J7.7 OPEN SOURCE TECHNOLOGIES IN SCIENCE EDUCATION: WHAT'S YOUR GEEK IQ?

Stephen L. Arnold *
Allan Hancock College, Santa Maria, CA

1. INTRODUCTION

The goal of education, whether public or private, K-12 or post-graduate, is fundamentally an altruistic one. Obviously, a well-educated public can better come to grips with both the environmental and technological issues that face today's societies. Knowledge empowers everyone, and free software (or Open Source) technologies such as GNU/Linux and BSD follow almost identical themes; by providing the same robust, standards-based capabilities to everyone, they not only foster education and cooperation, but also self-determination. By distributing the source, users have the freedom to fix bugs or add enhancements as their requirements dictate. Supporting and using open standards also helps ensure equality of access to these technologies for everyone. The GNU General Public License (GPL), among many other Free Software licenses, is intended to maintain these same rights and freedoms, by ensuring that no one individual or organization can take control. In this way, Open Source technologies are an ideal fit for many organizations, most especially those in a public education setting. The low overall life-cycle costs and flexibility are also a significant benefit to both public and private-sector organizations.

There are undoubtedly already many such Free Software and Open Source technologies, as well as in-house and commercial products, supporting various requirements at your own institution; this paper will present a broad overview of the how's, why's, and what's of Linux and other open source technologies, focusing on specific examples of both instructional and research support in multiple settings. The Open Source roll-your-own approach is also contrasted with 3rd-party solutions, including internal and external resources.

The definitions used here follow those of the Free Software Foundation (FSF), the principal organizational sponsor of the GNU Project. In the true spirit of the GPL, Figure 1 is made available on the GNU Project home page, and illustrates several types of software as defined in The Philosophy of the GNU Project (Stallman, 2004). Additional details, as viewed by the author, are provided in Table 1. The shareware category had its hey-day back in the days of mail-order CD-ROM distribution and limited Internet access, and is not very common today. Crippleware is “free” commercial software with some serious limitation (i.e., a time-limit on usage or restricted feature set), and seems to act as a marketing loss-leader.

In contrast to the majority of commercial software licenses, whether or not the software is freely available, the GNU definition of “free software” embodies four basic freedoms: a) the freedom to run the program, for any purpose, b) the freedom to study how the program works, adapt it to meet user requirements, and fix bugs, c) the freedom to redistribute copies as desired, and d) the freedom to change the program, and release these changes to the public, so that the whole community can benefit.

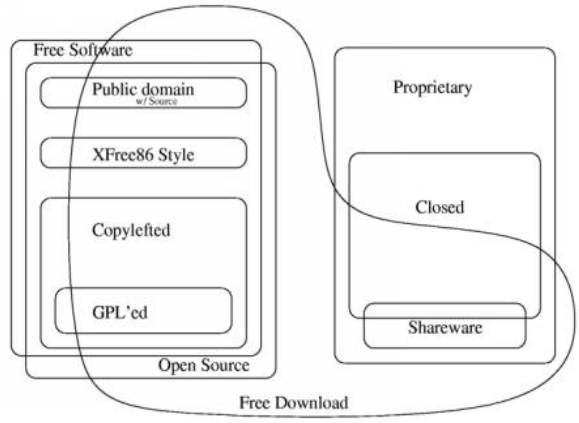

Figure 1. Definitions of Free and Non-Free Software (from gnu.org)

Freedoms b) and d) above obviously require access to the source code, thus, the GPL requires distribution of the source when software released under the GPL is used or modified and released again. Note, simply using or modifying GPL'd software internally does not require distribution or release of internal code, nor the release of any proprietary information or intellectual property. Only when software based on GPL'd code is released to the public are the source code and any associated changes required to be released as well.

The above principles, as defined in the GPL, are inherently compatible with the mission of public institutions, regardless of purpose (i.e., public software for public education). The same freedoms apply to any individual or organization who so choose to avail themselves of the technology, as well as the related communities of developers, users, and commercial consulting support.

Both education and the scientific process itself are dependent on freely shared and open ideas, including those expressed as program source code. In fact, without access to the source code, scientific verifiability becomes increasingly difficult, if not impossible (Gazelter, 1999). Releasing software under the GPL

helps ensure the ideas and their implementation in software are preserved and made available to all, in the same way that books preserve the ideas of past generations.

Open standards are the second key to making modern technology available to all. Only a public infrastructure can serve the needs of the public, and our modern infrastructure is based on protocol standards such as those developed by the Internet Engineering Task Force (IETF). Open standards, whether for software, network protocols, or file formats, serve to mitigate or prevent outside vendor control over others (i.e., vendor lock-in). As an example, the IETF requires demonstrated interoperability for any Internet standard specification proposed for public use. Without such open and publicly controlled standards, communication and data sharing between different brands or models of computer would be virtually impossible. For the Linux community, the Free Standards Group helps develop and promote a common set of behavioral specifications, tools and ABIs across Linux platforms.

2. FREE AND OPEN SOURCE SOFTWARE

As attractive as the above ideas are, there are many more compelling reasons to examine free software alternatives. One such topic is described in a report from the National Research Council entitled "Being Fluent with Information Technology (NRC, 1999). In it, the authors ask the question "Why know about information technology?", focusing on what an individual must know and understand about information technology in order to use it effectively and productively for their own purposes. The rationale motivating an understanding of information technology spans at least 4 broad categories: personal, workforce, educational, and societal, all of which are intimately connected with our educational system, impacting our economy and society as a whole.

Only through increased exposure at all educational levels can our students become truly fluent with information technology, and this exposure must be more than a single vendor's black-box wizards. GNU/Linux and other free and open source software provide the only way to truly look "under the hood" and learn core concepts, as well as get real hands-on experience without incurring any license fees or legal repercussions.

2.1 Example: Python

One such example technology is the programming environment and object oriented language known as Python. Python is an excellent scripting language and wrapper interface to libraries and legacy codes, as well as a full-featured Object Oriented Programming (OOP) language. At its simplest, Python is an interactive environment with intuitive variable types (i.e., a weakly-typed language) and built-in high level data structures such as lists, dictionaries (hashes), and tuples. Python is said to come with "the batteries included" which is a reference to the extensive set of included libraries for everything from network services to mathematics. Python was designed for ease of use, and, as evidenced in Kirby Urner's excellent essay "Python in the Mathematics Curriculum" (Urner, 2004), is truly the programming language for everyone (even K-12 students).

Python itself is available in both source and binary forms for a variety of platforms, along with a large library of documentation and tutorials. Python software is available for a variety of tasks, from web application platforms such as Zope, to scientific and numerical analysis (ScientificPython).

2.2. A Cornucopia of Software

Personal experience includes document production in multiple formats (HTML, PDF, DOC), course data and student data management, web server administration and application development, curriculum development, and communication with both students, faculty and the administration.

The office productivity suite from OpenOffice.org is used for production of documents and presentation materials in various formats, as well as management of student information and grades. Course delivery outside the classroom, as well as in, makes extensive use of the Zope web application framework for both on-line documents and interactive applications (e.g., the GeoZone discussion forum).

Future geography course plans include MapServer, an OpenSource development environment for building spatially enabled Internet applications. The MapServer software builds upon other popular GNU and Open Source systems such as Shapelib, FreeType, Proj.4, GDAL/OGR and others. The MapServer system also includes MapScript which allows popular scripting languages such as Python, PHP, Perl, and soon even Java, to access the MapServer C API. Zmapserver is a Zope product (essentially a plug-in) that provides an interface to MapServer within Zope.

The ad-hoc categories and examples shown in Table 2 give a glimpse into the depth and breadth of available technologies, as an exhaustive list is well beyond the scope of this paper. Such static lists are also in contrast to the fluid nature of modern digital media and communications; witness the popularity of news and software sites such as SlashDot and FreshMeat.

As the World Wide Web is really the definitive source for the latest information, the following short list of software sites is current as of this writing:

- Python: http://www.python.org
- GNU Project: http://www.gnu.org
- Open Standards: http://www.freestandards.org/
- Gentoo Linux: http://www.gentoo.org
- CentOS & cAos Linux: http://www.caosity.org

- Zope: http://www.zope.org
- MapServer: http://mapserver.gis.umn.edu/
- OpenOffice: http://www.openoffice.org
- UCAR: http://my.unidata.ucar.edu/content/software/
- The author's geography course materials and web applications: http://arnolds.dhs.org/geography
- Short Example List of Earth Science Software: http://arnolds.dhs.org/geography/software
- The author's Gentoo ebuilds and RPM packages: http://arnolds.dhs.org/software

Although all major Linux distributions, e.g., RedHat, Debian, SuSe, and CentOS, include a large selection of major software packages, the Gentoo Linux distribution provides the largest selection of additional packages, from scientific applications to obscure programming languages (in addition to the same core packages as above). The Gentoo portage tree (the available packages) currently contains 7941 ebuilds (although this includes multiple versions / revisions of individual packages).

3 VIEW FROM THE TRENCHES

It all starts on the desktop, thus all course materials, including but not limited to, course outlines, schedules & calendars, course notes & handouts, exams, web pages, and software, are produced on a Gentoo Linux desktop (typically Gnome). Other components such as Zope require a dedicated network server for maximum benefit, along with sufficient network bandwidth (although Zope can still be run on a local desktop if desired). The following software is central to a well-equipped educator's desktop:

- Gnome Desktop Environment: Integrated desktop applications with modern features such as drag-'n-drop, dynamic menus, auto-mounted removable media, etc. Includes Nautilus file-manager and utilities for graphics, text, archiving, etc. A lighter-weight alternative (i.e.,, smaller memory footprint, fewer core processes and package dependencies), either for an older machine with minimal resources, or perhaps a sub-notebook, would be Xfce-4.
- Desktop Document production: OpenOffice is used to generate all formats from master text documents. Student information and course data is maintained using spreadsheet documents, and lecture presentation slide-shows as well.
- Data Analysis and Graphics: Octave and gnuplot provide equivalent functionality, and even m-file compatibility with the basic Matlab(TM) package. Other scientific and statistical packages exist, as well as discipline-specific models and databases for everything from mesoscale meteorological analysis and forecasting to bio-informatics.
- Web Services: The Zope web application framework is used to serve course content and other information (schedules, etc), as well as host interactive applications such as the discussion forum, and web applets such as pymetar, zweather, and zmapserver.
- Third-Party Services: Other providers, such as Blackboard and Prentice Hall, have been evaluated with limited success. The textbook companion web sites have provided limited utility as optional assignments, while additional topical sites are used on an ad-hoc basis.
- Real-time Chat and Conferencing: Internet Relay Chat (IRC) was recently introduced as Virtual Office Hours, however, students have not yet utilized this resource. Video conferencing with GnomeMeeting is also an option, especially for distance learning.

Gentoo Linux is seen to be an almost perfect match between free software and education / research needs. Gentoo supports scientific & high performance computing, as well as general education, with a huge collection of cutting-edge applications, all optimized to extract the maximum performance from a given processor.

The main difference between Gentoo Linux and most other distributions is that Gentoo is designed to be built from the latest stable source packages, as opposed to a pre-built set of binary packages. When configuring and building a Gentoo Linux system, the user gets to specify his or her own set of compiler optimizations, so everything is built against their own processor and hardware architecture. Each package is also built using user-specified flags that control how each one is built, which optional features are supported, and which other library/package dependencies are built.

When performing a Stage 1 installation, the system contains only the build tools and basic system components required to run the system (i.e. a kernel, system logger, and basic system administration tools). At this point, the system requirements can dictate additional functionality and tools. The end result is a highly optimized system running the latest stable version of each package, and only the required packages. In many ways, Gentoo is still a standard Linux system, although some key characteristics are different from other Linux systems:

- A Gentoo system is always current; syncing and updating the system brings it to the current baseline (so there's no such thing as a “system upgrade”).
- Only what's needed is built (as determined by the build configuration) and everything is optimized for the host processor (as opposed to generic i386 binaries).
- Everything from installation to system configuration to deployment and maintenance is up to the user and not the distribution vendor, yielding one of the most flexible Linux systems available today.

The majority of the author's home systems currently run Gentoo (in both desktop and server roles), including all family member's PCs. In addition to standard network services, such as DNS, WWW, and email, a Gentoo server allows me to build, deploy, and maintain a small network of more than a dozen machines with minimum time and hassle. After many years of Linux experience with at least 2 dozen different distributions, Gentoo has shown itself to be an excellent learning tool, as well as an extremely functional desktop and manageable server platform.

4 RESULTS AND DISCUSSION

Overall, none of the results discussed here nor even the production of typical course support materials, would have been possible without GNU tools and other open source software packages. The cost of a commercial office suite, including the ability to create PDF documents, easily exceeds the cost of building two Linux machines, so the reader is invited to draw their own conclusions.

Beyond the basics, however, open source software brings many other benefits such as the ability to learn from the source code, extend its functionality, fix bugs, and generally make use of the software as discussed in Section 1 above.

In addition to sharable document formats such as PDF, all other documents (spreadsheets, presentations, etc) have transferred easily to campus computers, both Macs (as are used in the smart podiums) and Windows PCs. Minor formatting errors, mostly font-related, have been observed, but even repeated translations seem to cause no problems.

The ease of creation and management of course web sites is directly related to the modular Zope architecture and built-in management interface for both content and users. Creating and configuring a Squishdot discussion forum takes less than 5 minutes; the new site is then ready to add users and custom topic icons (finding the right icons takes longer than creating a new site).

4.1. Assessment: Student Scores

Current assessment methods used include a variety of exams, in-class group activities, individual homework assignments, and web applications. The majority of assignments are required; open-book exams, group and individual assignments and activities, and a semester project (e.g., contributing relevant articles and comments on the web-based discussion forum). Optional or extra-credit points are available through the web-based quizzes and exercises provided by the textbook publisher (in the example of the discussion forum project, additional articles or comments beyond the minimum are also accepted).

The student scores shown in Table 3 are from several evening geography courses, taught over the last 5 years (early data was lost in a proprietary data format). Although each data set is small (due to class size), and includes a different set of projects and assignments, there is still something of value to be learned. Outliers (informal drop-outs) have been removed.

The first and second rows of data (for Fall '01 and '02) show the results using standard exam questions (drawn from the publisher's “test data bank”) and the textbook companion web site for weekly required assignments. Participation rates were marginal, with only 25% of the class submitting at least 75% of the homework, and scores were corresponding low (as were previous semesters).

The third row of data (for Spring '02) shows the results of introducing the Squishdot discussion forum as the required semester project (one topical article posting per week, plus two comments on other articles). The participation rate jumped to almost 100% (all but one student completed 100% of the assignments), which may illustrate the importance of peer-interaction and real-life issues (recall that previous web-based assignments were apparently less attractive).

The Spring '03 semester (shown in the fourth row) introduced open-book exams, to replace the canned materials used previously, for a Human Geography course. The discussion forum project was replaced with a new semester project requiring students to research both the histories and migration routes of one or more family members, using any and all references (i.e., Internet resources, family members, letters, etc).

Subsequent semesters retained the open-book exam format for all courses, and allow the use of the textbook web site (if available) to obtain extra-credit points. The Squishdot discussion project is again underway for the Physical Geography course as of this writing; planned future enhancements include an article/author rating system.

Since the discussion forum functions are limited to one type of interaction only, in this case web “Blog” publishing, the next logical step would be a web portal environment (this is currently under evaluation on the author's web site).

The student data is relatively coarse and under-sampled, and was not intended to be strictly controlled from one semester to the next, since new project ideas, exam questions, and supporting technologies are continuously introduced. The data suggest, however, that one of the most promising components is the concept of “environment”, i.e., the discussion forum only allows students to post (although anyone can browse). The community environment is also one of the key factors in the success of community-driven open source projects such as Gentoo Linux, so this is not without precedent.

In addition to the “environment” factor, the second important point seems to be that of “connection”, i.e., how do the course materials and concepts connect with the students' real lives and experiences? The response to the above assignments suggests that both of these factors together can at least engender a fairly high level

of both interest and participation in required course assessments and activities.

The overall response to web applications and other on-line content delivery mechanisms has been very good, with most students participating in various on-line exercises at a high level, many enthusiastically (including some low-income and ESL students). All web applications and content are apparently not created equal, however, as the chapter exercises on the textbook's "companion" web site only seem to evoke a marginal response when used alone.

In general, the ability to submit assignments via the web and email has been well-utilized by the majority of students, and even critical for some students to retain their grades, given international travel and other family commitments (e.g., just before the winter break). However, a handful of students, regardless of the age group, are still somewhat hindered by a lack of experience and have a correspondingly low self-confidence or "comfort level" with the technology in general (relying on hard-copy or other workarounds for assignments).

4.2 Issues: Logistics and Training

The core issues impeding even higher participation levels and student scores are mostly related to logistics and technology training, including:

- Student Technology Proficiency: Medium on average, however, with a large range and variance. Requires additional information technology support, however, universal freshman technology courses are recent and not very wide-spread (still optional for most majors).
- Basic Skills: Also medium and highly variable in all areas, however, the Hancock enrollment population is high in recent immigrants and other ESL students. Requires additional math and writing support.
- Provision and Deployment of Technology: Effective use and management of resources requires a dedicated campus infrastructure, requiring at least a low-end Linux server per department (including appropriate routers and firewalls), as well as a shared T-1 data connection at a minimum. Effective IT support and deployment policies are also important, if only for network stability and non-interference with existing services. However, a small number of courses can easily be supported on a single machine and budget ADSL connection (i.e., 768k/128k) at the expense of some end-user browsing performance.

Linux-based web hosting can be a cost-effective, scalable, and flexible alternative, the main limiting factor being network bandwidth (depending on data and usage requirements). High-availability configurations and blade servers are currently popular, including hooks for several popular database back-ends, directory services, and authentication mechanisms (both open source and commercial).

The author's geography courses are currently hosted on a low-end Linux web server and home DSL connection (a dedicated geography web server is in the purchase order queue; total cost $600 plus build time). Additional planned services include live meteorological data and web-based analysis tools, as well as web-based GIS & mapping tools (requires additional network infrastructure and bandwidth for live data feeds).

5 SUMMARY AND CONCLUSIONS

Traditional exams constructed from a variety of questions provided by the text-book publisher (eg, multiple choice, true/false, essay, etc) typically result in the lowest scores. Additional weekly quizzes using similar questions (actually requested by the students) did not raise scores appreciably (Table 3, first two rows).

Open book exams, using a variety of thought-provoking questions, seem to promote the highest scores, although they require more work to prepare and grade than using the test-generation tools supplied by the publisher (eg, the Test Manager software provided with Prentice Hall geography texts). The students also seem to feel like they've really accomplished something, which is always a good thing. Effective questions must involve core concepts (eg, geostrophic wind, adiabatic warming and cooling) and provide for a range of expression types, including definitions, diagrams, essays, plotting data, and explaining the key relationships.

Extra credit assignments, whether traditional or on-line, and regardless of the point totals, typically help less than 10% of the class raise their grade (e.g., 1 or 2 students would actually benefit by a letter grade increase), and usually not those who need it the most. To see any large-scale benefits, assignments must be required, however, as noted earlier, electronic submission allows for student travel and other contingencies.

The highest participation rates in homework assignments are seen when the projects involve the use of conceptually-targeted assignments incorporating both technology and human interaction. For example, in the group discussion project, the articles are required to be topical and course-related, as well as have some impact on the local community (if possible). The students are also required to read and comment on articles posted by their classmates, which they seem to enjoy. The first time this approach was tried, participation went from 25% of the class completing 80% of the homework (using the text-book web site) to 92% of the class completing 100% of the above assignments on the discussion forum. The "community environment" of the discussion forum, as well as the convenience of an entirely web-hosted project, appears attractive enough to facilitate significantly more homework volume (and participation is the first step).

The additional chapter exercises on the publisher's course-companion web site are now used as extra-

credit assignments (graded on participation only), since they do not seem to produce the same interest level in the students, nor do they connect the course concepts so directly to their own lives. On-line participation rates in general are also significantly higher than with traditional paper assignments (which are still used a few times per semester class), however, the execution of the former is still somewhat hampered by individual student's proficiency with basic tasks such as handling web forms, email, and electronic documents (less than 10% usually fall into this group).

Above all, the use of GNU/Linux, Zope, Python, and other freely available open source technologies allows the flexibility, productivity, and freedom (frpm both license fees and the associated restrictions) to create and deliver course content, as well as attract students in positive and measurable ways. Some commercial alternatives are just too expensive for students or part-time instructors, or even many institutions in today's budget climate, and in reality, most just aren't designed for the kind of end-user requirements and flexibility discussed here.

Keep in mind, without the “environment” and “connection” factors discussed previously, the use of technology for its own sake does not seem to significantly enhance either interest level or participation (nor performance, for that matter), however, when applied in concert, the use of appropriate technology truly allows things to happen that would otherwise be impossible (or at least highly unlikely). The fact that GNU/Linux and other open source technologies are essentially free of license restrictions, as well as low in life-cycle costs, is just the icing on the cake.

REFERENCES

Gazelter, J. Daniel, 1999: “Catalyzing Open Source Development in Science,” paper presented at the conference “Open Source/Open Science,” Brookhaven National Laboratory, October 2, 1999 (slides available at http://www.openscience.org/talks/bnl).

National Research Council, 1999: *Being Fluent with Information Technology.* Report of the Committee on Information technology Literacy, Computer Science and Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications. Washington, D.C.: National Academy Press. Available on-line at http://books.nap.edu/books/030906399X/html/R1.html

Stallman, R.M., 2004: “The Philosophy of the GNU Project,” Free Software Foundation (FSF), http://www.gnu.org/philosophy/

Urner, Kirby, 2004: “Python in the Mathematics Curriculum,” paper presented at the PyCon '04 conference, March 24-26, 2004, Washington DC (slides available at http://www.python.org/pycon/dc2004/papers/15/).

Type	Platforms Available	Main Characteristics
Historical “freeware” Archives: Subtypes include “shareware” and “crippleware”	Windows, MacOS, OS/2, other	Format: usually binary-only. Licensing: usually copyrighted with restricted distribution. Quality: generally low (high variance). Origin: mostly individual developers and small software shops, some university and government organizations, a few larger technology firms. Destination: mostly dead-end and unmaintained packages for out-dated operating systems & hardware platforms, or hardware-specific drivers/utilities. Examples: WUStL archives, Walnut Creek CD-ROMs.
Public Domain Software:	Various (Linux, BSD, commercial Unix, Windows, Mac, other)	Format: usually source code, some binary. Licensing: unrestricted/unprotected (no copyright). Quality: low to high (medium variance). Origin: mostly university and government organizations, some students, individual developers, technology firms. Destination: good packages tend to get conscripted by private companies for their own gain, while the rest go stale and unmaintained. Examples: Various commercial Unix archives (e.g., Sunsite), public institutions (USGS, etc). Most modern code is released as GPL'd or other copyleft software now.
Free Software / Open Source Software: (covers many different subtypes based on license)	Mostly Linux, *BSD, and commercial Unix (on supported hardware), also various Mac and Windows flavors	Format: most licenses require source code to be made available, but binaries are often available for individual packages as well, and most Linux/BSD distributions are available as binary-format installations. Licensing: Various, mostly the GNU Public License (GPL/LGPL) and others (Apache, MIT, UCAR/Unidata). Most licenses are copyleft, i.e., copyrighted free software whose distribution terms do not let re-distributors add any additional restrictions when they redistribute or modify the software. Quality: generally high (low variance). Origin: foundations, public institutions, companies, and individual developers. Destination: literally everywhere, from powering the Internet and many organizations (DNS, Email, WWW) to embedded devices such as TiVo, to NASA's Mars rovers and NOAA's Weather Forecast Offices. Examples: GNU Project (Free Software Foundation), Gentoo Linux (source-based), CentOS, Fedora Project, and RedHat Linux (binary installation, source available), Python (Python Software Foundation), Apache (Apache Software Foundation), SourceForge, Savannah, Unidata, NOAA, NASA, etc.

Table 1. A Taxonomy of Free Software

Earth Sciences:	
Meteorological Analysis & Modeling	Winds On Critical Streamline Surfaces (WOCSS), Local Analysis and Prediction System (LAPS), PSU/NCAR mesoscale model (MM5).
GIS and Mapping	GRASS, GMT, iGMT, FreeGIS, GeoTools, OpenEV, proj-4, GDAL, shapelib, MapServer, Zmapserver.
Data Transports	Local Data Manager (LDM), CORBA, jabber (XML transport), standard network sockets and protocols (e.g., FTP. HTTP), custom protocols.
General Science:	
Numerical Computing	ScientificPython, Octave, Scilab.
Numerical Libraries	ATLAS, BLAS, LAPACK, GMP, Netlib.
Clusters / HPC	Openmosix, MPI, OpenPSV (SMP kernels now standard).
Data Formats	NetCDF, HDF4/5, BUFR, GRIB, ETOPO.
Analysis/Visualization	Vis5D, NCAR Graphics, VTK, OpenGL, GNUPlot.
Operating Systems and Hardware Architectures:	
Linux	Numerous distributions that run on x86, PPC, HPPA, MIPS, ARM, Sparc, etc.
*BSD	OpenBSD / NetBSD / FreeBSD (at least x86).
Embedded Devices	Sharp Zaurus, Ipaq, network devices, single-board computers. Both open source and commercial distributions and tools are available specifically for embedded development.
Network Services:	
Web Application Server Support	Zope, Webware, Jakarta/Tomcat, PHP, cgi.
User-side Services	Web applications/browsers, email, news, chat/messaging, conferencing, security/privacy.
Sub-user	Databases and data transports, firewalls and proxies, name resolution, authentication.
Distributed Computing	seti@home, distcc, various forms of clustering, grid computing.
Security	VPNs, SSH/SSL, NetFilter (iptables), SASL, Kerberos, Intrusion Detection, packet monitoring.
Software Development:	
Programming Languages	Ada, Python, C, Fortran, C++, Java, Perl, Ruby, AWK, SmallTalk, Lisp (and many more).
Development Tools	GCC/GDB, SWIG, SourceNavigator, Insight Debugger, Doxygen, Emacs, bugzilla.
Configuration Management	CVS / ViewCVS, subversion, arch.
Office Productivity and Administrative Support:	
Document production	Multiple formats from a single SGML source document XML, HTML, PDF, RTF, LaTex.
Office Productivity	OpenOffice, Koffice, ProjectMananger, Mozilla.

Table 2. Examples of Free & Open Source Software Useful in Academia

Semester	LOW	HIGH	MEAN	MEDIAN	STD-DEV		N
Fall 2001	45	84	70	76	13		16
Fall 2002	50	87	68	67	12		14
Spring 2002**	35	95	82	87	16		12
Spring 2003*,***	35	97	84	89	16		23
Fall 2003*	82	94	90	91	7		9
Spring 2004*	61	100	90	92	17		26

Table 3. Student Geography Scores by Semester

* includes web-site extra credit

** first use of Squishdot discussion forum project

*** migration research project