already there, pre-installed. Most common file operations are performed using point-and-click and drag-and-drop, not with command-line utilities. For years now, most versions of

UNIX have offered similar ways of manipulating files in a quick and easy graphical manner.

Where do these misconceptions about Linux originate? Like most things in life, people believe what they see and what they hear. We know what people are hearing about Linux, so what do they see? Typically, a person's first encounter with Linux is from a friend or colleague who is an established Linux user.

Most users who have spent a considerable time with Linux have learned the “Great Secret of UNIX”. The one feature that UNIX (and Linux in particular) has which no other operating system has been able to duplicate is its infinite configurability. Every moment of the Linux experience, from startup to shutdown and everything in between, can be tweaked, tuned and completely subverted to create a uniquely personal environment.

This Power Is Exactly the Problem

People who see Linux for the first time invariably encounter a system that is efficient, streamlined, and to their eyes, completely incomprehensible. The same features that make Linux attractive to those who know it, keep those who don't, away.

So what is the solution? How do we take the one system that really can be all things to all people and present it in an intelligible way to newcomers?

The answer may lie, at least in part, in a new movement known as the Linux/ UNIX Independent Group for Usability Information (LUIGUI). Dr. Nathaniel Borenstein of the University of Michigan announced LUIGUI at the inaugural meeting of the U of M branch of SIGCHI (the Computer and Human Interaction group of ACM). Dr. Borenstein's name is familiar to many in the computer world. Among many other accomplishments, he is the inventor of MIME, the recognized standard for identifying and transferring data over the Internet. Dr. Borenstein has also been an active participant in the push for responsible Internet development, an author and an entrepreneur.

LUIGUI is an attempt to formulate the ideal interface for those new to Linux. Taking the project to its ideal conclusion, anyone trying Linux for the first time would be guaranteed several things.

Intuitive installation: one of these days, you will be able to go to your nearest Best Buy or Sears and purchase the latest Compaq or Sony PC pre-installed with the industry standard desktop operating system: Linux. Okay, I hope that will happen. However, until it does, a PC will come with Windows, and anyone who wants to try out Linux will have to install it. Therefore, the process should be simple and painless, even enjoyable. This is an essential step for Linux. It

doesn't matter how many impressive demonstrations their friends have given —a person's opinion of an operating system is solidified the first time they sit down and use it on their own.

Full functionality: Linux, at its very heart, is simply the name of an operating system kernel. Linux provides the backbone of a useful system, but no utility on its own. What we commonly call “Linux” is actually the Linux kernel in combination with a wide variety of tools and applications. In fact, the sheer number of standard Linux utilities created by the GNU project alone has led to suggestions that the full Linux distribution be named “GNU/Linux” or even “LiGNUx”. While that can of worms is best left unopened, what is undeniable is that the quality of applications available for Linux has been as much a factor in its popularity as the quality of the kernel. Taking this idea further, there is a core set of functionality that every user needs, or at least should be able to access. Word processing, a spreadsheet, image manipulation—these are all tasks that have become everyday activities in the world of desktop computing.

Ease of use: this area is the key for many people and is where much of the current development work (outside of LUIGUI) is being focused. Projects like KDE and GNOME are proving that, yes, Linux can have a GUI that the average person can use comfortably. This is the “pretty face” Linux desperately needs. Users should be presented with an initial desktop that is simple, elegant, and above all, understandable. It should take minimal thought to launch an application, navigate the file system or do any of the other common tasks that users perform through the desktop.

Ease of configuration: this is an area where Linux presents two extremes. On one hand, the configurability of Linux is unmatched by any other mainstream operating system. This is very important, as having a user-friendly configuration system is not much good if there is little the user can configure. However, in the presentation of configuration, Linux is still lacking. Users should have a method of customizing their environment that is as easy to use and understand as the environment. The fact that Linux has so much that can be configured makes this task even more difficult, but considering the feats the Linux community has been capable of so far, it should not be beyond reach.

Standardization: this is, for LUIGUI, both the hardest and the most important area. All the above criteria can be met, in some form or another, by Linux today. New, easy-to-use installers are becoming common on the major distributions. Company after company has announced applications for Linux users, and the Open Source community is matching their output. The desktop environment is friendlier then ever, and graphical configuration utilities are appearing for everything from adding users to configuring high-end web servers. However, what every new Linux user must be guaranteed is an

environment that is predictable. A single, standard desktop environment, meeting all of the criteria above should be presented when a Linux system is booted for the first time. Naturally, people will still customize their environment, but if this standard is as good as we know it can be, a large enough base of users will form so that this desktop environment will be used as a sort of lingua franca with which all Linux users can communicate.

At first glance, accomplishing the above seems like a near-impossible task that would take millions of dollars of funding and years of programming talent. However, Dr. Borenstein approaches this task with a surprising mantra: “Zero lines of code.” While not meant literally, since new tools will inevitably have to be written, his statement points to the heart of the matter: almost everything we need is already present.

“People have been trying to improve the user interface for UNIX for over 20 years, and every single attempt has succeeded. It's like shooting fish in a barrel”, says Dr. Borenstein. The tools and configurability UNIX provide make this task technically simple. However, no consensus has ever been reached on what the best standard interface would be for new users and all users in general. The one graphical system that has become standard, the X11 windowing system, explicitly made no choice on what the standard interface should be and merely provided the tools to create new interfaces. This is where LUIGUI comes in.

Dr. Borenstein has called LUIGUI the “Consumer's Report” of Linux. His intent is that the members of LUIGUI will evaluate all of the various user-interface options currently available for Linux. Through methods used by human-computer interaction professionals and computer interface design specialists, it should be possible to take the various pieces that exist now and combine them to form the “ideal” standard interface for Linux. After individual pieces of the interface are evaluated, the best of the crop will then be advocated by LUIGUI to the rest of the Linux community. If enough public trust and support is placed in the project, then hopefully the major Linux distributors will make some effort to implement the standards proposed by LUIGUI.

The LUIGUI project is spearheaded by graduate students in the University of Michigan School of Information. The self-described “project cheerleader” of LUIGUI is T. Charles Yun, a master's student in the U of M's SI program. Yun, along with web-site manager Katherine Degelau, created the LUIGUI home page at http://www.luigui.org/. A list of the current LUIGUI projects and goals can be found there.

The LUIGUI project is still in its formative stage, and help from all areas of the Linux community would be appreciated. Instructions for joining the LUIGUI

mailing list can be found on the Web at www.luigui.org/community/involve.html. Members of the project feel that with broad support throughout the Linux community, we can once and for all put to rest the last of the Great Myths about Linux, and finally enable it to make inroads among desktop computing users.

Randy Yarger (randy@yarger.tcimet.net) is the Systems Administrator and Chief Programmer for H-Net, Humanities and Social Sciences On-line (www.h-net.msu.edu). His sign is Virgo, his favorite color is blue, he wants to work on world peace and is currently practicing for the interview portion of the Miss America pageant.

Archive Index Issue Table of Contents

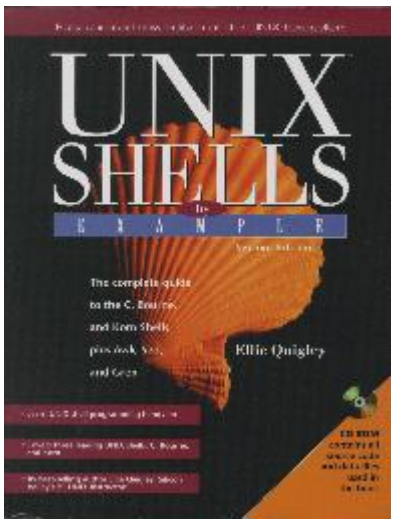
Advanced search

UNIX Shells by Example

Ben Crowder

Issue #71, March 2000

If you want to learn shell programming, this book is a must-have.



- Author: Ellie Quigley
- Publisher: Prentice Hall
- E-mail: corpsales@prenhall.com
- URL: http://www.phptr.com/
- Price: $49.95 US
- ISBN: 0-13460-866-6
- Reviewer: Ben Crowder

Unix Shells by Example truly fulfills the claim made in the title: over 630 examples are in the book, plus another 60 in the first appendix. If you want to learn shell programming, this book is a must-have.

The first chapter introduces the three major shells: the Bourne shell, the C shell and the Korn shell. The Bourne Again shell (bash), the one used by most Linux users, isn't specifically covered, but the information on the other three shells

usually applies. A history of each shell, plus the theory behind shells is given. Where the shell comes in on a UNIX system is described, as well as more general UNIX topics such as file attributes and permissions.

Chapter Two, “The UNIX Tool Box”, gives a very useful explanation of regular expressions which applies to much more than shell programming; for example, searching for text within Emacs or vim is built on basically the same principle. Over the next six chapters, what are considered to be three of the more important UNIX tools in regard to shell programming are covered.

Our friend **grep** appears in the third chapter—the meaning behind the name, how it works, and how to use it (with, of course, extensive examples). Chapter Four covers **sed**, the Streamlined Editor—again, replete with examples.

The next three chapters describe **awk**--the history, the input format, records and fields, patterns and actions, script files, comparison expressions, variables, pipes, loops and arrays, to name just a few topics. This section is quite extensive and very useful. I would have to agree with the author that these three are among the most important of all text processing utilities.

Each of the three major shells is covered in a separate chapter. Chapter Eight introduces the Bourne shell, discussing topics such as variables, metacharacters, file-name substitution, pipes, functions, arithmetic, looping and the built-in shell commands, in addition to many other subjects of interest.

The C shell appears in the next chapter. Each shell is given basically the same treatment: the first section of the chapter is devoted to the interactive shell (aliases, job control, the environment, redirection, etc.), while the second section covers programming in that shell (loops, flow control, arithmetic, etc.). Chapter Ten goes over the Korn shell in the same manner.

Appendix A lists useful utilities for shell programmers. These are stock UNIX programs, such as at, mv, passwd and so on. Quite a few examples are given (sixty, to be precise), as well as usage information and a short description. Appendix B provides a table for comparing the three shells. This is an invaluable resource for those who need to port shell scripts from one shell to another. For example, the table shows you how, in each shell, to assign the output of a command to a variable. In the C shell you use **set d='date'**, in the Bourne shell you use **d='date'**, and in the Korn shell you use **d=$(date)** or **d='date'**. Appendix C explains clearly and concisely how to use quoting correctly, especially for those tricky problems where you have multiple layers of quotes. The attached CD comes with all source code given in the book.

So, is the book worth your time and money? Yes, very much so. I found the examples to be particularly helpful, illustrating concepts far more clearly than prose could possibly do. If you're interested in shell programming, or even if you're just curious about shells in general, get over to your local bookstore and buy this book.

Ben Crowder is a young Linux aficionado living in Utah. In addition to fiddling with the insides of computers, Ben enjoys reading, writing and music. He can be reached at mlcrowd@enol.com.

Archive Index Issue Table of Contents

Advanced search