Quick Lookup Laptops in the Library: Leveraging Linux with a SLAX LiveCD

This article discusses the experiences over the past year at Laurentian University in deploying a Linux live CD solution to create customized kiosk browsers on 'quick lookup' laptops. This provides security, reliability, ease of use, and improved response time compared to other options. We explain why we chose this solution, reveal techniques we used to achieve some of our goals, and reflect on possible future iterations of this project.

By Dan Scott and Kevin Beswick

Background

Laurentian University is a bilingual university with over 9,000 students located in Sudbury, Ontario. Complementing the Learning Commons located within its walls, the J.N. Desmarais Library provides two additional classes of computers for its patrons: a set of 16 research workstations at traditional tables located near the reference desk for easy access to reference assistance; and a set of 20 laptops mounted at standing height throughout the stacks to provide quick lookup functions including call number retrieval for known items, reserve item access, and account access. The laptops had been purchased in 2003 in conjunction with a new library system to replace the VAX terminals used with the previous library system. As the old VAX terminals did not use CAT5 cable, the decision was made to rely on the laptops' wireless connectivity rather than running 20 new CAT5 connections that would have cost approximately $10,000 CDN.

Problems

The laptops handled the job for years, but by 2006 some problems were beginning to surface. The first problem was that the actual user experience made a mockery of the term “quick lookup laptops”, because the hardware was overly taxed by the configuration of the operating system and software that were installed on it. Although the laptops had only 256 MB of RAM, they were running Windows XP, an antivirus program, Windows Defender anti-spyware, and a GroupWise client. Additional programs that had been added to the laptops since their initial deployment, such as Microsoft Office and SciFinder Scholar, were virtually unusable due to the taxing memory requirements that resulted in continual paging to virtual memory on disk.

A second problem was that the laptop images required a significant investment of ongoing maintenance effort. At first, the new Systems Librarian tried to maintain the individual laptop images where they were mounted: applying manual upgrades where automated upgrades were not available or were not working; tweaking operating system security settings and profile settings; and running hard drive checks on the aging laptop hard drives. Despite limiting users’ privileges, it was not uncommon to find that a game had been installed on the laptop by simply unzipping a set of files onto the desktop. The manual approach to administration was quickly abandoned in favour of a reimaging process relying on the use of the open-source Partimage imaging utility.

A third problem was that the laptop images did little to protect the privacy of the individual user. By default, the laptops were logged into a local Windows XP account. If a user saved files to the hard drive, subsequent users could read those files. If browser sessions were left open (which was typical, because starting a new browser took so long), subsequent users could peruse the browser history and access cached pages. Users had the ability to log into their own campus accounts via these laptops, but when they chose this option they also frequently forgot to log out after finding the resource they needed.

For historical reasons, the maintenance of these laptops was the direct responsibility of the Library rather than the campus information technology (IT) department. The Library has one Systems Librarian responsible for all Library hardware and software, including the ILS, and until the fall of 2007 no additional resources were available to assist the Systems Librarian, so the solution to these problems had to be relatively inexpensive in terms of human resources.

Proposal

Over the course of the summer, the Systems Librarian investigated several possible means of alleviating the problems experienced with the quick lookup laptops. Purchasing more RAM was considered and quickly rejected, as the age of the
laptops meant that the capacity was limited to a total of 512 MB of RAM and the cost would be prohibitive for nearly obsolete technology. Continuing the existing approach of reimaging the laptops with new Windows XP images was ruled out as an option because the performance problems suffered by the patrons would not be resolved and, at a cost of approximately 20 hours to roll out a new image, there were more pressing demands for the Systems Librarian’s time. Asking the campus IT department to assume maintenance of the laptop images was rejected as the IT department was already months behind in two other Library projects on behalf of the Library and also faced a shortage of skilled human resources. Therefore, in the fall of 2006, the Systems Librarian brought forward the following proposal to the Library: we would deliver a more secure quick lookup laptop with better performance for our users under the existing memory constraints. To achieve this goal, we would replace Windows XP with a simple Linux desktop, with an estimated development and deployment investment of 30 hours. The Systems Librarian was given the go-ahead and challenged to deploy the initial solution by the end of 2006.

Implementation

Phase I: Evaluating alternative image delivery approaches

The laptops had originally been deployed with the batteries and CD-ROM drives removed. When we retrieved the drives from storage, we discovered that over the years a number of the CD-ROM drives had been lost (“inventory shrinkage”), so we were forced to pursue two approaches to loading Linux on the laptops. We looked upon this as a good opportunity to compare approaches and learn from the experience.

For those laptops that had CD-ROM drives, we opted to take a Linux live CD and customize it for deployment as a kiosk browser. A Linux Live CD is a Linux distribution that runs directly from a CD-ROM (or other media) rather than having to be installed on a hard drive. After investigating a number of candidate live CDs, we decided on the SLAX Popcorn Edition for the ease with which it could be customized, its minimalist Xfce desktop, and its support for our laptop hardware. We found an excellent tutorial for customizing SLAX live CDs that we were able to rely heavily on for our initial customization efforts. Recognizing that we would be going through a number of iterations of the image, we burned the images on CD-RW discs to be environmentally friendly.

For those laptops without CD-ROM drives, we built a “Linux kiosk” image based on Ubuntu 6.10. We installed additional Xfce packages so that we could reproduce a similar desktop look and feel as the live CD images. These images were installed directly on the laptop hard drives using the now-familiar Partimage.

<table>
<thead>
<tr>
<th>Quality</th>
<th>SLAX Live CD</th>
<th>Ubuntu installed image</th>
<th>Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>As only a minimal number of processes run in a live CD, most system resources are devoted to the display and the browser session, resulting in excellent performance for a five-year old laptop.</td>
<td>Services like updatedb were disabled, so despite a few additional processes over and above what was required for the live CD, the performance was largely comparable to the live CD.</td>
<td>Equal</td>
</tr>
<tr>
<td>Privacy</td>
<td>In a worst-case scenario, rebooting the laptop means that all data would be wiped clean. Logging out of a session shut down the laptop, forcing a wipe of all data.</td>
<td>Files written to disk survive reboots, posing a threat to privacy. We developed a shell script that used rsync with the –delete option to restore the user’s home directory to a known set of files on login as part of the profile.</td>
<td>Edge goes to SLAX</td>
</tr>
<tr>
<td>Security</td>
<td>When package upgrades are available, they can simply be copied to to the live CD and a new set of CD-RWs burned and deployed. However, while popular packages like Firefox were upgraded quickly, many core system packages did not receive upgrades.</td>
<td>Once the image was installed, Ubuntu’s apt-get update and apt-get upgrade can be run as a cron job to deliver security updates for all packages.</td>
<td>Ubuntu</td>
</tr>
<tr>
<td>Speed of deployment</td>
<td>Rolling out an updated image takes approximately 5 minutes to burn each disc at 12X write speed. We can burn a stack of discs in less than two hours, then simply swap the discs in the laptops out on the floor and reboot to deploy the image.</td>
<td>Rolling out an updated image took approximately 1 hour per laptop, and as we rely on wireless for connectivity, we had to bring each laptop physically into the office for re-imaging rather than being able to rely on BOOTP to start a remote re-imaging session over slow wireless.</td>
<td>SLAX</td>
</tr>
<tr>
<td>Quality</td>
<td>SLAX Live CD</td>
<td>Ubuntu installed image</td>
<td>Advantage</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Robustness</td>
<td>In the event of a hard reboot (for example, the power cord being unplugged so that a student can provide power to their own laptop), the system always recovers gracefully.</td>
<td>In the event of a hard reboot, the system runs a filesystem check. We found that the ext3 filesystem seemed prone to errors that eventually required human intervention (possibly due to the age of the hard drives). In this respect, the old Windows XP images had been much more robust than the Ubuntu images.</td>
<td>SLAX</td>
</tr>
</tbody>
</table>

We were able to complete the work required to build both images and deploy the images onto the 20 quick look up laptops in approximately 40 hours in December, 2006. This was slightly more than the estimated time, but was considered reasonable given the unexpected requirement to build two images.

**Phase II: Evaluation**

During January, 2007, the Systems Librarian evaluated the success of the newly deployed Linux images on the laptops by soliciting feedback in the Library Newsletter that was emailed to all university students, staff, and faculty; observing users directly; and approaching users to help those having difficulty. The most common complaint with the live CD images was that when a person shut down a session, it shut down the computer rather than automatically starting a new session. The most common complaint with the installed images was that the system was unusable due to the filesystem errors preventing a reboot. A secondary complaint was that many users accustomed to the Internet Explorer icon could not figure out how to start a browser session despite the desktop consisting of only two widgets: a clock, and a Firefox icon. No users complained about the loss of the additional functionality that had been offered in the Windows XP images, supporting our belief that most users did not have the patience to try to use those applications in such a resource-constrained environment.

Due to the instability of the installed images, we opted to purchase additional CD-ROM drives for the remaining laptops and to develop an improved live CD that better met users' requirements. The improvements consisted of:

- a cron job that automatically starts a new browser session if it detects that no browser session is current running, thereby avoiding the need for a user to figure out how to start a browser session
- changing the system “shutdown” command to a “reboot” command so that if a user does inadvertently exit a session, within a few minutes the laptop will automatically be ready for the next user rather than requiring the user to power on the system and wait for the live CD to load
- a customized Firefox profile that sets the default home page to the Library Web site, includes a LibX toolbar, paranoid privacy settings so that all cookies, cached files, memorized passwords, and the like are deleted when the browser session ends, and a browser cache reduced to 1MB to avoid using too much system memory
- adding an NTP module to ensure that the desktop clock was accurate

These modifications were trivially easy to accomplish with SLAX. By placing files in the /rootcopy/ directory of the SLAX image, or any subdirectory mirroring the normal file system hierarchy, the distribution automatically copies those files over the corresponding file (if any) in the real file system hierarchy at boot time.

**Customized /etc/X11/xinit/xinitrc for our preferred desktop**

Most of the customization reflects a paring-down of the defaults to only include those widgets that we require; to set up our Firefox browser profile and launch the browser; and to reboot the system rather than shut down so that it is ready for the next user. We extract the Firefox profile from a tarball rather than copying each file from rootcopy for the sake of simplicity.

```bash
# Set up the desktop to our liking
cd /root
# Restores Xfce desktop menu & configuration
tar xzf /etc/xfce4_config.tar.gz
# this is only necessary when running w/o xfce4-session
xsetroot -solid black -cursor_name watch
# or use old-fashioned startup script otherwise
xfce-mcs-manager
xfwm4 --daemon
xftaskbar4 &
xfdesktop &
```
# Restore Firefox profile settings
 tar xzf /etc/firefox_profile.tar.gz
# Install LibX extension
 /usr/bin/firefox --install-global-extension /etc/libx-lul.xpi
# Start Firefox
 /usr/bin/firefox -fullscreen http://www.laurentian.ca/Laurentian/Home/Departments/Library/Library+Home.htm &
# Customizing /etc/rc.d/rc.slax to reboot rather than shutdown
 if [ "$AUTOCMD" != "" ]; then
    echo "starting autoexec command: su --login -c "$AUTOCMD""
    su --login -c "$AUTOCMD"
    # telinit 0
    # we want to reboot (init level 6) rather than shutdown (init level 0)
    telinit 6
 fi

Perl script in /root/check_browser for starting a new Firefox session if none exists
In retrospect, Perl is probably overkill for this script - but it was easy to write and the system demands do not seem to be too onerous for a script that runs every minute.

```perl
#!/bin/perl
use strict;
use warnings;
my @procs = `ps wax`;<
my $ffox = 0;
foreach my $process (@procs) {
    next unless $process =~ m/firefox/;<
    $ffox = 1;
}
if ($ffox == 0) {
    system('firefox &');
}
```

Cron job in /var/spool/cron/crontabs/root
When the SLAX rootcopy process overwrites system files, you have to ensure that the permissions and ownership of the files in the /rootcopy/ directory match your destination. For example, the following cron job file must be owned by root.

```
# Check to see if firefox is running every 60 seconds
* * * * * DISPLAY=:0.0 /usr/bin/perl /root/check_browser 1> /dev/null 2> /dev/null
```

Screen shot of the Laurentian Library Linux Live CD in action
Phase III: Refine offering for 2007-2008 academic year

As Laurentian University is an officially bilingual (English and French) institution, users need to be able to search for resources in both languages. The initial deployment of the laptops had set the keyboards to a plain English (United States) layout and provided no means of entering French characters. This problem had to be addressed for the 2007-2008 academic year.

We began by investigating methods that could be used to switch the keyboard layout in SLAX. We had the option of setting the language at boot time, but this method was deemed unacceptable because the user would have to wait while the laptop was rebooted to use the different keyboard layout - if they were even capable of figuring out that the language could be switched by rebooting. We subsequently discovered the standard X command - setxkbmap - which changes the keyboard to a selected layout, and recognized that this could be the basis for our solution.

For our interface, we wanted to use affordances that would be familiar to the users. The research workstations in the Library, which run Windows XP, use a language switching icon in the taskbar to toggle the keyboard between English and French layouts. The icon is stateful and shows the language that is currently being used (EN for English, FR for French). When the icon is clicked, the language switches to the next configured language. We wanted to offer a similar, familiar mechanism on the laptops to our users.

Fortunately, as part of its standard desktop icons, Xfce offers a launcher widget called a ‘Two-state Launcher’ with an ‘on’ and an ‘off’ state. After consulting some documentation, and the forums found on the SLAX website, we determined that the launcher is controlled by a shell script that returns a 1 for the ‘on’ state, and 0 for ‘off’. We wrote a script that, when the icon is clicked, checks which keyboard layout is currently being used, switches the keyboard layout to the opposite language using the setxkbmap command, and then returns the opposite state depending on which language is currently in use. This script, paired up with the double state launcher, provided a perfect language switching function for the laptops.

Script for toggling keyboard layouts - keyboardswitch.sh:

```bash
#!/bin/bash
#Shell script to toggle between keyboard languages in
# Xfce using the two-state launcher
state="/usr/X11R6/bin/setxkbmap -v | grep -o \+us"
if [ "$1" = "0" ];
  then
    /usr/X11R6/bin/setxkbmap -layout ca; echo "1"
elif [ "$1" = "1" ];
  then
    /usr/X11R6/bin/setxkbmap -layout us; echo "0"
else
  if [ "$state" = "+us" ];
    then echo "0"
  else echo "1"
fi
fi
```

We made a pair of EN and FR icons in .PNG format using the GIMP and set these icons to appear on the two-state launcher widget. The script and buttons can easily be modified to offer any keyboard layout. One simply needs to change the keyboard layouts that the script passes as arguments to the setxkbmap command, and to create corresponding 100 pixel square icons to represent the language names.

The other modification we made to the lookup laptops in the Fall of 2007 was to associate MIME types to their respective SLAX applications to extend the use of the laptops slightly beyond quick catalogue searches. For example, we observed that users often search for an article in the Library databases and want to preview it quickly to see if it meets their needs. We added the XPDF application and associated the application/x-pdf MIME type with XPDF so that users can view the PDF on the lookup laptops. Similarly, we associated Microsoft Word documents with the AbiWord application. Although we initially attempted to set these preferences in the Firefox global preferences files, we found that we instead had to edit the settings in the Firefox browser, then create a tarball of the Firefox profile to include in the SLAX rootcopy directory to be extracted at boot time.

Reflections
The use of a customized Linux Live CD for the quick lookup laptops at Laurentian University has been a success. We have achieved our goals of increased performance, security, privacy, and robustness over the previous solution without compromising the ease of use, despite changing from Windows to Linux. The project has been both a financial success, in that we have been able to defer the purchase of new laptops for at least the 2007-2008 academic year, and an environmental success, as we avoided having to dispose prematurely of the old laptops. However, there are some risks and concerns that we will have to address as the project continues to mature.

One risk in depending heavily on a small Linux distribution is that it can stop development or disappear relatively quickly. In November, 2007, the project home for SLAX 5 was replaced by a blog announcing that SLAX 5 was dead and discussing the slow progress towards the next versions of SLAX. Fortunately, a SLAX 5 site mirror continues to be available and to publish updated modules, so new security vulnerabilities found in packages like Firefox can continue to be addressed until the new versions of SLAX are released.

For maximum hardware compatibility, SLAX live CD runs X in a generic VGA framebuffer, rather than using the native driver for the video card. The downside to this approach is that power management for the display is not supported through the standard “xset dpms 300 600 900″ command, so our laptops currently use more power than is required. Initial attempts to address this problem with SLAX 5 have been unsuccessful, but we are optimistic that hardware support in future versions of SLAX, or in alternative live CD distributions, will improve this situation.

As our laptops physically deteriorate, we will purchase newer machines that, even at a third of the cost of the original laptops, are more powerful than the research workstations that we deployed just a year ago. We will have to decide whether to maintain the minimalist profile of the laptops to satisfy a few specific use cases despite their increased power, or whether we shall expand the functionality of the hardware, thus increasing their potential utility but also increasing the potential confusion of the interface. We may opt to purchase more conventional desktop workstations with separate monitors and keyboards; while laptops in 2003 offered a net savings in purchase price over a desktop + monitor + wireless card, the maintenance cost is substantially higher once the hardware is past warranty. For example, repairing a single missing key on our current laptops requires a replacement of the entire keyboard, at a cost of approximately $150. The challenge to deploying conventional desktops becomes a problem of physical security, as the CD drive would be exposed and malicious users could conceivably replace the standard CD image with their own customized CD.

If we add wired Ethernet ports, we could entertain the possibility of running thin clients that connect to a Linux terminal server rather than running independent copies of the operating system. However, as the estimated cost for 20 ports remains at $10,000, this possibility seems unlikely. It is not clear that there would be a sufficient return on investment in learning and deploying a new thin client system.

About the Authors

Dan Scott. Dan has been the Systems Librarian for Laurentian University since 2006, and has been a library geek even longer than he has been an open source geek. You can reach Dan via email at dscott@laurentian.ca and follow his blog at http://coffeecode.net.

Kevin Beswick. Kevin is a third year computer science student at Laurentian University. He enjoys computer programming and playing various musical instruments. You can reach Kevin via email at kx_beswick@laurentian.ca.