

Libre Software for Computer Science Classes

In 1993, the authors began using 386BSD to teach computer science classes at Madrid's Carlos III University, in Spain. Seven years later, NetBSD and GNU/Linux are the operating systems of choice for several of the University's computer science teaching laboratories.

Jesús M. González-Barahona, Pedro de-las-Heras-Quirós, José Centeno-González, Vicente Matellán-Olivera, and Francisco J. Ballesteros, King Juan Carlos University, Spain

During 1991 and 1992, the landscape of libre software, and of software development in general, was ready for a change. In two different communities, two very exciting events were taking place: 386BSD, a libre derivative of the BSD code, was born, and Linus Torvalds distributed the first versions of the Linux kernel. Soon after, the software community had two libre operating systems at its disposal (for some basic references to online information on libre products and distribution, see the “Online Resources” and “Terminology” sidebars).

In late 1992 at Carlos III University, we were introducing new computer science studies and planning a distributed systems course. We were looking for a software platform suitable for practical lectures on distributed systems (our practical lectures take place in a computer lab and usually involve a programming exercise). Unfortunately, we couldn't afford workstations—we could only share a PC laboratory running MS-DOS, used to teach other computer science subjects. Therefore, we were pleased to discover that 386BSD could run on those PCs and already had the complete environment we needed for the practical lectures. We decided to give the new, exciting libre operating systems a chance—

both GNU/Linux and 386BSD were reasonably stable platforms.

History of the Unix Lab

After a testing period, we established a stable environment that several students helped us maintain and improve. When the semester started, a small group of around 20 students used 386BSD to learn to build client-server applications. The experiment was a success. The students, using the environment, were productive and fulfilled the goal we had set: to build simple RPC-based applications. 386BSD exposed them to a system similar to Unix workstations but at a fraction of the cost and with many new features—such as access to source code.

Terminology

Libre software means software whose source code has been fully freed ("liberated") from any legal entanglements that would otherwise prevent users from distributing and modifying it. Libre software includes GNU Public License (GPL) software, Open Source Initiative software, and other similar forms of open-source software.

The NetBSD operating system

Although several problems were evident—the lack of a GUI and several common applications, difficulties with maintenance, and trouble with some of the hardware devices—we decided to continue using systems based on libre software. After considering GNU/Linux, we selected NetBSD, a successor of 386BSD, as the new operating system for several reasons:

- Familiarity: Our background was mainly in BSD-derived systems, such as SunOS.
- Quality: BSD's networking code was said to be better, especially regarding NFS.
- Availability of bibliographies: books about BSD and its networking code were widely known.^{1,2}

We didn't consider any proprietary systems, in part because of budget constraints, but also because it was obvious to us at the time that NetBSD, and perhaps GNU/Linux, were technically the best available Unix-like systems for PCs.

We compiled most of the applications from source code, because NetBSD lacked a collection of binary packages. Therefore, a lot of source code was always around, and interested students could look at the code that experienced programmers had written—an activity uncommon in Spain (and in many other parts of the world) at that time.

In 1994, our department allocated funds for buying new PCs and developing the Unix Lab. Thus, we introduced our first libre software laboratory. We devoted the next two years to improving the Unix Lab—further integrating it with classroom work and lectures and polishing the user interface. Although GNU/Linux was quickly becoming a viable alternative, we were still committed to BSD derivatives and, in particular, to NetBSD. However, students started asking for GNU/Linux, and colleagues in the Computer Science Department pressured us to switch so they could run Linux-only applications. Looking back, it is astounding that we had almost no pressure to switch to

proprietary Unix-like systems.

The new environment was much more user-friendly. We installed XFree86 in every computer, with a GUI that provided a faster learning curve for students. We also installed new applications, many of them with a GUI.

The switch to GNU/Linux

In 1997, the department decided to renew and expand the Unix Lab. Because of the social pressure for moving toward GNU/Linux, we decided to install it instead of NetBSD in the new hardware. Among the many distributions of GNU/Linux, we considered RedHat, Slackware, and others, but we chose Debian because of

- its development model, which was based on hundreds of developers spread around the world;
- the number of included applications—around 1,000 at the time;
- the development team's spirit, open to external suggestions and fully committed to libre software; and
- advanced package management tools, which simplified large-scale upgrades.

The new Unix Lab required no funding for buying software, letting us allocate additional funds to improved networking hardware and multimedia equipment. This let students use Mbone applications for establishing, for instance, video conferences.

The switch to GNU/Linux has proven beneficial for students, because it is more extensively used in Spain than NetBSD and is much easier to obtain. Currently, the lab runs Debian 2.1 (which includes almost 2,000 packages), with an improved GUI based on Gnome. More and more teachers (in computer science departments and elsewhere) are using or planning to use GNU/Linux for their practical assignments, and we are sharing our

Online References

Debian: www.debian.org
FreeBSD: www.freebsd.org
Gnome: www.gnome.org
GNU: www.gnu.org
KDE: www.kde.org
Linux: www.linux.org
Mozilla project: www.mozilla.org
NetBSD: www.netbsd.org
OpenBSD: www.openbsd.org
RedHat: www.redhat.com
Subjects taught at the libre software lab: gsync.escet.urjc.es/docencia
Xemacs: www.xemacs.org
Xfree86: www.xfree.org

About the Authors



Jesús M. González Barahona teaches at the King Juan Carlos University, Móstoles, Spain. He joined the Computer Science Department of the Carlos III University, Leganés, in 1992, where he worked until September 1999. His current research interests are distributed systems, multicast protocols, and fault-tolerant systems. He is also interested in the technical and teaching implications of libre software. He received his PhD in telecommunication engineering from Universidad Politécnica de Madrid. He is a member of the IEEE Computer Society and of ATI (Association of Computing Technicians). Contact him at jgb@gsync.esct.urjc.es or jgb@computer.org.

Pedro de-las-Heras-Quirós teaches at the King Juan Carlos University, Móstoles, Spain. He worked at the Computer Science Department of Carlos III University, Leganés, until October 1999. His research interests include distributed systems, fault tolerance, and large-scale network protocols and applications. He is also interested in the impact of libre software in the software industry and academic circles. He earned an MS in computer science from the Universidad Politécnica de Madrid and expects to receive a PhD in computer science during 2000 from the same University. He is a member of the IEEE, ACM, and ATI. Contact him at pheras@gsync.esct.urjc.es.



José Centeno-González teaches at the King Juan Carlos University, Móstoles, Spain. He joined the Computer Science Department of Carlos III University, Leganés, in 1993, where he worked until September 1999. His research interests include distributed-systems programming, and protocols for distributed systems. He is also interested in the impact of libre software on computer science teaching and industry. He graduated from Universidad Politécnica de Madrid with an MS in electrical engineering and expects to receive a PhD in computer science from the same university this year. Contact him at jcenteno@gsync.esct.urjc.es.

Vicente Matellán Olivera teaches at the King Juan Carlos University, Móstoles, Spain. He joined the Computer Science Department of Carlos III University, Leganés, in 1993, where he worked until September 1999. He is interested in multiagent and distributed systems and has worked on applying these topics in robotics. He received his PhD in computer science from the Universidad Politécnica de Madrid. He is a member of the IAS (Intelligent Autonomous Systems) and ALI (Association of Computer Science Graduates). Contact him at vmo@gsync.esct.urjc.es.



Francisco J. Ballesteros does research on distributed and adaptable operating systems and teaches operating systems and advanced programming courses at the King Juan Carlos University, Móstoles, Spain. He received his PhD in computer science from Universidad Politécnica de Madrid. Contact him at nema@gsync.esct.urjc.es.

experiences with almost every university in the Madrid area.

Lessons Learned

After many years of operation, the Unix Lab has become, for the most part, a positive resource. There are, however, both advantages and disadvantages as compared with a more traditional lab based on proprietary software.

Advantages

The software is available to students without restriction. They can install exactly the same software environment on their home computers with no licensing problems or costs. In addition, they can install more applications for their own use and modify them without bothering with expensive proprietary programs or illegal copying. Moreover,

they gained immense learning value by being able to inspect the internals of the software and system.

Similarly, all the teaching materials (including the software) are available to students and colleagues at other universities without restrictions. We can thus easily share our teaching tools and techniques with colleagues. Another advantage is that libre software ensures “brand-neutral” teaching—any reseller can provide the software because no one owns it. This way, the University doesn’t endorse any particular brand or company.

Also, source code availability for computer science students is extremely important so they can learn how other programmers solve common problems. Because of the availability of source code, we were able to customize and adapt the software to our environment. This included porting rare applications to the operating system we were using.

Another positive side effect is extremely low software costs. Funding for software laboratories is limited, and with libre software, the only significant costs are hardware and maintenance.

In addition, the libre software development model ensures that if a product has a strong enough user base, it will maintain high quality standards. Among libre software programs, there is a range of technical quality, but our experience has been that if you carefully select the system’s components, it is more reliable and capable than its proprietary counterparts.

Finally, in several niches, libre software is already the preferred choice. Today, the most prominent of these niches is probably the Internet infrastructure and related services. Many of our students are finding jobs in this area because they have some background in similar or exactly the same mainstream programs used in that environment.

Disadvantages

Of course, using libre software for teaching also has its downside. It is sometimes perceived as second class or unprofessional, which can negatively influence the students’ motivation. Similarly, many students feel they need to learn specific characteristics of the so-called standard programs, instead of general skills about using those kinds of programs. Some students are reluctant to install a new operating system on their home computers.

There are also institutional barriers to libre software. Many departments within universities are reluctant to use or directly oppose libre software, not on technical grounds, but based on their unfamiliarity with the software. Though less common, we also experienced problems regarding a lack of support for certain hardware devices.

Another problem is that finding technical staff well trained in the installation and administration of libre software is still not easy.

Many of these disadvantages are not directly related to libre software characteristics, but rather to its current situation and perception. Should this situation change in the future, many of these problems will disappear or else turn into advantages.

In October 1999, our team moved to the King Juan Carlos University, and we have already started implementing the lessons learned at the Carlos III University. In addition, we are exploring the suitability of libre software for teleteaching and for implementing virtual campus facilities. We are also holding discussions in which we explain to

students the benefits of a lab based on libre software.

Both in Carlos III University and King Juan Carlos University, we have helped create Linux users groups, which are a good source of informal support for both students and teachers using libre software. In those groups, we can also find well-trained technical staff to help maintain the lab.

We are very interested in the use of libre software for teaching computer science and welcome any comments you have about your experiences in this area. ☺

Acknowledgments

This article, and the experiences it describes, has only been possible thanks to the outstanding work of hundreds of libre software developers. We are very grateful to all of them.

References

1. G.R. Wright and W.R. Stevens, *TCP/IP Illustrated*, Addison-Wesley, Reading, Mass., 1995.
2. M.K. McKusick et al., *The Design and Implementation of the 4.4 BSD Operating System*, Addison-Wesley, Reading, Mass., 1996.

**IEEE
COMPUTER
SOCIETY**

**Career
Service
Center**

- Certification
- Educational Activities
- Career Information
- Career Resources
- Student Activities
- Activities Board

<http://computer.org>

Advance your career
Search for jobs
Post a resume
List a job opportunity
Post your company's profile
Link to career services

<http://computer.org/careers/>