Short Note

Linux at SEP

Matthias Schwab and Paul Hargrove[1]

keywords: *not available*

INTRODUCTION

SEP continuously balances its commitment to geophysical research with its maintenance and improvement of its computing environment. We suggest that SEP could save a significant amount of time currently expended on the computing environment by simplifying its system, in particular, by reducing the number of supported computer platforms. We suggest a conceptually different computer environment in which SEP researcher would use a Linux workstation for standard UNIX services and a few special purpose computers for specific tasks such as huge computations. We prefer Linux because of its software quality, its portability, and free availability. Additionally, the PC hardware is not limited to a single vendor's operating system.

First we discuss features that demonstrate the capability of Linux-PCs to serve as office and home workstations at SEP. Next we point out the ease with which we ported the SEP environment to a Linux-PC. Finally, we point out the problems we see with SEP's current computer environment and suggest how to improve it.

WHY LINUX-PCS?

What is Linux?

Linux is a freely-distributed UNIX system (OS, utilities, and applications) for PCs. Linux is a straightforward POSIX version of UNIX with many utilities drawn from the GNU and BSD systems. The file system is highly configurable and seems to be closer to System V rather than BSD. Most commands, however, look like BSD commands. The feel of the user environment is similar to a Sun under SunOS. Additionally, Linux supports a wealth of standard PC and UNIX hardware.

[1] **email:** matt@sep.stanford.edu, hargrove@sccm.stanford.edu

The Linux and GNU tools are documented by man pages, GNU "info" hypertext manuals, and HTML-format "HOWTO" pages, and supported (in many cases by their authors) on public newsgroups. The help that we received through newsgroups during our port to Linux was prompt and accurate. Finally, several fine books (Welsh and Kaufman, 1996) on Linux are available in a good technical bookstore.

While our experience and this note are limited to Linux on PC hardware, Linux is cross-platform. It is available now for SPARC and DEC Alpha machines and ports are under way to PowerMacs and MIPS machines.

Software quality

The GNU utilities and the Linux operating system are of high software quality. The free distribution of the software and the accessibility of the source code seems to motivate many users to contribute bug reports, patches, and enhancements. Also any GNU or Linux package has a personal caretaker, who users contact when they experience problems or when they have discovered a bug. This mode of operation knows no guarantees, but has, nevertheless, been highly successful.

A team of researchers at the University of Wisconsin compared various UNIX flavors by feeding random input into various tools such as `diff`, `ftp`, `mail`, etc. In the original study in 1990, as well as in a repeat study in 1995, the GNU and Linux software proved superior to its commercial competitors. The results of both studies were published in a curiously amusing paper (Miller et al., 1995).

Portability

SEP distributes its software to sponsors and colleagues who want to use it on their local computers. Thus, software portability is an important issue for SEP. In the simplified environment we advocate, systematic compilation checks and occasional ports to non-SEP computers should help to ensure reasonable software portability. However, we would loose the safeguard currently offered by SEP's daily use of its software on multiple platforms. Therefore, it is important that we choose a platform that encourages software portability. We believe Linux meets this requirement.

If SEP software works smoothly on a Linux system, then it will work smoothly on nearly any GNU environment. Since GNU is easily portable to most common platforms, the SEP software becomes portable to all these platforms as well. This indirect portability is especially attractive since SEP may distribute the GNU source code as part of its own SEP software distribution.

PC Hardware

Because of its sheer size, the PC world sets the standards for many computer devices. For example, SEP's NEC PC plugs directly into the two computer projection devices used by the Geophysics Department. Additionally, the PC mass-market offers low prices, customer-friendly warranties and incremental hardware upgrades.

Perhaps more importantly, SEP's graduating students can afford an inexpensive PC, copy the SEP environment, and seamlessly continue their research wherever they may go. Former students with SEP computer environments on their own PCs may become an important source of bug fixes and software improvements.

Because the PC hardware is not tied to a single vendor's operating system, the potential utility of the hardware is multiplied. A Linux-PC can be configured to allow the user a boot-time choice between Linux and Windows. A Windows session allows a user to use standard PC software such as Excel, Power Point, or Quicken on his PC. The PC versions of these applications are often cheaper than the corresponding Macintosh software. Also, SEP may want to experiment with Microsoft's Windows NT as an alternative future UNIX system.

Interesting details

- We chose the Redhat commercial distribution of Linux. Redhat maintains an ftp site offering the complete Linux distribution and includes a utility that can install upgrades directly from the ftp site. Additionally, Redhat offers CD-ROM versions of its releases.
- Linux supports a wide variety of hardware due to the multi-vendor nature of the PC market. We are confident that there would be no problem supporting our older SCSI disks.
- Linux CD-ROM handling is as easy as CD-ROM handling on a Sun; in contrast to many UNIX vendors, Sun and Linux support the Rock Ridge File System Standard.
- NAG offers its Fortran 90 compiler for the Linux platform.
- SEP's off-site system administration, CAST, and Stanford's networking services (AFS) support Linux machines.

OUR PORT TO LINUX

We ported the entire SEP environment to the Linux-PC platform and tested its functionality. We copied, compiled, and installed the SEP specific software. As usual that required a little debugging and fixing. However, most of the public software was available as precompiled versions; no debugging or adapting was required.

Things we needed to do

The following table lists software we had to port to Linux. The effort expended on these ports varied from simple recompilation to significant configuration and debugging.

Package	Description
Xtpanel	Simple GUI script language
xtex	DVI viewer with active push buttons
SEPlib	SEP's processing system
gmake	SEP's patched copy of GNU make-3.74
gmake rules	SEP's makefile rules
setup	SEP's shared environment
automounter map	System file for mounting information

Things we did not need to do

In the past, SEP spent a considerable effort to install popular tools such as `emacs, gcc, ghostview`, etc. on all its UNIX platforms. For Linux these tools were available in precompiled form and most were included as installation options on the CD-ROM. The following table lists just a few of the tools that were installed with a single click or command.

Package	Description
ghostview, ghostscript	Postscript viewer and interpreter
emacs	Editor and much more ...
gcc, g++, f2c, g77	Compilers
TeX, LaTeX2e	Typesetting languages
perl	Practical extraction and report language
gawk	Document formatting language
XFree86	The X11 release 6 window system
xfig	Idraw alternative: interactive drawing tool
xdvi	DVI Previewer
java, javac, jdb	Java language development tools

WHERE DO WE GO FROM HERE?

Currently, SEP researchers spend a considerable effort maintaining their computer environment. The SEP research group was shrinking, and even more importantly, not all of the students showed an interest in computers. SEP has only been partially successful in tapping outside help. Consequently, a few SEP students are burdened with maintaining the complex SEP computer environment. These system administrators often find the additional tasks detrimental to their research efforts.

We believe that SEP should drastically reduce the complexity of its computer system: we especially would decrease the number of supported platforms, since the amount of maintenance work increases sharply with the number of platforms.

Concept of a computer environment

To simplify SEP's computer environment, while preserving its technical capabilities, we suggest restricting the environment to two types of platforms: a base platform (SEPers desktop) and special purpose platforms (a compute server such as the SGI Power Challenge and maybe a Promax, AVS, or Focus server).

The **base system** should be a capable UNIX platform that offers email, NFS, NIS, printer, and web support, as well as reasonable compilers. The platform should require little maintenance and, ideally, should excite its users. We believe Linux is an excellent candidate for this role.

The **special purpose platforms** are limited to their specific task. We would restrict the general purpose software on these machines to a bare minimum. We also would limit the number of special purpose platforms to as few as possible. The authors believe, that currently, the SGI supercomputer, and one or two HP graphics workstations, could serve as our sole special purpose machines.

Reducing maintenance

At SEP a Linux "base system" could potentially replace four existing computer platforms (Apple, Sun, HP and IBM). Additionally, SEP could eliminate the porting of GNU software to those platforms and could drastically simplify the general system administration and the maintenance of X11 and TEX. Finally, SEP could avoid the transition from SunOS to Solaris.

We envision replacing the Mac, Sun, IBM, and HP system administration jobs with a single Linux system administration job. Since all the software we installed from precompiled binaries can be upgraded that way, the Linux administration job should be less time-consuming than any one of the four jobs it would replace. The use of the precompiled software would also eliminate the GNU and X11 job entirely.

The system administration of the special purpose platforms would continue. However, that should be limited to one or two machines and maintenance of these machines would be limited to a few important applications.

Our plan could reduce the number of computer jobs to as few as three: system administration for the Linux platform, system administration for the special purpose platforms, and the SEP publication job (TEX, World Wide Web, CD-ROM). Three or four energetic students could manage these three jobs. Besides being manageable, such a computer environment would be attractive for hackers (Linux) as well as computational speed enthusiasts (SGI).

RECOMMENDATIONS

Certainly everyone at SEP would enjoy using an environment that offers all the available software and hardware options. However, such an environment has an associated cost in the form of the time required to maintain it. If SEP will agree that we cannot afford to pay this price, then the authors suggest that the solution is to reduce the SEP computing environment to Linux-PCs (as a "base system" for both office and home use) and to the SGI Power Challenge (as a computational work-horse.)

POSTSCRIPTUM

SEP has ordered 23 Linux-PCs. These will replace the Macs at home and provide 4 additional office workstations. The total number of system administration jobs remains unchanged.

ACKNOWLEDGEMENT

Bill Harlan kept us on our toes by criticizing the amount of time we spent on system administration. Zhang Zhifang attempted to compile vplot on his Linux-PC. Martin Karrenbach and Matthias Schwab completed the first port of the entire SEP environment to Linux. Stew Levin removed the remaining bugs. Jon Claerbout, Bob Clapp and Sean Crawley shared the authors' enthusiasm and helped by kindly attending to our bug reports. James Rickett manages the arrival of the new Linux-PCs and is gearing up as SEP's future Linux system administrator. Biondo Biondi and Mihai Popovici regularly challenged our follies by asking the right questions. Thanks to you all.

REFERENCES

Miller, B. P., Koski, D., Lee, C. P., Maganty, V., Murthy, R., Natarajan, A., and Steidl, J., 1995, Fuzz revisited: A re-examination of the reliability of unix utilities and services: `ftp://grilled.cs.wisc.edu/technical_papers/fuzz-revisited.ps.Z`.

Welsh, M. ., and Kaufman, L. ., 1996, Running Linux: O'Reilly, Sebastopol.