1-2014

Linux for Academics, Part II: The Advantages of Free and Open-Source Software

Steven Ovadia
CUNY La Guardia Community College

Behavioral & Social Sciences Librarian, 33:47–51, 2014
Published with license by Taylor & Francis
ISSN: 0163-9269 print / 1544-4546 online
DOI: 10.1080/01639269.2014.866024

Routledge
Taylor & Francis Group

INTERNET CONNECTION

Linux for Academics, Part II: The Advantages of Free and Open-Source Software

STEVEN OVADIA
LaGuardia Community College, Long Island City, New York

The Linux operating system represents an intriguing set of tools for academics looking to take advantage of free and open-source software. This column continues a discussion of Linux (see the preceding issue of this journal for the first part of that discussion), describing the software installation concepts within the operating system, as well as specific software that academics might find helpful. None of the software described in this column is Linux specific, so readers without any intention of trying Linux will still be able to experiment with the software described here.

Installing software within a Linux environment should be familiar to mobile device users accustomed to the "app store" concept as seen in mobile operating systems like iOS and Android. Traditionally, a user searches the Internet for software, either specifically (with the searcher aware of the exact program she wants) or generally (the user knows she wants some version of an image editor), downloading a file from a website or a CD/DVD, and then installing software. With app stores, software is installed by visiting a centralized application, searching it, and then installing via the native application. Linux programs are installed and removed in this same manner.

This workflow is often confusing to Windows users, who are often more familiar with a decentralized software installation process, but centralized software repositories are becoming more common. OS X now has a software center, called App Store, that allows users to search for and install software via a centralized application. Linux software repositories are maintained by the teams behind a particular distribution, with maintainers making sure that software within a repository does not break a user's system. However, distributions offer ways for software not within a repository to be

Address correspondence to Steven Ovadia, LaGuardia Community College, Library Media Resources Center, 31-10 Thomson Avenue, Long Island City, NY 11101. E-mail: sovadia@lagcc.cuny.edu

downloaded and installed, or manually appended to the system. The risk in this is that software from outside an official repository might somehow break or compromise a system.

Every Linux distribution has a slightly different take on installing software. Some, like Ubuntu, have a full-blown software center that will remind users of Apple's iTunes, with graphics and recommendations. Other distributions have a simple, text-based, searchable list of all software available within a given repository. Often, the software is roughly sorted into categories, like multimedia, games, or productivity. Repositories represent the software available for a given distribution. The searching and installation of software, as well as its removal, are controlled by a package manager. The package manager also tracks dependencies, which are the pieces of software required by other pieces to operate. If a user wants to download a piece of software requiring other pieces of software, the package manager handles the automatic download and installation of any required software that was not explicitly requested. Package managers also handle software updates for users, letting them know when updates are available and allowing the software to be updated.

Updates are an important part of Linux. Linux systems tend to be more dynamic than Windows systems, due to the fast rate of updates coming into so many different Linux projects. Different distributions handle updates in different ways. All distributions try to prioritize security updates that, left unfixed, could compromise the integrity of a machine, but some add software updates faster than others. Most distributions are supported for a fixed period of time, with updates ceasing after a certain duration. At that point, if a user wants updates, she will need to move to a newer release. OS X users are used to moving between releases but this could be an unfamiliar process for Windows XP users, who had support for 13 years. Release points allow distributions to update software for a fixed period of time, and then free the project up to move on to newer software once a release no longer needs to be supported. In essence, a new release represents a chance to start over with certain ideas.

Linux distributions that are constantly updating software are called rolling releases. These types of distributions are always updating to the latest versions of software. As a result, they have no release points because they are constantly changing. Many users like this approach because it allows them to be on the cutting edge of new software. However, rolling release distributions require a certain amount of expertise to maintain, since their ever-changing nature makes them more susceptible to breakage, where a system becomes unusable or inaccessible in some way. Rolling release users tend to know how to resolve these breakages, but a certain degree of either expertise or diligence is often required.

Software can be installed either via a command in the terminal or via a graphical package manager. Different distributions use different commands

and different package management tools, but for the most part, it's a fairly simple, well-documented process. When using the graphical package manager, the user just searches for software and then indicates whether she wants it installed or removed.

Some Windows software can be run within Linux. The previous column mentioned the use of a virtual machine to run one operating system inside another, but there is also a Linux program called Wine, which allows some Windows program to run within Linux. The Wine project maintains a list of how well certain Windows programs run in Wine at http://appdb.winehq.org/index.php.

Once the user is ready to delve into the installation of software, there are quite a few options to explore. Users probably want to begin with the Web browser, since that is such an important part of most people's workflow. Every distribution ships with at least one Web browser, but if your browser of choice is not preinstalled, that will probably be the first order of business. Internet Explorer and Safari are not available for Linux, but Firefox and Chrome are. Most Chrome apps and Firefox add-ons should work fine in Linux. Most distributions also provide access to Adobe Flash, which is used by many video sites, in their repositories. Users can also download Flash from the Adobe site. Netflix does not work in Linux, although there are some unofficial workarounds.

In terms of software, none of the projects discussed here are Linux specific, but the Linux implementations of these projects are strong, so anyone using these programs in a Linux environment should not experience too many technical issues. Windows and OS X users are also encouraged to experiment with these projects on those operating systems, even if they have no desire to try Linux.

Microsoft Office is not available for Linux, but there are a number of Office-like alternatives available. LibreOffice and Apache OpenOffice are probably two of the more popular options. The two started off as a single project, with LibreOffice eventually becoming a fork of OpenOffice. *Fork*, an open-source software development term, describes when a single project diverges into two separate ones, with each presumably taking a different fork in the road (open-source software means the code behind the software is publicly available to anyone; this makes it relatively easy for someone to take a project in a new direction, using the public codebase). LibreOffice and OpenOffice are similar, offering a word processor, a spreadsheet tool, and a presentation tool (comparable to Microsoft PowerPoint or Apple Keynote), as well as some other productivity tools. Both open source projects have their fans, but for users accustomed to the various Microsoft Office products, moving to LibreOffice or OpenOffice can be a challenge. But for users not too invested in Office, either project could be interesting to sample—especially

given that both software suites are cost-free, while Microsoft Office often is not, although that varies between institutions.

Something else to consider is that locally installed productivity software is not important for all users. Many users can get by using something Web-based like Google Drive documents or the Microsoft Word Web App, which is a Web-based version of Microsoft Word.

Power spreadsheet users have another consideration: the accuracy of the software. Keeling and Pavur (2011) compared the accuracy of various spreadsheet programs, including OpenOffice and Google Docs (265). They found Gnumeric, a Linux-only spreadsheet program, to be the most accurate and Google Docs (now part of Google Drive) to be the least. Users looking for a certain level of precision might want to keep their work in mind when choosing a tool.

Of course, serious data crunchers probably want something more powerful than a spreadsheet. That's where R (http://www.r-project.org) comes in. R is a statistical language that allows users to manipulate data in many ways, from computations, to creating graphs and charts. The default R environment is nothing more than a terminal where users can use syntax to work with their data sets. The interface can be challenging for anyone used to something more like SPSS, with menus and buttons. Luckily, there are a variety of complementary projects designed to provide a more user-friendly interface for R. One is RStudio (http://www.rstudio.com), which is available for Windows, OS X, and Linux. Another graphical interface is R Commander (http://socserv.mcmaster.ca/jfox/Misc/Rcmdr), which also works across multiple operating systems. Because R is free and open-source, there is a wide variety of tools and resources built around it. There is also a robust R community that is constantly developing tools for R, like the aforementioned RStudio and R Commander. One resource is *R Journal* (http://journal.r-project.org), an open-access, peer-reviewed journal. There are also many online guides to R, as more schools are using it to teach subjects like statistics and data visualization, due to its flexibility and cost-free nature. The University of California at Los Angeles (UCLA) Institute for Digital Research and Education has a very thorough guide to beginning with R: http://www.ats.ucla.edu/stat/r. Carnegie Mellon's open course on probability and statistics (http://oli.cmu.edu) gives users the option of using R to complete exercises, walking the user through R syntax. It should be noted that Ward (2013) compared R to SAS, SPSS, and Stata, three proprietary statistical programs, and found SAS to be the best choice, mostly due to the ability of SAS to handle large data sets and because SAS is commonly used in many industries (116–118). R compared favorably to SAS, though.

Once a user is used to desktop Linux, she'll discover an entire world of free and open-source software she might not have been aware of previously. While these tools might lack the aesthetic polish of commercial software, open-source tools tend to be more flexible. Neteler, Bowman, Landa, and

Metz (2012) eloquently discuss this in their analysis of GRASS GIS, open-source geographic information system (GIS) mapping software (124). They explain that the GIS software market is comprised of free and open-source options and proprietary tools, with GRASS becoming an effective, powerful GIS tool because of the robust community that has developed around it (129). This same kind of support community is not always possible with closed-source projects, where users do not have the same kind of access to the underlying codebase.

Just as the software described here has become innovative and responsive due to the community supporting it, desktop Linux has also evolved in a similar manner. Learning to use a new operating system can represent a significant time investment, but it has the potential to open up an entire new world of software. For social scientists, who increasingly depend upon statistical tools, these programs can represent a way to work more effectively, while also supporting a free exchange of code and ideas. Linux makes this kind of work possible and manageable.

REFERENCES

Keeling, Kellie B., and Robert J. Pavur. 2011. Statistical accuracy of spreadsheet software. *American Statistician 65*(4): 265–73.

Neteler, Markus, M. Hamish Bowman, Martin Landa, and Markus Metz. 2012. GRASS GIS: A multi-purpose open source GIS. *Environmental Modelling & Software 31*: 124–30.

Ward, Brian W. 2013. What's better—R, SAS, SPSS, Stata? Thoughts for instructors of statistics and research methods courses. *Journal of Applied Social Sciences 7*(1): 115–20.