Robert E. Hassett. The Chapel Hill Linux Lab: A Case Study in the Use of Linux and Other Open Source Applications in the High School Setting. A Master's paper for the M.S. in L.S. degree. November, 2002. 18 pages. Advisor: Gregory B. Newby

This case study describes the design, installation, and maintenance of a small, Linux-based computer lab consisting of a console-server and four thin-client terminals at Chapel Hill High School in Chapel Hill, North Carolina. The project further involved the development of a 12-hour course in basic Linux system administration and its presentation to a self-selected group of high school students.

Despite some early difficulties with the high school's hardware, the lab was successfully installed and remained in place with little call for maintenance throughout the fall. The course was taught during these months to seven students. Although the project was not without setbacks and minor failures, each of these suggested not overall deficiency, but a way in which such a project could be more profitably deployed in a production, rather than an experimental, setting.

Headings:

Client server computing

Linux operating system

Microcomputers -- Equipment

Open source software

THE CHAPEL HILL LINUX LAB: A CASE STUDY IN THE USE OF LINUX AND OTHER OPEN SOURCE APPLICATIONS IN THE HIGH SCHOOL SETTING

by
Robert E. Hassett

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Library Science.

Chapel Hill, North Carolina

November, 2002

Approved by:

_____

Advisor

1. In The Beginning

This project grew out of an interest I developed during the summer and fall of 2001 in open source software, particularly Linux-based systems, and the possibilities accompanying the use of this technology in the educational setting, especially secondary school. Attracted initially to the open development model and to the communitarian outlook that supports it, I soon became aware, as I inquired more deeply, of claims made by open source backers that Linux could significantly reduce the cost of implementation, maintenance and upgrades of instructional technology, as well as provide increased security and reliability, all while engaging both students and teachers with an interface no less inviting or intuitive than the proprietary Microsoft or Apple systems they were perhaps already used to.

In December 2001 I contacted the Director of Instructional Technology and Media for the Chapel Hill-Carrboro City School System, Robert Stocking, to ask him whether he might be willing to host an experimental deployment of a small Linux-based computing lab in one of the schools within his jurisdiction. I had met Stocking in another context before and our meeting now was both amiable and productive. Like many school systems, Chapel Hill is very enthusiastic about thin-client solutions, which allow technology investment to be directed to a small number of very powerful servers, which drive terminals having almost no computing power of their own. In fact, these terminals may be operated without hard drives, providing little more than a user interface to

applications running on the server. The schools, Stocking told me, would be installing Citrix, a commercial terminal server application, beginning in the spring, and he was interested in checking out any additional options as well. Stocking suggested I contact Michelle Ammann, the technology specialist at Chapel Hill High School, about carrying out the project there.

During January 2002 I exchanged several emails with Stocking outlining and refining what the project would entail and what requirements I would have. He was very encouraging, but suggested I work primarily with Ammann, with whom I would ultimately be working during the execution of the project. In March I met with Tammy Bradley, who teaches a network administration class at Chapel Hill High, and outlined my intentions for the project, which at that time included only the installation and administration of the Linux cluster as a freely available option for all students. During subsequent meetings in March and again in May, these with both Bradley and Ammann, we determined that, in addition, I would offer some instruction in basic Linux/Unix system administration to a small number of interested students.

Throughout the time between the project's conception in December 2001 and its execution beginning in September 2002, I continued to work with Linux systems both at home and on campus, in order to familiarize myself with as much of a typical system's vicissitudes and anomalies as possible. With the assistance of Scott Adams, the SILS director of information technology, I set up a networked system in the SILS computing lab, first with Red Hat Linux, version 7.2, in order to work through general system administration tasks, and later with the K12LTSP software, which is based on Red Hat's Linux distribution, setting up a small (two terminal) cluster to test its functionality and to

understand the distinct configuration and added responsibilities of administrating this type of system. Again with some assistance from Adams, I borrowed a digital tape drive, which involved installing a SCSI controller and recompiling the kernel for SCSI support, in order to manage data backups. I installed and configured Samba, the Linux program that facilitates network file sharing with Windows-based systems. I did a great deal of reading, both in printed and online sources, in order to understand to the greatest extent possible, the inner workings and likely difficulties to be encountered in deploying Linux in a production environment. The O'Reilly & Associates book, *Running Linux* (Welsh, 1999), was an invaluable resource, as were the materials provided online by *The Linux Documentation Project* (TLDP) and by both the *K12LTSP* and the original *LTSP* projects.

Security also was a primary consideration. While I knew that the school system would have a fairly substantial firewall in place, I felt that the integrity of any data the students might choose to put on the system was a paramount concern. In addition, the likelihood of attacks through automated means, as well as considered attacks by unscrupulous students working within the firewall, had to be assumed. There was no reason to suppose that the school system's security policy would be enough to ensure the highest possible level of protection. I studied the “iptables” firewalling program, as well as making myself familiar with the "hardening script," Bastille Linux. I consulted extensively two books, *Practical Unix Security* (Garfinkel, 1991), which is somewhat out of date but which provides excellent background, and Bob Toxen's more recent *Real World Linux Security* (Toxen, 2001), which was invaluable.

2. Help Me, Somebody

In March I began trying to find a donor willing to contribute a reasonably powerful system to be used for the server. Perhaps naively I had assumed there must be some business or even some individual in the community who had a system that was out of date by the standards of high technology but which still might possess enough resources to drive a small cluster. Rather than deal with disposing of or selling such a system, I thought, this beneficent group or individual might just as well donate it to a school in exchange for a tax deduction.

Towards this end I posted notices to Triangle Internetworkers, an Internet mailing list populated by Triangle-area computing professionals representing a wide range of interests and backgrounds. I posted a similar note to the Triangle Linux Users Group (TriLUG) list, the membership of which includes, not only many of the same technology professionals as Internetworkers, but Linux enthusiasts of all stripes, including members of the various university communities, hobbyists, and developers, some of them employees of Red Hat Inc., with an interest in promoting a wider use of Linux-based systems and applications. This seemed a reasonably broad net to cast. Finally, I tried to cultivate some relationship directly with Red Hat, first through a series of emails to Jeff Johnson, a contact gleaned from the Web page of Simple End User Linux (SEUL), and then through a letter sent directly to the company's CEO, Matthew Szulik. Szulik had made a number of public announcements about Red Hat's commitment both to pushing the use of Linux in education and to investment in the North Carolina and specifically the Triangle community. None of my attempts to reach anyone at Red Hat was ever successful.

It should be noted that I did receive more than a dozen messages from various quarters offering technical assistance with setup, installation, or support. Although I thanked each of these people individually and suggested I would possibly be in touch come fall, I did not find the need to pursue any of the offers.

There was one promising lead on the hardware front. A subscriber to Internetworkers, Jason Tower, a technician with a VAR firm in Raleigh, Computer Service Partners (CSP), e-mailed me back suggesting that his firm was interested in breaking into the Linux market and might have a server class machine available for my project. I phoned him and we discussed the particulars. He promised someone from the company would contact me. When I got a call the following day from a sales consultant, we arranged a meeting. Already skeptical beforehand that the meeting was with a sales representative, I did some research and discovered that one of the primary products sold and supported by CSP was none other than Citrix. My suspicions were soon confirmed. While the representative was very cordial and encouraging, she offered little hope that either one of two machines the firm had slated for disposal would be donated to my project. She, too, had done some research before our meeting and discovered that the Chapel Hill schools had already entered into a Citrix contract with a competitor.

In the mean time, I had remained in contact with Michelle Ammann, the technology specialist with Chapel Hill High School. Through Ammann, I met Dave Scott, the network specialist for the school system. Not long after my meeting with CSP, I received a message from Scott that he did have a system I could use for the project, a somewhat outdated Pentium II, 300 MHz server with 256MB of RAM and a 2GB/4GB DDS tape drive, as well as three 9GB SCSI hard drives and a RAID controller. Its Novell

license would run out in June, he said, when it would be wiped clean, and I could pick it up any time after that. While the CSP representative had been kind enough to have provided me with the name of someone at IBM who, she said, oversees the company's education projects, and encouraged me to mention she had recommended I contact him, I felt now that using the school's machine might after all be the best course to follow. Part of what I had hoped to demonstrate in the first place was the feasibility of running such a project on existing school system hardware, that is, at no cost, potential or otherwise, to the school. It seemed to me, and does so still, that soliciting a high- or even medium-grade system from private industry to anchor the project would, at the very least, dilute any positive results that might eventually be returned. That is, what model might the project be for other schools considering Linux if we had received the sort of preferential treatment another school might not hope for?

During July I took the system home with me. It would be an understatement to say that I experienced difficulties with this machine from the first time I booted it. The BIOS was password-protected and Dave Scott conceded he had no idea what the password might be. Some research turned up several backdoor passwords used by Phoenix BIOS technicians, but none of them worked. This proved to be a waste of time, however, as in the end I simply pulled the battery out of the motherboard for a few hours, clearing the password memory and resetting the BIOS without authentication, which paved the way for booting from the CD drive for installation. After a brief flirtation with the RAID controller, I decided to go without it. For the likely volume of traffic on this machine, it seemed to me more trouble than it was worth. I would be better off with 27 GB of storage than with 9GB of RAID storage. This would actually be 18GB, as Scott

had informed me in advance that one of the hard drives was inoperable, though he admitted he didn't know which. After some trial and error, I determined which one was most likely fried and pulled it out.

One could follow this branch of the story a good deal further out into the cold, but the short of it is that I did manage to get the system installed and worked with it on a limited basis for a few weeks before a spate of seemingly random segmentation errors, and other glitches suggested bad memory. I managed to replace the memory with some borrowed from the high school, but when read errors on the CD drive scotched a demo install with students during the first scheduled class, I decided I had no choice but to abandon this system as a hopeless black hole of broken hopes and unanswered prayers. I posted a note to the K12LTSP list detailing my CD drive dilemma and a technician with a school district in Oregon wrote a script to allow for a network installation using archived ISO images. However, I was wary of going forward with the machine that had caused me so much difficulty. In addition, I was wary of hogging the kind of bandwidth necessary for a network installation during the middle of the school day. Having made this decision, though, with limited physical resources and no money, I was now in a bit of a bind.

I am eternally grateful to Scott Adams for extricating me from this pinch. Adams lent me an unused Dell PII, 266 MHz, with 128 MB RAM, 4GB and 8GB IDE hard drives, and two Intel EtherExpress Pro 100 NICs. The system would prove to be just shy of the task in many ways, but it was worlds better than the stark nothing I had to work with before that point. Along with a switch, cable, monitors, keyboards, and 4 very stripped-down terminal boxes, all supplied by the school, we finally had our five-terminal LTSP cluster, just in time.

3. Doing It

In August, Bradley, the networking instructor, identified seven students from her own classes who agreed to give up their lunch period, once per week, to attend the planned class. All of them were males, all with a high degree of interest in computing technology and most with some considerable skill in administrating Windows or Macintosh systems. One student had installed Linux on a home system and had some experience with installing software and basic administration tasks. An attempt was made to inspire some interest among female students, but this was ultimately a failure. Bradley herself initially expressed an interest in attending the classes; the time period is her lunch break as well. However, I had from the very beginning detected a note of skepticism in her approach. She seemed suspicious that the software was available at no cost and without licensing. And her questions about the open source development model suggested she perceived the entire Linux project as a hobbyist's plaything, clever perhaps, but fickle and easily broken. Still, she did attend two of the classes and encouraged the students' interest.

Ammann, the technology specialist, also voiced an interest in joining the class. She discussed extreme frustration at having to deal seemingly endlessly with Microsoft's restrictive and expensive license agreements, particularly for Office, which the school uses extensively, and hinted that she might be interested in trying out Linux on a broader basis. I had little hope that she would be able to spend much time in class, however. With the Citrix roll-out to add to her already brutal schedule, the amount of time she could

devote to non-required training was nil. One day, for example, when I had to track her down for the key to the computer lab, a blade failure on a Citrix server had left the entire school without access to any network applications, including Office for the students and state-mandated attendance reports, which are Web-based, for the teachers. The week before, she said, there had been a fire in the server room.

It would be a grave oversight not to mention that, even more helpful than Ammann was her assistant, Luca Ruzzeddu, who already had an interest in Linux, who had an experimental Mandrake 8.0 system running in his workroom, and whose early support I believe to be one of the strongest reasons this project saw execution. Throughout our early meetings, whenever Ammann expressed uncertainty about issues such as space or scheduling, it was always Ruzzeddu who assured her that it could be and should be done. Ruzzeddu, too, expressed an interest in attending some of the classes, but his schedule proved too demanding as well. However, it was Ruzzeddu who assisted with the initial setup, marshalling all the hardware, helping to transport it to the lab, and then working with me to establish the network setup.

The relative slowness of the system proved to be a liability from the very start. The K12LTSP documentation suggests a minimum of 50MB of RAM per terminal. I had presumed that the 256MB in the original server would be sufficient to drive our five terminals. When we were reduced to 128MB, the maximum memory for this particular machine, I despaired. And not in vain. A thread I followed later on the K12LTSP mailing list included a suggestion by one of the principals of the K12LTSP project that the minimum really should be thought of as 256MB plus at least 50MB per terminal. What this meant was that, while I was demonstrating anything at all, I had to ask the students

not to work on the system because the drain on resources made my illustrations run ludicrously slow. I early on found myself frequently making apologies and assuring my students that this would happen much more quickly on a more powerful system. Of course, being relatively young and spoiled by the current state of technology, they seemed constantly amused at the folly of using such an outdated box for anything but a doorstop anyway.

Indeed, at one point, after a class had dispersed, I was alone in the lab, finishing up some work, when another student with whom I was not acquainted, wandered in and sat down at one of the terminals. The room is full of iMacs, but the teachers remove the mice when they're not in use for a supervised class. A Mozilla window happened to be open on the Linux box, so she sat down, saying nothing to me, and began browsing. I watched her in silence, eager to see what her reaction might be. After a few minutes of surfing, she turned to me and asked if something was wrong. Somewhat at a loss, I told her it wasn't Windows she was using, but Linux. "Oh," she said, as though this had cleared the skies for her, "it's, like, really slow." I was crestfallen. I had seen this as an opportunity to show a student who might otherwise have no exposure to Linux that it could function just like whatever she was used to. Instead she had been left with the impression that Linux only further aggravates the trickle she was already accustomed to from the school's limited-bandwidth network connection.

Despite these shortcomings, however, the students, who, with the occasional exception for illness or detention, dutifully attended every class, were clearly engaged with the material and committed to learning more. Where things really took off was when I began to discuss the “themes” which allow users a high degree of customization of their

individual desktops. As soon as I showed the students the ease with which the desktop could be reconfigured and demonstrated the huge and constantly growing number and variety of themes available online from http://gtk.themes.org, I suddenly had their attention. It seems obvious now in retrospect, but throughout the course, it was the bells and whistles that drew their attention. The port scanner, Nmap, which finds open ports on networked computers, was a big hit. Even where the results of a particular scan were relatively uninteresting, they loved that they could do this. I scanned the school's main Web server and they practically giggled. The network monitor, Ethereal, also was popular. As networking students, they were intrigued by this graphical tool that could return so much information about what was out there on the network and present it in an easily digestible fashion. I tried always to emphasize the importance of knowing how to work from the command line ─ we compiled a number of programs from source code ─ and they seemed to get it and appreciate it. But it was always the graphical utilities that elicited the most palpable responses. Even explaining to them about psDoom, a combination video game and system management utility that presents a three-dimensional landscape populated by the computer's running processes and allows the administrator to terminate them with weaponry, brought howls of excitement.

For the course I had tried to develop a syllabus that covered the basics ─ installation of software (compiling from source and using the RPM installer), security, configuration of common applications such as the Apache web server and Sendmail mail transfer agent, the Unix file system, process management ─ as well as touched on some of the more end-user-friendly aspects, such as the GNOME and KDE desktops, the Mozilla web browser, and a bit on games. The syllabus was posted online and updated

from time to time as needed. The syllabus also included a links to some of the more useful Linux mailing list archives and to a collection, which I maintain, of other Linux-related links. I encouraged the students to contact me either by telephone or by email if they had any questions or suggestions, though none of them did so. It took a few weeks for them to warm up to me, but by the middle of the course I think we had developed a good rapport, at least within the class. All of them felt comfortable interrupting me with comments or questions and they joked freely.

4. What Does It All Mean?

Despite, or perhaps in some sense because of, a number of obstacles encountered along the way, this project was highly instructive. In brief, what was demonstrated was that it is eminently possible to make extensive use of Linux-based, open source systems in a secondary school environment. Further, it was shown that technical support for such systems can be provided by an existing faculty member with little formal training and not much more experience than that provided by a strong interest and a willingness to experiment. The seven students to whom the system was made available expressed enthusiasm over the desktop functionality provided by such Microsoft-compatible applications as Open Office and, as evidenced by records of system usage, without prompting engaged in some more complex administration tasks. The extensive capability for customization was another strong selling point.

Further, the obstacles encountered suggested several things about the way it might be done differently to increase success. Linux is obviously not a cure-all. It may be true, in general, that a smaller system profile is required than for Windows-based systems. But

it is not practical to try to run a terminal farm on a thoroughly outdated system. If a relatively new, server-class system were not available, some investment would be required in fairly high-end hardware to drive the cluster efficiently. Still, no software costs were incurred beyond the nominal cost of a few blank CDs. And no licenses needed to be secured or maintained.

Really to get a project like this going over the long term would require a commitment on behalf of someone on staff, either a technology specialist or some teacher or librarian, who, with the full cooperation of the technology specialist, was willing to oversee it, someone who works on campus and is available and committed to ensuring success. Had I been employed at the school and been able to spend more time there troubleshooting, particularly in the weeks before instruction began, it is likely I would have been able to get the initial server running by swapping out a few components, which were available from Ammann and Ruzzeddu. Thus, the cost of deployment would effectively have been zero and the problems with slowness would at least have been greatly reduced, if not eliminated altogether.

The IP address assigned to our server was a private network address in the 10.15.x.x range. This was, of course, great for system security, shielding it behind a firewall from the greater Internet, but made for difficulty for the students, eliminating the possibility of remote administration as well as denying the opportunity to serve Web pages off campus. This was a source of great frustration for me during the early weeks of the course. I had made certain that the server ran a secure shell daemon (sshd) and had pointed the students to several freely available ssh clients available on the Internet, which would have allowed them to experiment with administration tasks, particularly with

installing software, from home. Because of our position on the network, I was forced to abandon an assessment task which would have involved requiring each of the students to obtain and install some program of their own choosing. Their schedules when at school left little time for heavy experimentation, which I believe strongly to be the only way really to learn how software works.

After exchanging several emails with Dave Scott and with one of his colleagues, I finally, after several weeks, got him to allow us to use port forwarding to an external IP address. But this turned out to be ineffective. Because of the understandably low priority assigned to this project by the school and the persistent difficulty in maintaining prompt and efficient communication, as well as because of the narrowing time frame, I did not pursue the issue further, relying instead for assessment on class participation and on personal observation of students' work during class time. While disappointed, the students, all having some experience with network administration through Bradley's class, understood the issue and accepted it with equanimity.

It was a disappointment, too, that other students were not able to participate. I had hoped to gauge to some degree the attitudes of students not already prejudiced in favor of computer technology. At one point I was asked by Ammann if it would be possible to provide access to the Linux terminals for a large class which required more Web browsers than the Mac lab provided. I happily obliged, creating a guest account and providing instructions for logging in and starting up Mozilla. But, for reasons never fully explained to me, the Linux boxes were never used.

We did demonstrate reliability. The system had to be rebooted only once during the twelve-week period and this was due to a student accidentally pulling the power cord

while detaching the server from a data projector. Granted that at no time was the system placed under heavy load, I still believe it significant that we were able to keep a four-terminal cluster running without having to consult outside for technical assistance, other than to pose questions on a free listserv, throughout this time.

In conclusion, the weaknesses of this case study, taken together, may be accounted simply as valuable instruction in how a similar project might be successfully carried out in a production environment. Nothing in the experience of this project suggests that it could not be implemented successfully on a much broader scale by providing perfectly realistic investments in hardware and time.

References

Garfinkel, Simpson, and Spafford, Gene (1991). *Practical Unix Security*. Sebastapol, Calif.: O'Reilly & Associates. 1991.

K12 Linux Terminal Server Project (K12LTSP). Retrieved November 27, 2002, from http://k12ltsp.org.

Linux Terminal Server Project (LTSP). Retrieved November 27, 2002, from http://ltsp.org

Simple End User Linux (SEUL, n.d.). Retrieved November 27, 2002, from http://www.seul.org/

The Linux Documentation Project (TLDP). Retrieved November 27, 2002, from http://tldp.org

Toxen, Bob (2001). *Real World Linux Security: Intrusion Prevention, Detection & Recovery*. Upper Saddle River, N.J.: Prentice Hall PTR. 2001.

Triangle Internetworkers (n.d.). Retrieved November 27, 2002, from http://lists.ibiblio.org/mailman/listinfo/internetworkers

Triangle Linux Users Group (TriLUG, n.d.). Retrieved November 27, 2002, from http://trilug.org/

Welsh, Matt, Kalle, Matthias Dalheimer, and Lar Kaufman (1999). Running Linux. Sebastapol, Calif.: O'Reilly & Associates. 1999.