

Linux Newbie Administrator Guide



Linux Newbie Administrator Guide (LNAG)

Stan and Peter Klimas

Feedback welcome at: [LinuxNAG at gmail.com](mailto:LinuxNAG@gmail.com)

November 22, 2009

Preface

LNAG License

The Linux Newbie Administrator Guide (LNAG) ("The Guide") is distributed under the Open Content Licence (<http://opencontent.org/openpub/>) with the following addition:

THE GUIDE IS BEING INCREMENTALLY UPDATED. THEREFORE, TO PROVIDE THE BEST VALUE TO OUR READERS, UNLESS THE GUIDE HAS BEEN SUBSTANTIALLY MODIFIED BY AUTHOR(S) OTHER THAN THE ORIGINAL GUIDE AUTHORS, ANY DISTRIBUTOR SHALL DISTRIBUTE A REASONABLY RECENT VERSION OF THE GUIDE, I.E., ITS MOST RECENT VERSION OR A VERSION NOT OLDER THEN ONE YEAR ON THE DATE OF WEB SERVING, CD WRITING, OR HARD COPY PRINTING. THE MOST RECENT VERSION OF THE GUIDE IS AVAILABLE AT [HTTP://LNAG.SOURCEFORGE.NET](http://LNAG.SOURCEFORGE.NET).

THIS LICENCE MEANS THAT ANY PUBLICLY ACCESSIBLE MIRROR MUST BE UPDATED AT LEAST ONCE A YEAR, IF A NEWER VERSION OF THE GUIDE IS AVAILABLE. PLEASE DO NOT CREATE A MIRROR IF YOU DO NOT INTEND TO UPDATE IT.

All the code examples are distributed under the GNU General Public Licence (<http://www.gnu.org/copyleft/gpl.html>).

The maintainers of this Guide (Stan and Peter Klimas) can be contacted by email: LinuxNAG at Gmail.com The latest version of this guide is available at <http://lnag.sourceforge.net>.

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0, 8 or later <http://opencontent.org/openpub/>

Acknowledgments

The Linux Newbie Administrator Guide (LNAG) is hosted *FREE OF CHARGE* on the DotSrc.org server at Aalborg University, Denmark <http://lnag.sourceforge.net>. Thanks to Esben Haabendal Soerensen jbart@sunsite.auc.dk.

Thanks to Kenan Bektas, VP Engineering for hosting our free, timely updated North American mirror: (<http://dbstreams.ca/mirrors/linux-newbie/>). Thanks to Andamooka for hosting LNAG among their great free books. We have to figure how to update it.

Major help and advice was received from (alphabetical order): Alan W. Irwin jirwin@beluga.phys.uvic.ca, Benjamin Smith jbens@saber.net, Barbara Klimas, Ben McCosh, Bill Staehle jstaehle@netvalue.net, Bill Unruh junruh@physics.ubc.ca, Brian Kelsay ripicrd6@kcinter.net, Ding-Hou Lee jdl105@columbia.edu, Gary jswear@aa.net,

Greg Mizell ;GMizell@peek-traffic.com, Jaakko Alarto ;pikkumyy@ikiliekki.org, Jeff Greenlee ;greenj@analogy.com, jeff covey ;jeff.covey@pobox.com, Jo ;rainbow@linuxfocus.org, Juhapekka Tolvanen ;juhtolv@st.jyu.fi, Ken Foskey ;waratah@zip.com.au. Special thanks to the Portland Linux User Group for review!

Thanks to the several dozen others who sent comments, minor corrections, good suggestions, or good word to us. We always appreciate any feedback. We received no flames so far :-). Sorry if we could not answer specific questions you sent us on your particularly annoying problem :(Sorry we are unable to provide free support for individual users. Sorry we do not provide any support to MS Windows users.

Thanks to the few thousands who developed GNU/Linux. Should I mention Richard Stalman and Linus Torvalds by name?

Other Matters

You can see that this guide was written by newbies for newbies. It should never be considered an authoritative source on any topic—there are much more exhaustive docs, typically more difficult to read too :-), most of them available right on your Linux system in the directory `/usr/doc`. Please e-mail us immediately if you spot a mistake that can confuse or mislead a new Linux user—this work is in progress and the current version may contain such mistakes. Don't use this guide if your life or well-being was to depend on it!

If you wanted to contribute a part (on a useful Linux topic of your choice), we would like to hear from you—we will be happy to include a part with your name, etc. The only condition is that your advice be easy to follow by newbies like us.

If you create your own mirror, pls make sure to update it at least from time to time. LNAG is under development and inaccuracies are found and corrected on regular basis. Our licence requires updates (if available on <http://lnag.sourceforge.net>) else please remove your outdated mirror.

Hope this helps. Best regards, Stan and Peter Klimas (LinuxNAG at gmail.com)

Contents

1	Introduction	1
1.1	General description of this Guide	1
1.2	Conventions	1
1.3	Why Linux?	2
1.4	Is Linux for me?	2
1.4.1	What are the benefits of Linux?	3
1.4.2	What are the differences between Linux and UNIX?	6
1.4.3	What are the differences between Linux and MS Windows?	7
1.4.4	I don't believe in free software, etc.	8
1.4.5	"There ain't no such thing as a free lunch"	8
1.4.6	I need warranty and security. With commercial software, I can sue if things go wrong.	9
1.4.7	I need standards. Big software corporations (Microsoft) provide standards.	9
1.4.8	I Need MS Windows for Reading Writing MS Word Documents	10
1.4.9	MS Windows popularity insures that it is "here to stay"	11
1.4.10	But LINUX may fork into many different systems	11
1.4.11	Linux lacks a central development blueprint (or roadmap)	12
1.4.12	Linux is a cult	12
1.4.13	The total cost of ownership (TCO) of Linux is high	13
1.4.14	Linux is idealistic "dreaming"; it is business that rules the world nowadays	13

1.4.15	Linux may be found to violate somebody else’s “intellectual property” (IP)	14
1.4.16	Linux sux etc.	15
2	Before Linux Installation	17
2.1	Which Linux distribution should I use?	17
2.1.1	Popular Distributions	18
2.1.2	Comments About Linux Distributions	20
2.1.3	Will my hardware work under Linux?	21
2.1.4	Getting Linux	21
3	Linux Shortcuts and Commands (<i>Highly recommended</i>)	23
3.1	Notes for the UNIX Clueless	23
3.2	Linux Essential Keyboard Shortcuts and Sanity Commands	24
3.2.1	Moving around your Linux computer	24
3.2.2	Command-line Shortcuts	25
3.2.3	X11 (graphical-interface) Shortcuts	27
3.2.4	Non-Essential Shortcuts	28
3.3	Help Commands	30
3.4	System Info	31

Chapter 1

Introduction

Like everyone, we were Linux newbies when we started playing with Linux in the summer of 1998. Throughout these years we have tried many distributions (Debian, Redhat/Fedora, Mandriva, Slackware, and Zenwalk). The solutions within this guide were written with mostly RedHat/Fedora and Mandriva in mind, so the solutions might not be directly applicable to other Linux distributions (although most of them are). Hope this helps; we try to be as practical as possible. Of course, we provide no warranty whatsoever. If you spotted a bad error or would like to contribute a part on a topic of your choice, we would like to hear from you.

1.1 General description of this Guide

A complete reference for new Linux users who wish to set up and administer their own Linux home computer, workstation and/or their home or small office network. The answers are meant to be simple, with just sufficient detail, and always supported with a readily usable example. The work is still in progress, but we hope the Guide can be helpful already. We welcome your corrections, advice, criticism, links, translations, and CONTRIBUTIONS. Pls note that there are no ad banners on our pages.

1.2 Conventions

<> = single special or function key on the keyboard. For example <Ctrl> indicates the "control" key.

italic = name of a file or variable you probably want to substitute with your own.

`fixed width` = commands and filenames.

1.3 Why Linux?

If you truly enjoy working with computers, Linux is the operating system of your dreams. It is more fun than any other computer operating system around. Linux is a computer hobbyist's paradise.

However, the reason why Linux is truly revolutionary is that it is Open Software. Our science and technology work owing to the free availability of information, peer review, and the capabilities to pick up ideas and modify or extend. Open Software is an implementation of the scientific method into the field of software development. The freedom to pick-up, modify and extend, that comes with the Linux licence, offers a promise that the software development under Linux licence will continue in the way that science does.

The making of horseshoes, good glass, or measuring time were once closely guarded trade secrets. Science and technology exploded 500 years ago thanks to the sharing of knowledge by the means of printing, and thus breaking the monopoly of the few on the know-how. Why were the science and technology relatively stagnant before the printing era? Because the "trade secret" approach to growth has its limits: the development continues until the rate of learning equals the rate of forgetting. Moving the know-how to the public domain (printed on paper) shifted the position of the equilibrium to a higher level, which, after 500 years of turbulent development, we are still to achieve. For interested, I can present equations, borrowed from chemical equilibrium field, that describe this in a more formal way. The main point is that printing increased the audience, and allowed passing information across space and time.

In the early days of printing, many of those who dared to share were assassinated for revealing "trade secrets." Linux is for the computer age what Gutenberg was for writing. Hopefully there will be no assassinations this time :-). Linux does clash, violently at times, with those who claim the "ownership" of information, trying to push time back 500 years.

Open Source Linux has also implications for computer security. Imagine a plane based on secret "scientific laws", built to an unreviewed design, a plane at internals of which nobody but the manufacturer could inspect. Would you believe that this unique, completely proprietary plane, obviously built by clever and well-financed marketers in their basement, is any safer than one built in the open world, under the eyes of thousands of critical engineers and curious hobbyists with no association to the manufacturer? Then why would you trust a secret computer program? Open-source Linux is ideally suited for a mission-critical application—its security and power are based on robust solutions which are reviewed with no restrictions whatsoever, and continuously improved upon.

1.4 Is Linux for me?

Only you can answer this question. Linux is a mature, powerful, secure and extremely versatile UNIX-like operating system. The power and versatility come with a price—you may need to be computer-literate in order to set-up and maintain Linux. Linux is relatively easy to use once the operating system and applications are set up properly. So, your mother will also be able to use Linux, if you set up an easy graphical account for her and put the proper icons/menus on her GUI desktop. Linux is secure, so your mother will not be able to damage the system no matter how hard she tries—unless it's with a hammer :-).

Linux is quite different from MS Windows, so do not expect that if you can get around MS Windows, then Linux will be straightforward for you. You may need to learn. On the other hand, if you come from UNIX, Linux will be easy for you. If you don't know much about computers or you don't enjoy them, chances are Linux administration is not for you. If you don't know your hardware, Linux installation may be a challenge.

1.4.1 What are the benefits of Linux?

Linux can give you:

1. A modern, very stable, multi-user, multitasking environment on your inexpensive PC hardware, at no (or almost no) monetary cost for the software. Linux is a rich and powerful platform—don't think of it as a "poor people" operating system. Out-of-box Linux has as much capability as MS Windows NT with \$5000 in software add-ons, is more stable, and requires less powerful hardware for comparable tasks.
2. Standard platform. Linux is VERY standard—it is essentially a POSIX compliant UNIX. (Yes, Linux is a best-of-the-breed UNIX. The word "UNIX" is not used in conjunction with Linux because "UNIX" is a registered trademark of "The Open Group".) Linux includes all the tools and utilities typically associated with UNIX, plus more, significantly more than even the most expensive commercial UNIX implementations. To be able to run Linux software, proprietary UNIX vendors implement "Linux compatibility layers" into their platforms.
3. Unsurpassed computing power, portability, flexibility, and customizability. A Linux cluster recently (April 1999) beat a Cray supercomputer in a standard benchmark. Linux is most popular on Intel-based PCs (low price for the hardware), but it runs very well on numerous other hardware platforms, from toy-like to mainframes. One distribution (Debian) expresses the idea like this: "Linux, The Universal Operating System." Linux can be customized to perform almost any computing task.
4. Advanced graphical user interface. Linux uses a standard, network-transparent X-windowing system with a "window manager" (typically KDE or GNOME but several are available). The graphical desktop under Linux can be made to look like MS Windows (or probably ANY other graphical user interface of your choice).
5. Dozens of excellent, free, general-interest desktop applications. These include a range of web browsers, email programs, word processors, spreadsheets, bitmap and vector graphics editing programs, file managers, audio players, CD writers, some good games, typing tutor, etc., even a sophisticated planetarium.
6. Thousands of free applets, tools, and smaller programs. "Small is beautiful" goes well with Linux philosophy. The small Linux tools and applets often work in tandem to perform more complex tasks.
7. Hundreds of specialized applications built by researchers around the world (astronomy, information technology, chemistry, physics, engineering, linguistics, biology, ...). In many fields, Linux seems like "the only" operating system in existence (try to find out what your friend astronomer runs on her computer). The software in this category is typically not very easy to use, but if you want the power, it is the best software that humanity has in these areas.
8. Scores of top-of-the line commercial programs including all the big databases (e.g., Oracle, Sybase, but no Microsoft's). Many (most?) of these are offered free for developers and for personal use.
9. A truly great learning platform. If you are a parent, you should be really glad your daughter/son does Linux—s/he will surely learn something of lasting value. If you are a teacher, you should consider the installation of Linux at your school. "It is indeed a strange world when educators need to be convinced that sharing information, as opposed to concealing information, is a good thing." [edge-op.org¹] You select Linux if you care to provide education, not training. The better the university, the greater the chance their computer department uses Linux in teaching. For example, under Linux, you can immediately begin modifying and compiling for yourself a spreadsheet application which is in every bit as advanced and capable as MS Excel. Linux puts you right on the cutting edge (in technology, project management, QA, methodology of science). Many teachers won't use Linux in schools because they are lacking in computer education themselves (at least that's what I see).

¹<http://edge-op.org/grouch/schools.html>

10. Excellent networking capability built into your operating system. You think you don't need a network? Once you try home networking, you will never be able to live without it! How about connecting the two or more computers that you have at home and sharing your hard drives, CDROM(s), sound card(s), modem, printer(s), etc.? How about browsing the net on two or more machines at the same time using a single Internet connection? How about playing a game with your son over your home network? Even your old 386 with MS Windows ver.3.11 may become useful again when connected to your Linux Pentium server and when it is able to use your network resources. All necessary networking software comes with standard Linux, free, just setup is required. And it is not second-rate shareware—it is exactly the same software that runs most of the Internet (the Apache "web server" software runs 66% of all Internet web servers and Sendmail touches some 70% of all e-mail). The pleasure of home networking is something I was able to discover only owing to Linux.
11. Connectivity to Microsoft, Novel, and Apple proprietary networking. Reading/writing to your DOS/MS Windows and other disk formats. This includes "transparent" use of data stored on the legacy MS Windows partition of your hard drive(s).
12. State-of-the-art development platform with many best-of-the-kind programming languages and tools coming free with the operating system. Access to all the operating system source codes, should you require it, is also free. The "C" compiler that comes standard with Linux can compile code for more platforms than (probably) any other compiler on earth. Perl, Python, Guile, Tcl, Ruby, powerful "shell" scripting, and even assembler tools also come as standard with Linux.
13. Freedom from viruses, "backdoors" to your computer, software manufacturer "features," invasion of privacy, forced upgrades, proprietary file formats, licensing and marketing schemes, product registration, high software prices, and pirating. How is this? Linux has no viruses worth mentioning because it is too secure an operating system for the viruses to spread with any degree of efficiency. The rest follows from the open-source and non-commercial nature of Linux: Linux evolved itself by "bazaar-like" mechanisms to encapsulate the best computing practices, code legibility and maintainability, security, flexibility, usefulness, coolness, and performance. (The most important of these attributes is probably maintainability.)
14. The operating platform that is guaranteed "here-to-stay." Since Linux is not owned, it cannot possibly be put out of business. The Linux General Public License (GPL) insures that development/maintenance will be provided as long as there are Linux users. There are a great number of highly-educated Linux users and tens of thousands of actively developed projects.
15. A platform which will technically develop at a rapid pace. This is insured by the modern, open-software development model which Linux implements: "build-on-the-back-of-the-previous-developer" and "peer-review-your-code" (as opposed to the anachronistic closed-software model: "always-start-from-scratch" and "nobody-will-see-my-code"). Even if the current "Linux hype" died out, Linux will develop as it did before the media hype started. Open source development does have its peculiarities: the development appears rather slow (vertically) but it proceeds on a very wide front, dangerous security bugs are fixed almost upon discovery, there are typically several alternatives for a program of similar functionality. Linux depth cannot be overestimated.

If you wanted to learn first-hand about the General Public License, check these famous GNU documents:

<http://www.gnu.org/copyleft/gpl.html>

<http://www.gnu.org/gnu/linux-and-gnu.html>

<http://www.gnu.org/philosophy/categories.html#TheGNUsystem>

In a nutshell, the GNU General Public Licence (GPL) allows anybody to:

- use the software at no charge, without any limitations,

- copy, and distribute or sell unmodified copies of the software in the source or binary form,
- use the software with proprietary (e.g., your own) modifications, free of charge, as long as you do not distribute or sell the modified version,
- modify, and distribute or sell a modified version of the software as long as the source code is included and licenced on the same terms as the original you received (the GPL),
- sell support for the software, without any limitations.

What the GPL license *does not* allow code recipients to do is to take somebody else's software licenced under GPL, modify the software, and then distribute a this modified version of the software under a proprietary licence. Speaking plainly, the GPL licence just forbids stealing existing (somebody else's) software for incorporation into a closed, commercial-only product. However, you may incorporate GPL software in a commercial computer program if you obtain permission from the copyright holder. GPL is certainly not more restrictive or imposing than a "typical" proprietary licence. GPL is a licence that grants the recipient right which he otherwise does not have, but takes away none. Excluded from the use of GPL are persons who have violated the GPL.

In general, copyright laws regulates 5 rights: to copy the work, to make derivative works, to distribute the work, to perform the work, and to display the work.

Here is a table which contrasts the licence of Linux with that of MS Windows (put together by a RedHat lawyer, based on a Groklaw article²:

Linux / MS Windows

- Right to copy the work: Yes / No
- Right to make derivative works: Yes / No
- Right to distribute the work Yes, under the same licence No
- Right to perform the work: Yes / Yes
- Right to display the work: Yes / Yes

The GPL license under which Linux is distributed is probably the most important part of it. It is designed to perpetuate the freedom of information. Other important open-source projects include science and law (hardly a joke). The Linux method is really nothing new—it is simply the application of the scientific method to software: you get information free, you add your ideas and make your living, and finally, you leave it free. However, some big corporations and their lawyers seem to be trying hard to change this, to push us back in time, to the dark ages, when information was kept "proprietary." Hence, you see in newspapers some famous Linux-connected persons involved in all kinds of struggles.

To get a flavour for the value of Linux, here are some prices for commercial software as listed at www.amazon.com. All prices are in \$USA, as listed on 2001-02-03, with discounts. Roughly equivalent Linux software is included on almost any Linux CD set (but with no restrictions on the number of clients). In addition, the hardware for Linux is typically significantly less expensive, since Linux can run all services on a single server:

- Microsoft Windows 2000 Server (5-client)—\$848.99
- Microsoft Exchange 2000 Server (5-client)—\$1,279.99

²<http://www.groklaw.net/article.php?story=20031231092027900>

- Microsoft Outlook 2000 (1-client)–\$94.99
- Systems Management Server 2.0 (10-Cals)–\$994.99
- Proxy Server 2.0–\$886.99
- Microsoft SQL Server 2000 Standard Edition (5-client)–\$1,229.99
- Microsoft SQL Server 2000 Standard Edition (1-user License)–\$4,443.99
- Microsoft BackOffice Small Business Server 4.5 NT (Add-On 5-CAL)–\$264.99
- Windows NT Server Prod Upgrade From BackOffice SBS Small Bus Server (25-client)–\$558.99
- Microsoft Windows 2000 Advanced Server Upgrade (25-client)–\$3,121.99
- Microsoft FrontPage 2000–\$129.99
- Microsoft Internet Security and Acceleration Server –\$664.99
- Site Server Commerce 3.0 (25-client)–\$4,092.99
- Visual C++ 6.0 Professional Edition with Plus Pack–\$525.99
- Microsoft Visual Basic Enterprise 6.0 with Plus Pack–\$1,128.99
- Microsoft Visual Sourcesafe 6.0 CD–\$469.99
- Microsoft Office 2000 Standard (1-client)–\$384.99
- Adobe Photoshop 6.0–\$551.99
- Microsoft Plus Game Pack–\$19.99

Linux (and thousands of other programs distributed under GPL) is often described as "free software". The word "free" has two quite different meanings in the English language, and it sometimes leads to misconceptions about the free nature of Linux. These two meanings follow the Latin adjective "liber" and the adverb "gratis," and they are often illustrated with the phrases "free speech" and "free (of charge) beer." Most Linux software is free in both senses, but it is only the first sense which is essential to Linux.

1.4.2 What are the differences between Linux and UNIX?

Command-line-wise, almost none, although this has been changing (for better or worse). Linux has a much larger market appeal and following than any commercial UNIX. GUI-wise there are also no major differences–Linux, as most other UNIXes, uses an X-Windowing system.

The major differences:

- Linux is free, while many UNICES (this is supposed to be the plural of UNIX), are very expensive. The same for applications–many good applications are available on Linux free. Even the same commercial application (if you wanted to buy one) typically costs much more for a commercial UNIX than for Linux.
- Linux runs on many hardware platforms, the commodity Intel-x86/IBM-spec personal computers being the most prominent. In contrast, a typical UNIX is proprietary-hardware-bonded (and this hardware tends to be much more expensive than a typical PC clone).

- With Linux, you are in charge of your computer, whereas on most UNICES you are typically confined to be an "l-user" (some administrators pronounce it "loser").
- Linux feels very much like DOS/Win in the late 80s/90s, but is much sturdier and richer, while a typical UNIX account feels like a mainframe from the 60s/70s.
- Some UNICES may be more mature in certain areas (for example, security, some engineering applications, better support of cutting-edge hardware). Linux is more for the average Joe who wants to run his own server or engineering workstation.

1.4.3 What are the differences between Linux and MS Windows?

Mouse-click-wise, almost none, once Linux is properly installed. Linux installation can be a challenge though, whereas MS Windows comes, most likely, pre-installed with your computer.

The major differences:

- Linux is free, whereas MS Windows costs money. Same for applications. If MS Windows or Office comes preinstalled with the computer it is unlikely it is free. Ask in the store to take it off your computer (your run Linux) and you are likely to obtain a discount, at least in smaller stores.
- Linux file formats are free, so you can access them in a variety of ways. On MS Windows, the common practice is to make you lock your own data in secret formats that can only be accessed with tools leased to you at the vendor's price. How corrupt (or incompetent?) must be the politicians who lock our public records into these formats! "What we will get with Microsoft is a three-year lease on a health record we need to keep for 100 years." [news.bbc.co.uk³]
- With Linux, you are unlikely to violate any licence agreement—all the software is happily yours. With MS Windows you likely already violate all kinds of licenses and you could be pronounced a computer pirate if only a smart lawyer was after you (don't worry, most likely none is after you).
- MS Windows tries to be the "lowest-common-denominator" operating system (for better or worse), whereas Linux is built for more sophisticated, feature-hungry computer users (for better or worse).
- MS Windows is based on DOS; Linux is based on UNIX. MS Windows Graphical User Interface (GUI) is based on Microsoft-owned specifications. Linux GUI is based on an industry-standard network-transparent X-Windowing system.
- Linux beats Windows hands down on network features, as a development platform, in data processing capabilities, and as a scientific workstation. MS Windows desktop has a more polished appearance, smoother general business applications, and many more games for kids (these are not better games though—Linux games tend to be more sophisticated).
- Linux is more feature-rich than you could imagine. Heard on the Internet: "Two big products came from the University of California: UNIX and LSD. And I don't think it's a coincidence." MS Windows is simpler.

³<http://news.bbc.co.uk/hi/english/health/newsid.1694000/1694372.stm>

1.4.4 I don't believe in free software, etc.

Do you believe in the Internet? The Internet and Linux share underlying ideas and have common roots. Do you remember the disbelief about the Internet a few years ago, the endless, seemingly unbeatable arguments that free Internet cannot exist? "Who pays for that, anyway?"

The reality is simple. Cooperation and good will can benefit many at the same time: your gain is not my loss. The Internet works fine and is expanding at a rapid pace. So does Linux.

Here is the opinion of an IBM executive: "The reason we are so excited about Linux is we believe Linux can do for applications what the Internet did for networks." [linuxtoday.com⁴] IBM just (May 2002) spent 1 billion dollars making Linux run on all their hardware platforms (mainframes, workstations, PCs, laptops).

We may add that Linux seems to do to the operating system the same what IBM's open PC specification did to the computer hardware.

1.4.5 "There ain't no such thing as a free lunch"

"The economic paradigm which makes this true depends upon scarcity of resources. Software resources are only scarce because we all keep software proprietary and secret. But not Linux! When I give you my software, it may create an opportunity cost for me, but I get to keep it even after I've given it to you. It is a free lunch only rivaled in history by the loaves and the fishes" (Brett Bazant ;bbazant@shaw.wave.ca; [linuxtoday.com⁵]).

And here is a quote from Thomas Jefferson explaining, in the year 1813, that intellectual property (IP) does not exist. There is only a limited monopoly to profit for the author, which is society-given on the conditions that the monopoly strikes the balance for the common benefit:

"If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it. He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me. That ideas should freely spread from one to another over the globe, for the moral and mutual instruction of man, and improvement of his condition, seems to have been peculiarly and benevolently designed by nature, when she made them, like fire, expansible over all space, without lessening their density in any point, and like the air in which we breathe, move, and have our physical being, incapable of confinement or exclusive appropriation. Inventions then cannot, in nature, be a subject of property. Society may give an exclusive right to the profits arising from them, as an encouragement to men to pursue ideas which may produce utility, but this may or may not be done, according to the will and convenience of the society, without claim or complaint from anybody." ["The Writings of Thomas Jefferson". Edited by Andrew A. Lipscomb and Albert Ellery Bergh. 20 vols. Washington, Thomas Jefferson Memorial Association, 1905. Quoted after: "The Founders' Constitution" Volume 3, Article 1, Section 8, Clause 8, Document 12, The University of Chicago Press⁶.]

⁴http://linuxtoday.com/news_story.php3?tsn=2000-08-17-001-04-PS-EL

⁵<http://linuxtoday.com/cgi-bin/showtb.pl?tbsn=12450&sn=5418>

⁶http://press-pubs.uchicago.edu/founders/documents/a1_8_8s12.html

1.4.6 I need warranty and security. With commercial software, I can sue if things go wrong.

Don't count on suing. Things go wrong on many MS Windows NT machines every day, and there are no damages awarded by courts. Read your MS Windows license agreement to find out that there is no guarantee whatsoever that ANYTHING will work. Trying to sue would be a waste of your money.

Linux also provides no guarantees, although it is far more secure than any version of MS Windows (assuming comparable functionality is installed on both). If you are really security-sensitive, you can use high-security tools built by companies that rely on the availability of the source code to design and test their security features (e.g., Kryptokom in Germany provides high-security firewalls). The "security in obscurity" implemented in MS Windows has repeatedly been demonstrated to be a naive approach.

"Risk aversion is what dictates you use Linux and other open products, rather than NT. The risks with NT are entirely out of your control, and there is nobody you could sue if anything goes wrong. Why people still believe the myth that Windows in any form offers any bit of accountability "more" than Linux remains a complete riddle to me." [David Kastrup, Research Engineer, Bochum, Germany, Internet Week⁷]

1.4.7 I need standards. Big software corporations (Microsoft) provide standards.

Perhaps that's what people would expect from large corporations, but the reality is rather different. Once, big companies loved inventing nuts that could be undone only by their own service shops. Did these nuts become standard? Hardly. They didn't because there was no public benefit involved, and they couldn't because they were patented. Luckily, now we have open and free standards for nuts. A "proprietary standard" is such a ridiculous oxymoron that it is hard to believe that educated people can believe in it. (Currently, marketing types use the term "de facto standard" or "industry standard" to cover up the ugliness of the lack of standards.)

The era of proprietary standard nuts seems over. But the idea lives on in the computer field. For example, the "standard" MS Word file format has changed numerous times over the recent years. This keeps happening probably for a good business reason: as soon as other companies "reverse-engineer" the current Word format, Microsoft changes it. There are even sub-formats (an MS "fast-save" anybody?). The "standard" is completely closed—Microsoft does not publish any specifications. How can the user benefit from this in a longer term? What is the Microsoft guarantee that MS Word 6.0 file format will still be legible in the year 2020? None I could find.

"... Microsoft's standards are both proprietary and arbitrary- the stealth incompatibility of Office 97 file formats with older versions of Office or the subversion of Open standards like XML with proprietary extensions that require Internet Explorer 5, MS Active server and so on, are sober reminders of what the company does to a market." [Xavier Basora⁸].

"... Microsoft's monopoly doesn't guarantee that your current MS Office will work with any previous or future MS Office. This is in spite of any number of Microsoft apologists arguing that the benefit of Microsoft's monopoly has been a standard for productivity applications." [Wesley Parish⁹]

To add to the confusion, companies typically do not "standardize" on file formats but on the applications that are supposed to produce them. It is like standardizing on a manufacturer of nuts instead of on nuts. How is this supposed to work if the file manufacturer keeps changing the specification to drive their sales?

"We need standardized, open file formats so that users can exchange documents between platforms. The actual word

⁷<http://www.techweb.com/se/directlink.cgi?INW19990329S0050>

⁸<http://www.osopinion.com/Opinions/XavierBasora/XavierBasora47.html>

⁹<http://www.osopinion.com/Opinions/WesleyParis/WesleyParis10.html>

processing software used to generate these documents shouldn't even be an issue." [Ted Clark, linuxtoday.com¹⁰]

There are a few text/document oriented file formats that are quite definitely more standard than MS Word file format: ASCII, XML (with non-proprietary stylesheets), HTML, SGML, LaTeX, TEX, PostScript, pdf, dvi ... and all of them have excellent support under Linux. The MS Word file format can also be read/written very well under Linux by OpenOffice (and a number of other applications) to cover your current needs. Advanced, "universal," open-source document formats (XML-based) are developed by an independent organization. The story is similar with other proprietary computing "standards" (*.giff vs. *.png anyone? *.mp3 vs. *.ogg?).

Linux, by its very nature, is based on true, published and free standards because "open source" makes the full specifications available to everybody (competitors or not). We believe that the urge for open standards is the very driving force behind Linux. Many people feel that they cannot afford to trust their algorithms and data to a commercial entity, let alone a single one that has repeatedly demonstrated its untrustworthiness.

Have a look at a draft of this Argentinean law for a taste of the future. It sounds like the Argentineans may be the first to decide that their public records cannot be held hostage by a commercial entity: "... Public National Organizations mentioned in article 1 of this law, will not be allowed to use programs that store data in non-public format ...". Several other countries are also contemplating or enacting legislations requiring storage of public data in public file formats. [slashdot.org¹¹]

There is a strong perception in the Linux community that there is a serious problem with the computing "standards" championed by large software vendors. This includes their standards for storing our "static" data, as well as the processing algorithms embedded in our computer codes. Can we afford to trust somebody decide for us when, how, and at what cost we can access our own work? This problem is ignored and even aggravated by people who are paid to take care of it. Linux is a grass-root answer to this problem.

Here is an example from Life, as narrated by [The Economist¹²]:

"IN MAY, the city of Munich decided to oust Microsoft Windows from the 14,000 computers used by local-government employees in favour of Linux, an open-source operating system. Although the contract was worth a modest \$35m, Microsoft's chief executive, Steve Ballmer, interrupted his holiday in Switzerland to visit Munich and lobby the mayor. Microsoft even dropped its prices to match Linux—a remarkable feat since Linux is essentially free and users merely purchase support services alongside it. But the software giant still lost. City officials said the decision was a matter of principle: the municipality wanted to control its technological destiny. It did not wish to place the functioning of government in the hands of a commercial vendor with proprietary standards which is accountable to shareholders rather than to citizens."

My favourite example of how Microsoft, instead of promoting standards, keeps confusing them. For decades, there has been one standard way to write all-numeric date and time in the country I live. This standard is accepted in most countries of the world. MS Excel offers, conveniently in a drop-box, almost any possible permutation to format date/time, except the one required by the international standard. I guess, there is no lesson learned from the billions spent on the "Year 2000 issue".

1.4.8 I Need MS Windows for Reading Writing MS Word Documents

In a large corporate environment, you may have little choice—they locked themselves by cheerful productions of non-portable forms, templates, visual basic-driven web pages and other MS Office-bound "tools". Perhaps a more adequate

¹⁰http://linuxtoday.com/news_story.php3?tsn=2000-09-29-004-06-OP-MR-0010

¹¹<http://slashdot.org/articles/01/04/28/010216.shtml>

¹²http://www.economist.com/business/displayStory.cfm?story_id=2054746

name for proprietary software would be “lock-in software”?

In a smaller environment, you can use OpenOffice.org suit (OO) that runs on Linux, MS Windows, Mac, Solaris (and more), with full file-level compatibility. It can be downloaded and installed for free (no restrictions whatsoever) so nobody should really complain about the file format (some control freaks still will). Just to make sure, OpenOffice.org can import and export MS Word and Excel documents of reasonable complexity very well. The native file format in OpenOffice.org is fundamentally much better than Microsofts (plus it is non-proprietary). Feature-by-feature, OpenOffice.org can do almost anything MS Office can, plus some extras. Depending on whom you ask, the ease of use varies between “50% more difficult” to “20% easier” (measured on a sample of experienced MS Office users). Very complex documents are best transferred as *.pdf, and OpenOffice.org can make them on the fly.

So, probably you do not need MS Office any more. Download your OpenOffice.org¹³ for MS Windows and/or Linux.

Latest MS marketing joke: “Wait, don’t install OpenOffice. Microsoft is ALREADY working on a file format that is based on the same principle as that OpenOffice is using. Microsoft will even extend the file format to make it even BETTER.” Well, we do not need a better format. We need a open-standard file format.

1.4.9 MS Windows popularity insures that it is “here to stay”

This is likely true. Nintendo is probably also “here to stay.” However, I like computing; therefore, I choose a computer with a powerful operating system, not a lowest-common-denominator piece designed for “everybody.”

Linux is quite positively here-to-stay because of its open-source nature (Linux cannot possibly be put out-of-business). It is a standard selected for countless projects that are not going to go away, and some of them are quite “mission-critical.” Try the International Space Station, for which Linux is the operating system. [linuxjournal.com¹⁴]

Plus, never underestimate the strength of the Linux community. Linux is “here to stay” at least for the computer avant-garde. Many Linuxers do not even want Linux to become very popular because they fear it could “dumb down” the elite Linux platform.

1.4.10 But LINUX may fork into many different systems ...

This is a typical argument of the type spread by experts in the marketing tactics known as “fear, uncertainty and doubt” (FUD) [about the competing product]. The “forking” argument was once commonly used by various “analysts” and the mainstream computer press to explain to the undecided why Linux is not worth their attention.

“Forking” in this context means “branching a computer program,” so as to create parallel “subversions” of the program, and consequently fragment Linux and presumably reducing its usefulness.

There is very little (if any) evidence of harmful forking of any software included with a typical Linux distribution. Where forking did occur, it has always turned beneficial. Quite possibly, this is because although there are no artificial barriers to fork software under Linux, there are also no artificial barriers to merge the best pieces back.

The theoretical background on how forking software can be good for its development might have been actually given quite some time ago by the German philosopher Georg Wilhelm Friedrich Hegel (1770 -1831), with his concept of

¹³<http://www.openoffice.org>

¹⁴<http://www2.linuxjournal.com/lj-issues/issue59/3024.html>

dialectic development. E.g., in “Phenomenology of Spirit”, Hegel concludes: “... the schism incipient in a party, which seems a misfortune, expresses its fortune rather.”

1.4.11 Linux lacks a central development blueprint (or roadmap)

Linux has no central steering committee, design specifications, etc. It has many leading bodies for the many ongoing, and often competing projects. Some of the projects work on integration, overall consistency, etc. The value of their concepts is verified in practice.

Following Hegel’s (and Parmenides) thought, one may expect that what is rational comes into existence in Linux, and what became a permanent part of Linux did so because it was rational (Hegel’s original is “What is rational is real and what is real is rational”). The same cannot be expected to be the end-product of development of a system build to a rigid blueprint prepared by a committee.

1.4.12 Linux is a cult

The Linux community has repeatedly been labeled “religious zealots” by journalists whose well-established computer magazines received massive feedback after they had published highly unfair articles on Linux. So yes, the Linux community is numerous, literate, and willing to express its opinions. And many computer journalists/magazines know that Linux means less money for them (users pay less for their computing and the associated advertisement, while expecting more). Does this explain the “zealots”?

Face it, you salespeople pretending to be journalists. There is hardly any integrity left in the computing press. How many words on Linux did your PC Magazine (or any other IT magazine) publish by 1999-01-01? Wasn’t Linux at least an interesting technology by that time? It surely was, yet you conspired to keep your readership in the dark, selling your journalistic integrity for a few dollars. And now, after Linux has surfaced in the mainstream (non-computer) media, you keep writing misleading articles about it saying “yah, but it will/cannot/may ...” whatever (trying the “fear, uncertainty and doubt” tactics to kill it). And adding “Microsoft is already ...”, continuing to write about the vaporware and the future paradise in the face of the increasingly stealthy, unstable, pricey, architecturally unsound computer platform, whose greatest achievement has been exhorting unheard-of-before money by denying inter-operability, and killing any existing or proposed standard (by “embracing” and then proprietary-extending it). Whom do you serve? Surely not your readers.

I worded it as strongly as I could. Am I a zealot? Or am I just trying to voice my disapproval for the self-serving actions of the computer “powers-that-be”?

You think “self-serving” is OK in business? How pathetic must your business be then! I always thought that business was a social contract in which we exchange good values, for a mutual benefit. As I read history, societies use to hang / guillotine / electrocute those members who really persisted in their self-serving business. Well, times have changed. A bit for the better, a bit for the worse :))

To be fair, there seems to be a number of hard-core devotees around any computer program/platform, and Linux might have accumulated a fair share of them. Oh, well, you may be religious about whatever you like. I can assure you, most Linux users I interacted with are rational.

1.4.13 The total cost of ownership (TCO) of Linux is high

Nobody really knows how to calculate the “total cost of ownership” of a general piece of software, no matter how clever terms they use. So a good lawyer + accountant can prove whatever point they are paid to make. They appear to regularly do so.

For example, does your calculation of the “total cost of ownership” of MS Word include the cost of exiting the platform? If not, do you really believe, that MS Word will be your documentation platform *forever*?

Let me try a simple estimate of how much the average total cost of the ownership of MS Windows is. Let’s add the fortunes accumulated by all the MS Windows software makers. Add all the salaries of all generic Windows programmers, consultants, support and training personnel, IT management, etc. Add the cost of the hardware for MS Windows. Now, add the losses in productivity customers must surely have suffered while the software corporations were presenting them with “new features” so as to make them upgrade and/or achieve their current lock-in status. Divide this sum of money by the number of years (whatever time frame you selected), and the number of MS Windows users (only in the countries in which software is normally paid for). Here is the TCO of MS Windows. However you count it, it will be many thousands of good US dollars per average joe per year. You didn’t pay that much money? Well, you must have, it has just been hidden from you (probably in the price of seemingly unrelated products and taxes). Yes, developed countries waste billions of dollars on software every year.

How much did Linux cost? Hardly anything. The number of users is much lower, too, but you will be hard pressed to come up with \$10 per user per year.

Yet, in my opinion, the total cost is not what matters the most. What value did I receive for my money? You would have to calculate the total value of ownership (TVO?), then subtract from it the total cost of ownership (TCO) to obtain the “net benefit of the ownership” (or “return on investment”). Well, I cannot see how I would be making a good investment by purchasing the latest version of MS Windows or Office, by putting myself deeper into a single-platform dependency. My Linux based email, web browser and word processor work as well as anything available on MS Windows.

I guess accountants typically talk about the TCO for software “necessary for doing business,” and thus skip the issue of the value, benefit, and the return on investment. There is really no value in the mainstream software, it is just the necessity for doing business these days. Well, Linux satisfies my computing necessities at zero monetary cost, and the personal pleasure and learning value are great.

1.4.14 Linux is idealistic “dreaming”; it is business that rules the world nowadays

Think of Linux as a consortium. Businesses/individuals get together to address a common computing need or problem. They may chip in labour or money, hire a technical leader or team, or otherwise cooperate to make Linux address their requirements. The solution is totally theirs for keeps, and it does not have to cost a lot—they can re-use the pre-existing Linux software pieces. They may even cooperate to overcome the advantage that a big “industry leader” may have and use against their interests.

Linux is the end-product of activities of many such loose “consortiums” who “scratch their needs.” So Linux is a business, but it is not necessarily about centralized production and marketing of software. It is a de-centralized, small-scale development performed close to the end-users, so that they have access to reasonably-priced software that matches their need, solves their problems, sells their hardware or service, and which is totally theirs to keep: the licence never expires, and the user will never be cut off.

1.4.15 Linux may be found to violate somebody else's "intellectual property" (IP)

This is a serious and timely issue (July 2003). The most powerful Linux adversary (Microsoft) stated in their internal memorandum (leaked to the press) that they should try the legal route to deal with Linux ("The discussion of IP rights needs to be tied to concrete actions"). The entertaining (and "scary") Microsoft series of blueprints on how to deal with Linux have been leaked and published as so-called Holoween Documents [<http://www.opensource.org/halloween/>].

So yes, there is some probability that Linux might be sued out of existence in the United States because of political pressure.

Linux developers are not copiers or thieves. They produce thousands of lines of computer code every day. The Linux licence is based on the respect for the "intellectual property", i.e., the exclusive right of the author to distribute her creation. Nevertheless, misappropriation of code into Linux can happen. Considering the amount and complexity of code in a typical GNU/Linux distribution and the rate of development, it is likely going to happen, sooner or later. A dishonest developer may "lift" code from somebody else and submit it to any of the number of public Linux-related projects, pretending that the code is his own. This IP problem is not limited to Linux. Cases of misappropriation of code happen to close-software companies as well (e.g., it happened to Microsoft) and, no doubt, will happen again.

As far as potential for misappropriation of code is concerned, the major difference between the "open source" and "proprietary, closed software" is that a misappropriation is trivial to detect in the former, and almost impossible in the latter. It is the obligation of any IP owner to protect their IP and report potential (involuntary) infringements so it can be promptly remedied, and their potential losses minimized.

Linux developers are not negligent. They continuously submit all the Linux source code for public review and comment so as to detect and prevent any IP violations. Well-organized Linux archives are maintained. The open development method of Linux follows that employed by science, and it is likely a more diligent practice for protection of other authors' rights than almost any IP control system conceivable for implementation in a closed-software house. Thus, Linux management of IP rights can rightly be called as "among the best in the industry" if not "the best".

Linux has an excellent IP record. Certainly, it is not known to harbor misappropriated code. Is it then legally-safe for me to use Linux? I feel, it is for me. In case a misappropriation of code into Linux is ever found, I can be certain that any infringing code will be immediately expelled from Linux and "clean" version will become available for me to upgrade. It has always been a specific goal of the Linux community at large to produce code uncontaminated with any proprietary code.

The whole issue should be placed into a broader context. There appears to be a recently increasing tendency to go overboard with "intellectual property" which clashes with notions of basic justice, decency and common sense. It seems that some dishonest corporations try to impose nebulous "licensing" to keep extracting, indefinitely, money while delivering no new value. In some cases, IP is becoming a tool with which some major corporations try to tax the average joe (and business startups), not unlike emperors used to tax salt. Linux is an answer to this unhealthy trend, an answer which is based on respect to the existing copyright laws. Therefore, Linux clashes with those who wish to extract money while delivering no value. I do not see a fundamental problem with IP in Linux but there is certainly a conflict of interests and hence attacks on Linux.

To summarize, Linux licence is based on respect to the current copyright laws. For all what is known, Linux is free of "intellectual property" infringement. It employs the best due diligent practices to make sure it remains free of any such infringements. However, due to the revolutionary nature of source code development, Linux is being attacked with nebulous IP claims and smeared by some paid "experts" with various IP "worries". So yes, publicity is expected as Linux big names are dragged through U.S. courts but it does not affect me.

If worst comes to worst and the intellectual property laws became so restrictive as to impede the progress of Linux

and other open software, I would expect that it will be the laws that crumble. This is because the politicians seem to forget the intellectual property does not really exist (as Jefferson knew) and have been put in place only for further development and common benefit. Linux was born from genuine frustration about the quality and cost of software produced by the past methods, and such frustration cannot be contained by naked force without political repercussions.

1.4.16 Linux sux etc.

If you don't like Linux, then do not use it. You are not doing anybody a favor by using Linux. GNU/Linux is free and powerful software, but only for those who like or need it. There are alternative operating systems for you to choose from and they may better match your requirements. Although most Linuxers enjoy the growth and welcome new users, some are not very happy about it because "the crowd and commerce can spoil the hackers' paradise we created." Therefore, you really aren't doing anybody a favour by using Linux.

In this context, it may be worthwhile to briefly summarize Linux strengths and weaknesses: Linux is owned by its fans (your piece of ownership comes free with your free subscription to the fan club), definitely very powerful and feature-rich, highly configurable, as flexible as you want it to be (comes with complexity), low on the cost of hardware, comes with any networking bell-and-whistle known to man, requires a computer literate administrator, some essential desktop applications are still behind commercial offerings on other platforms (e.g., spreadsheet and word processing), a number of excellent end-user applications come "standard" and free with the operating system, the graphical user interface is very nice but still not as polished as Apple or MS offerings, Linux is highly standard (POSIX UNIX), open file formats used all along, thousands of programs available for free download (although the ease of use and quality of these varies vastly). And most of all, Linux is enjoyable!

Chapter 2

Before Linux Installation

What distribution should I use, how to obtain it, Linux hardware requirements, how to partition your hard drive, about dual boot, which packages to install, which graphical user interface (GUI) to install (gnome or kde?), and how to login for the very first time.

2.1 Which Linux distribution should I use?

Linux distribution is a coherent collection of free software with the Linux kernel (operating system) at its center. To run Linux, you normally need to download (or buy) a Linux distribution on CD/DVD. There are many distributions available, but most are based on the Big Three: Debian, Redhat/Fedora or Slackware.

The differences between the various Linux distributions ("distros") are minor: the installation program, choice of the bundled applications and tools, arrangement of a few things on the hard drive. Regardless of your choice of distro, most of Linux is still at the same, and standard hard drive locations are used for essential items. Whichever distribution you decide to install, you will end up with essentially the same Linux.

For desktop use, a common choice is Mandriva Linux (originally called Mantrake), Fedora (formerly known as "Red Hat Linux" "RedHat" or RH), or the increasingly more popular Ubuntu for the following reasons:

1. They are very popular (both an advantage for a newbie and a testimony to their quality).
2. They are general-purpose distributions.
3. They come with relatively easy setup programs.
4. Mandriva, Fedora and Ubuntu contributions to Linux are "open software" (this means that all the software written by the packaging corporations and included on the distribution CDs is licensed under the General Public License, GPL, so that it can be legally copied, given away, reused, etc.).
5. Mandriva, Fedora and Ubuntu can be obtained cheaply or free if you don't care for commercial support. This is a consequence of (4).
6. Mandriva was originally based on RedHat, so the distributions are quite similar. Software packages for RedHat often work on Mandriva (and vice versa) without problems. However, Mandriva is a bit more automatized and

makes a somewhat nicer desktop than Fedora. At the same time, Mandriva sometimes is not as rock stable as Fedora.

7. Ubuntu is a distribution based on Debian, one of the oldest and largest distributions available.

In short, as a newbie, you can safely bet on Mandriva, Fedora, or Ubuntu unless you like something else or have specialized needs, or your environment suggests using something else (e.g., if you have an experienced guru nearby, or a bunch of friends who are using Linux, you may want to use the same distribution - makes getting help a whole lot easier).

The most recent distributions we recommend (May 2007) is Mandriva 2007 Spring, Fedora 7 (the predecessor of Fedora was "RedHat 9.0"), or Ubuntu 7.04. These are all excellent distributions. Be sure to specify the most recent version if ordering your software from a dealer—many dealers like to clear their inventory by sending you an older version (this applies not only to Linux). Generally, development under Linux is fast, and you don't want to waste your time with older distributions. The authors of this guide have no connection to Mandriva, Fedora and Ubuntu (or any other Linux distributor) whatsoever.

Our recommendation for newbies does not mean that other distributions don't offer benefits or unique features which may surpass Mandriva, Fedora or Ubuntu in specific areas. We do believe that we benefited from exposure to a different distribution because it helped us understand Linux better.

2.1.1 Popular Distributions

Debian

We tried Debian and we liked it very much. It was probably as easy as Fedora, but Debian seems less common (hence, being newbies, we picked up Mandriva or Fedora). The great benefit of Debian is that it is 100% non-commercial (put together by volunteer hackers, the true Linux way) and it probably most strictly adheres to Linux standards (it probably sets the standards too). Another great benefit is that Debian crams on their numerous distribution CDs thousands of tools and applications—easily much more than any other distribution. All these tools/apps are nicely "packaged" (for ease of installation) and tested for compatibility. This makes Debian distro look monumental, safe, conservative, and always somewhat outdated. So yes, we would not have a problem recommending Debian as a great general-purpose Linux distribution. Debian calls itself "The Universal Operating System" for a good reason. At any time, Debian carries 3 versions. (1) The "stable" version (sometimes called "potato"), and we would not recommend it, unless you are really paranoid on stability and don't mind quite outdated packages. (2) The Debian "testing" version (sometimes called "woody") is probably as stable as the latest RedHat, and more stable than your current Mandriva. It is much more up to date than Debian "stable". Debian Woody is the version we like. (3) If you don't mind occasional trouble, you can also take the third branch called "unstable", which is likely quite up-to-date.

Slackware

Slackware is a favorite among experienced users who don't like to deal with the ever changing graphical setup programs. Slackware is a very simple distribution without any fancy tools while still being extremely capable and hence is one of our favourites. It relies on the user to know which setup files to edit for almost all configuration tasks. Therefore, with a bit of effort, a computer-literate administrator can actually understand what is going on in her operating system (this is not something I can always say about Mandriva, or MS Windows for that matter). We would have trouble recommending Slackware for a Linux newbie unless the newbie has command-line experience.

Zenwalk

Zenwalk is a minimalistic distribution based on Slackware. The idea behind Zenwalk is to provide a modern environment with all the latest stable software. The distribution is optimized for performance and only has one mainstream application for each task while including a full development/desktop/multimedia environment. I find this distribution easy to use and would recommend this distribution for anyone with an older computer or laptop.

S.u.S.E

S.u.S.E Linux distribution (<http://www.suse.com>) is very popular in Europe. It surely looks German—a solid, general-purpose distro with an easy setup and an excellent reputation. Many users swear by SuSe. We couldn't find cheap Suse CDs though but it appears you can download it (I cannot find a link). Their product includes proprietary additions that will satisfy enterprise-level need to interface some popular proprietary applications (MS Exchange, Cross-over office, etc).

Knoppix

Knoppix Linux (<http://www.knoppix.net>) is another distribution worth consideration. The main point of Knoppix is that it is a "live distro", i.e., it can be booted from a CD, without installation. This is excellent for trying Linux (if you like it, you can also install Knoppix on the hard drive). It is also makes a perfect disaster-recovery tool (distro on a CD is also safe because no malicious program can do anything to your executables, and non-invasive for the local storage as required for post-mortem analyses). Knoppix is also useful if you have to work under Linux on somebody else's computer: you insert Knoppix CD into the CDdrive, and perhaps external storage on the usb port (for personal storage), and you are all set to work in your own environment. When done, you take your chips home. You can mount the local computer resources if you have to. Interesting tool.

Gentoo

Gentoo seems to have some strong following. In Gentoo (hearsay, never used it), they have a cool installation/upgrade system which does anything from sources (a local compilation is required). Long compilations can be joy to watch but, well, they can take time. The resulting executables are tailored to your hardware so they are perhaps smaller or faster than those on a more-standard "already cooked" (binary) Linux distribution.

Xandros (formally Corel Linux)

Corel was once working on their own Linux distribution (based on Debian) apparently geared towards a nice and easy platform to run the Corel suite of office applications: WordPerfect wordprocessor, QuattroPro spreadsheet, Corel Presentations, Paradox database, CorelDraw artist package.... Xandros is a company that was founded in May of 2001 and is headquartered in Ottawa, Canada. Xandros acquired Corel Linux along with the development team behind the product from Corel Corporation in August 2001 after Corel decided to leave the Linux distribution market. Apparently, Xandros is configured to look like MS Windows by default. Is that a good thing?

Caldera

Caldera was once another well-known distribution. It was said to be aiming at corporate users, had a fancy (and pricey) configuration tools, and other corporate goodies. In Aug.2000, Caldera purchased SCO Unix (the original trademarked ancient UNIX) which gave them an even more "corporate" look in my eyes. Caldera did not seem to care too much about home Linux users, so I never considered it for my home use. In early 2003, Caldera (renamed SCO) evidently switched to different, perhaps more profitable, business model ("fire programmers, hire lawyers"). I will surely stay clear of anything that might bear the name SCO or Caldera on it because I do not like the idea of paying US\$1399 for the right to run Linux on a single-processor computer or being sued. Caldera/SCO Linux distro certainly does not have any future.

2.1.2 Comments About Linux Distributions

There are "localized" versions of Linux for specific countries or languages (Korean, Chinese, Japanese, Hungarian, French, ...)—they likely contain (on default) all the hacks and docs (documentation) that the users in these countries want to see. Says Bill Staehle: "You may want to mention the Conectiva Linux distribution, loosely based on RH from Brazil. As such, it is in Portugese, and is also available in Spanish. Try: <http://www.conectiva.com.br/>". I heard several good things about Conectiva, so if Portugese or Spanish was my language, I would probably give it a try.

There are also "special purpose" distributions, e.g. the "real-time" editions of Linux (might be useful if you are in for automation, robotics, fast speed data acquisition, etc.), very small distros (if you like the idea of running Linux from a single floppy which can be useful for system security or recovery), Linux for embedded systems (if you wanted to customize Linux as a small "special purpose" device, which could be good for the next-generation stereo, MP3 player, palm computer, or a fancy cellular phone), parallel computing and clustering systems (might be great if you plan to do your own weather forecasting :-) or at least nuclear explosion simulations :p), etc. Here the differences will be larger, but these distributions are not meant to be "general purpose". As a newbie, you likely don't want to start with any of these, although you might be tempted to. (They surely show Linux strength and viability—Linux runs on toys, even a wrist watch, as well as computer clusters that make the currently fastest systems in the world.)

The distribution you need is of course specific to the hardware platform you have. This means that for your PC hardware containing an Intel 386 processor, or Intel 486, or Intel Pentium, or Intel 586, or Intel 686, or Cyrix, or K6, AMD, or similar, you need the binary distribution called "Intel" or "386" or x86. [Unless you are prepared to start with your own compilation of the Linux source code, which is not typical for a newbie :-)] . This happens because there are binary distributions for other hardware platforms too: PowerPC, Alpha, Apple, IBM mainframe, "Intel StrongARM", Transmeta, and perhaps a dozen more—you don't want to get those binaries for your PC clone; they surely will not work on a PC machine with an "Intel" or "AMD" processor inside. If you have no-Intel hardware, you may want to search the Internet to find who supports it (chances are Debian does, they seem to support even the most exotic ones. Then, you need to obtain "Debian ARM" or "Debian Motorola 680x0" or "Debian PowerPC" or "Debian SPARC ", ...).

In short, although newbies get confused with the multiple Linux distributions, there are reasons to have different distros. They should be viewed as a Linux strength rather than weakness. Linux is simply filling all application and hardware platform niches. The drawback is that there are some "funny" distribution to avoid if you plan your serious business to depend on Linux.

This guide concentrates on Fedora and Mandriva for the PC (Intel) platform but many of the answers may work fine on other distributions and platforms.

2.1.3 Will my hardware work under Linux?

Not every piece of PC hardware is supported under Linux, but most are, particularly the more standard, older, and popular ones. This applies to SCSI adapters, CDRoms, writable and rewritable CDs (CD-R and CD-RW), video cards, mice, printers, modems, network cards, scanners, Iomega drives, etc.

As Linux support for hardware changes on a daily basis an internet search is the best way to determine the compatibility of your hardware. A good place to start includes:

<http://www.tldp.org/HOWTO/Hardware-HOWTO> <http://www.linuxcompatible.org/compatlist3.html>

If a piece of hardware of yours is (apparently) not supported in your current Linux distribution, don't give up. Chances are that: (1) It is supported, but you don't know how to set it up. (Solution: stay around with Linux for a few weeks, don't waste your time, when you get some understanding of how your system works, then you may be able to set it up.) (2) You have to go through a more complex setup to support the hardware (for example some cryptic command or a kernel re-compile, which is not as difficult as it seems). (3) An updated (different?) distribution already supports it "out-of-box". (4) There is already an upgrade somewhere on the Internet, you have to find it, download it, and figure out how to install it. (5) The upgrade will be available next month—Linux development goes really fast!

2.1.4 Getting Linux

If you do have a speedy Internet connection (definitely not a 56k-modem, but perhaps cable modem) a Linux download may be an option to you. In general you can download a specific distribution of Linux from the respective distributions website. Another option is try <http://distrowatch.com/> for ready-to-burn CD/DVD images (the ISO format) of your selected Linux distribution. If you don't have the bandwidth, you can buy ready CDs/DVDs from many websites online (including distrowatch) or borrow a set from friends already using Linux

Chapter 3

Linux Shortcuts and Commands (*Highly recommended*)

Maybe this should have come first. A practical selection of Linux shortcuts and commands in a concise form. Perhaps this is everything that a computer-literate newbie Linuxer really needs. *Highly recommended.* This is a practical selection of the commands we use most often, find useful, and which came on most Linux distribution CD/DVDs. Press <Tab> on the empty command line to see the listing of all available commands (on your PATH). On my small home system, it says there are 3786 executables on my PATH. Many of these "commands" can be accessed from your favourite GUI front-end (probably KDE or Gnome) by clicking appropriate menu, icon, or button. They can all be run from the command line (unless you didn't install the package, but they all came on my Linux CDs). Programs that require GUI have to be run from under the X11, for example from a terminal opened in kde or gnome (e.g., `konsole` or `xterm`).

3.1 Notes for the UNIX Clueless

1. LINUX IS CASE-SENSITIVE. For example: Mozilla, MOZILLA, mOzilla, and mozilla would be four different commands (but of the four, only mozilla is available on my system). Also my_file, my_file, and my_FILE are three different files. Your user login name and password are also case sensitive. (This goes with the tradition of UNIX and the "c" programming language being case sensitive.)
2. Filenames can be up to 256 characters long and can contain letters, numbers, "." (dots), "_" (underscores), "-" (dashes), plus some other non-recommended characters.
3. Files with names starting with "." are normally not shown by the `ls` (list) or `dir` command. Think of these "dot" files as "hidden". Use `ls -a` (list with the option "all") to see these files.
4. "/" is an equivalent to DOS "\" (root directory, meaning the parent of all other directories, or a separator between a directory name and a subdirectory or filename). For example, try `cd /usr/doc`.
5. Under Linux, all directories appear under a single directory tree (there are no DOS-style drive letters). This means directories and files from all physical devices (plus network filesystems) are merged into this single-view tree.

6. In a configuration file, a line starting with “#” is a comment. When changing a configuration file, don’t delete old settings—comment out the original lines with “#”. Always insert a short comment describing what you have done (for your own benefit!).
7. Linux is inherently multi-user. Your personal settings (and all other personal files) are in your home directory which is `/home/your_user_login_name`. Many settings are kept in files with names starting with a dot “.” so as to keep them out of your way (see point 3 above). You have to login to do anything.
8. System-wide settings are kept in the directory `/etc`.
9. Under Linux, as in any multiuser operating system, directories and files have an owner and a set of permissions. You will typically be allowed to write only to your home directory which is `/home/your_user_login_name`. Learn to use the file permissions else you will be constantly annoyed with Linux.
10. Command options are introduced by a dash, “-”, followed by a single letter (or – when the option is more than one letter). Thus “-” is an equivalent of DOS’s switch “/”. For example, try `rm --help`.
11. Type `command &` (the command name followed by an &) to start a command in the background. This is usually the preferred way of starting a program from the X-window terminal.

3.2 Linux Essential Keyboard Shortcuts and Sanity Commands

3.2.1 Moving around your Linux computer

Shortcuts and commands to help you get around your computer like a pro.

`<Ctrl><Alt><F1>`

Switch to the first text terminals. Under Linux you can have several (6 in standard setup) terminals opened at the same time. This is a keyboard shortcut, which means: “press the control key and the alt key, hold them. Now press `⌘F1`. Release all keys.”

`<Ctrl><Alt><Fn> (n=1..6)`

Switch to the *n*th text terminal. (The same could be accomplished with the rarely used command `chvt n`. “chvt” stands for “change virtual terminal”). In text terminal (outside X), you can also use `<Alt><Fn>` (the key `<Ctrl>` is not needed).

`tty`

Print the name of the terminal in which you are typing this command. If you prefer the number of the active terminal (instead of its name), it can be printed using the command `fgconsole` (“foreground console”).

<Ctrl><Alt><F7>

Switch to the first GUI terminal (if X-windows is running on the 7th terminal, where it typically is).

<Ctrl><Alt><Fn> (**n=7..12**)

Switch to the nth GUI terminal (if a GUI terminal is running on screen n-1). On default, the first X server is running on terminal 7. On default, nothing is running on terminals 8 to 12—you can start subsequent X server there.

3.2.2 Command-line Shortcuts

Command-line shortcuts and commands. Behold the power of the console.

. (dot)

Current directory. For example, `./my_program` will attempt to execute the file "my_program" located in your current working directory.

.. (two dots)

Directory parent to the current one. For example, the command `cd ..` will change my current working directory one level up.

~ (tilde character)

My home directory (normally the directory `/home/my_login_name`). For example, the command `cd ~/my_dir` will change my working directory to the subdirectory "my_dir" under my home directory. Typing just `cd` alone is an equivalent of the command `cd ~`. I keep all my files in my home directory.

`reset`

Restore a screwed-up terminal (a terminal showing funny characters) to default setting. Use if you tried to "cat" a binary file. You may not be able to see the command as you type it, but it still will work.

`exit`

Logout. I can also use `logout` for the same effect. (If you have started a second shell, e.g., using `bash`, this command will make you exit the second shell, and you will be back in the first shell, not logged out. Then use another `exit` to `logout`.)

<Ctrl><Alt>

Shut down the system and reboot. This is the normal shutdown command for a user at the text-mode console. Don't just press the "reset" button for shutdown!

<Tab>

Autocomplete the command if there is only one option, or else show all the available options. On newer systems you may need to press <Tab><Tab>. THIS SHORTCUT IS GREAT, it can truly save you lots of time.

<ArrowUp>

Scroll and edit the command history. Press `⏎` to execute a historical command (to save on typing). <ArrowDown> scrolls back.

<Shift><PgUp>

Scroll terminal output up. This works also at the login prompt, so you can scroll through your bootup messages. The amount/usage of your video memory determines how far back you can scroll the display. <Shift><PgDown> scrolls the terminal output down.

<Ctrl>c

Kill the current process (works mostly with small text-mode applications).

<Ctrl>d

(pressed at the beginning of an empty line) Log out from the current terminal. See also the next command.

<Ctrl>d

Send [End-of-File] to the current process. Don't press it twice else you also log out (see the previous command).

<Ctrl>s

Stop the transfer to the terminal.

`<Ctrl>q`

Resume the transfer to the terminal. Try if your terminal mysteriously stops responding. See the previous command.

`<Ctrl>z`

Send the current process to the background.

`<MiddleMouseButton>`

Paste the text which is currently highlighted somewhere else. This is the normal "copy-paste" operation in Linux. It is a fast and powerful supplement to the widely-known GUI "copy-paste" menu-based operation. (It works in the text terminal if you enabled "gpm" service using "setup". It also works inside any dialog boxes, etc.—really convenient!) It is best used with a Linux-ready 3-button mouse (Logitech or similar) or else set "3-mouse button emulation". The `<MiddleMouseButton>` is normally emulated on a 2-button mouse by pressing both mouse buttons simultaneously.

3.2.3 X11 (graphical-interface) Shortcuts

Shortcuts and commands that can be used when running the X-window system.

`<Ctrl><Alt><+>`

Change to the next X-server resolution (if you set up the X-server to more than one resolution). For multiple resolutions on my standard SVGA card/monitor, I have the following line in the file `/etc/X11/XF86Config` (the first resolution starts on default, the largest resolution determines the size of the "virtual screen"):

```
Modes "1024x768" "800x600" "640x480" "512x384" "480x300" "400x300" "1152x864"
```

Of course, first I had to configure the X server, either by using `xorgconfig`, `xorgcfg`, or manually by editing the file `/etc/X11/xorg.conf`, so that it supports the above resolutions (mostly the matter of uncommenting the line that defines my video chipset, and specifying the synchronization frequencies my monitor supports). See also the commands `xvidtune` and `xvidgen`.

`<Ctrl><Alt><->`

Change to the previous X-server resolution.

`<Ctrl><Alt><Esc>`

Kill the window I am going to click with my mouse pointer (the pointer changes to something like a death symbol). Similar result can be obtained with the command `xkill` (typed in X-terminal). Useful when an X-window program

does not want to close (hangs?).

```
<Ctrl><Alt><BkSpc>
```

Kill the current X-windows server. Use if the X-windows server cannot be exited normally.

```
<Ctrl><Alt><d>
```

Toggle hide/show the desktop (great to hide the Solitaire game when your boss walks in).

```
<MiddleMouseButton>
```

Paste the text which is currently highlighted somewhere else. This is the normal "copy-paste" operation in Linux. It is a fast and powerful supplement to the widely-known GUI "copy-paste" menu-based operation. (It works in the text terminal if you enabled "gpm" service using "setup". It also works inside any dialog boxes, etc.—really convenient!) It is best used with a Linux-ready 3-button mouse (Logitech or similar) or else set "3-mouse button emulation". The `<MiddleMouseButton>` is normally emulated on a 2-button mouse by pressing both mouse buttons simultaneously.

3.2.4 Non-Essential Shortcuts

This is a group of key combinations implemented at the Linux kernel level (a low level). It means, chances are these key combinations will work most of the time. The combinations are meant for debugging purposes and in an emergency (mostly developers); you should try other, safer solutions first. The key `<SysRq>` is also known on PC as `<PrintScreen>`. The combinations can be enabled/disabled by setting the relevant kernel variable to "1" or "0", e.g. `: echo "1" > /proc/sys/kernel/sysrq`

```
<Alt><SysRq><k>
```

Kill all processes (including X) which are running on the currently active virtual console. This key combination is known as "secure access key" (SAK).

```
<Alt><SysRq><e>
```

Send the TERM signal to all running processes except init, asking them to exit.

```
<Alt><SysRq><i>
```

Send the KILL signal to all running processes except init. This may be more successful in killing runaway processes than the previous key combination, but it may cause some of them to exit abnormally.

<Alt><SysRq><l>

Send the KILL signal to all processes, including init. The system will not be functional.

<Alt><SysRq><s>

Run an emergency sync (cache write) on all mounted filesystems. This can prevent data loss.

<Alt><SysRq><u>

Remount all mounted filesystems as read-only. This has the same effect as the sync combination above, but with one important benefit: if the operation is successful, fsck won't have to check all filesystems after a computer hardware reset.

<Alt><SysRq><r>

Turn off keyboard raw mode. This can be useful when your X session hangs. After issueing this command you may be able to use <CTRL><ALT>.

<Alt><SysRq>

Reboot immediately without syncing or unmounting your disks. You will likely end up with filesystem errors.

<Alt><SysRq><o>

Shut the system off (if configured and supported).

<Alt><SysRq><p>

Dump the current registers and flags to your console.

<Alt><SysRq><t>

Dump a list of current tasks and their information to your console.

<Alt><SysRq><m>

Dump memory info to your console.

```
<Alt>SysRq<digit>
```

The digit is '0' to '9'. Set the console log level, controlling which kernel messages will be printed to your console. For example, '0' will cause only emergency messages like PANICs or OOPSes displayed on your console.

```
<Alt><SysRq><h>
```

Display help. Also, any other unsupported `<Alt><SysRq><key>` combination will display the same help.

3.3 Help Commands

A simple list of commands that can give you help when you are trying to do something.

```
any_command --help |more
```

Display a brief help on a command (works with most commands). For example, try `cp -help --more`. `--help` works similar to DOS `/h` switch. The `more` pipe is needed when the output is longer than one screen.

```
man topic
```

Display the contents of the system manual pages (help) on the topic. Press `q` to quit the viewer. Try `man man` if you need any advanced options. The command `info topic` works similar to `man topic`, yet it may contain more up-to-date information. Manual pages can be hard to read—they were written for UNIX programmers. Try `any_command --help` for a brief, easier to digest help on a command. Some programs also come with README or other info files—have a look to the directory `/usr/share/doc`. To display manual page from a specific section, I may use something like: `man 3 exit` (this displays an info on the command `exit` from section 3 of the manual pages) or `man -a exit` (this displays man pages for `exit` from all sections).

The man sections are:

- Section 1 - Commands
- Section 2 - System Calls
- Section 3 - Library Calls
- Section 4 - Special Files (Devices etc.)
- Section 5 - File Formats and Conventions
- Section 6 - Games
- Section 7 - Miscellaneous
- Section 8 - System Administration

- Section 9 - ()

To print a manual page, I use: `man topic | col -b | lpr` (the option `col -b` removes any backspace or other characters that could make the printed manpage difficult to read).

```
info topic
```

Display the contents of the info on a particular command. info is a replacement for man pages so it contains the most recent updates to the system documentation. Use `␣` and `␣` to move around or you may get confused. Press "q" to quit. A replacement for the somewhat confusing info browsing system might be `pinfo` - try if you like it any better.

```
apropos topic
```

Give me the list of the commands that have something to do with my topic.

```
whatis topic
```

Give me a short list of commands matching my topic. `whatis` is similar to `apropos` (see the command above)—they both use the same database. But `whatis` searches keywords, while `apropos` also searches the descriptions of the keywords.

```
help command
```

Display brief info on a bash (shell) built-in command. Using `help` with no command prints the list of all bash built-in commands. The shortest list of bash built-in commands would probably include: `alias`, `bg`, `cd`, `echo`, `exit`, `export`, `fg`, `help`, `history`, `jobs`, `kill`, `logout`, `pwd`, `set`, `source`, `ulimit`, `umask`, `unalias`, `unset`.

```
kdehelpcenter &
```

(in X-terminal). Browse the whole system help using the graphical KDE help navigator. Normally, KDE help is invoked by pressing the appropriate icon on the KDE control panel. Use `gnome-help-browser` for the GNOME equivalent. (The older KDE versions used `kdehelp&` instead of `kdehelpcenter&`).

3.4 System Info

Commands that let you determine what is going on with your system.

pwd

Print working directory, i.e., display the name of my current directory on the screen.

hostname

Print the name of the local host (the machine on which I am working). Use netconf (as root) to change the name of the machine.

whoami

Print my login name.

id username

Print user id (uid) and his/her group id (gid), effective id (if different than the real id) and the supplementary groups.

date

Print the operating system current date, time and timezone. For an ISO standard format, I have to use `date -Iseconds`. I can change the date and time to 2000-12-31 23:57 using the command `date 123123572000` or using these two commands (easier to remember):

```
date --set 2000-12-31
date --set 23:57:00
```

To set the hardware (BIOS) clock from the system (Linux) clock, I can use the command (as root) `setclock`. The international (ISO 8601) standard format for all-numeric date/time has the form: 2001-01-31 (as in Linux default "C" localization). You can be more precise if you wish using, for example: 2001-01-31 23:59:59.999-05:00 (representing 1 millisecond before February 2001, in a timezone which is 5 hours behind the Universal Coordinated Time (UTC)) . The most "kosher" representation of the same point in time could be: 20010131T235959,999-0500. See the standard at <ftp://ftp.qsl.net/pub/g1smd/8601v03.pdf>.

time

Determine the amount of time that it takes for a process to complete + other process accounting. Don't confuse it with the `date` command (see previous entry). E.g. I can find out how long it takes to display a directory content using: `time ls`. Or I can test the time function with `time sleep 10` (time the commands the does nothing for 10 seconds).

clock **and** hwclock

(two commands, use either). Obtain date/time from the computer hardware (real time, battery-powered) clock. You can also use one of this commands to set the hardware clock, but setclock may be simpler (see command above). Example: `hwclock --systohc --utc` sets the hardware clock (in UTC) from the system clock.

who

Determine the users logged on the machine.

w

Determine who is logged on the system, find out what they are doing, their processor usage, etc. Handy security command.

rwho -a

(=remote who) Determine users logged on other computers on your network. The rwho service must be enabled for this command to run. If it isn't, run `setup` (RedHat specific) as root to enable "rwho".

finger user_name

System info about a user. Try: `finger root`. One can use finger with any networked computer that exposes the finger service to the world, e.g., I can do (try): `finger @finger.kernel.org`

last

Show listing of users last logged-in on your system. Really good idea to check it from time to time as a security measure on your system.

lastb

("=last bad") Show the last bad (unsuccessful) login attempts on my system. It did not work on my system, so got it started with: `touch /var/log/btmp`

"There's a good reason why `/var/log/btmp` isn't available on any sane set-up – it's a world-readable file containing login mistakes. Since one of the most common login mistakes is to type the password instead of the username, `/var/log/btmp` is a gift to crackers." (Thanks to Bruce Richardson). It appears the problem can be solved by changing the file permissions so only root can use "lastb":

`chmod o-r /var/log/btmp`

```
history | more
```

Show the last (1000 or so) commands executed from the command line on the current account. The `”| more”` causes the display to stop after each screen-full. To see what another user was doing on your system, login as `”root”` and inspect his/her `”history”`. The history is kept in the file `.bash_history` in the user home directory (so yes, it can be modified or erased).

```
comman
```

Run the most recent command from my bash history commands that starts with the string `”comman”`. This is usefull for re-running often the same, command which may be complicated to type. For example, I use the following:

```
so
```

and this is automatically expanded to

```
sol --variation freecell &
```

and a solitaire game with a special option is lunched for me.

```
uptime
```

Show the amount of time since the last reboot.

```
ps
```

(=`”print status”` or `”process status”`) List the processes currently run by the current user.

```
ps aux | more
```

List all the processes currently running, even those without the controlling terminal, together with the name of the user that owns each process.

```
top
```

Keep listing the currently running processes on my computer, sorted by cpu usage (top processes first). Press `<Ctrl>c` when done.

- PID = process identification.
- USER = name of the user who owns (started?) the process.
- PRI = priority of the process (the higher the number, the lower the priority, normal 0, highest priority is -20, lowest 20).

- NI = niceness level (i.e., if the process tries to be nice by adjusting the priority by the number given). The higher the number, the higher the niceness of the process (i.e., its priority is lower).
- SIZE = kilobytes of code+data+stack taken by the process in memory.
- RSS = kilobytes of physical (silicon) memory taken.
- SHARE = kilobytes of memory shared with other processes.
- STAT = state of the process: S-sleeping, R-running, T-stopped or traced, D-uninterruptable sleep, Z=zombie.
- %CPU = share of the CPU usage (since last screen update).
- %MEM = share of physical memory.
- TIME = total CPU time used by the process (since it was started).
- COMMAND = command line used to start the task (careful with passwords, etc., on command line, all permitted to run "top" may see them!

gtop, ktop **and** htop

(in X terminal) Three GUI choices for top. My favourite is gtop (comes with gnome). In KDE, ktop is also available from the "K" menu under "System"->"Task Manager".

uname -a

(= "Unix name" with option "all") Info on your (local) server. I can also use guname (in X-window terminal) to display the info more nicely.

Xorg -version

Show me the version of X windows I have on my system.

cat /etc/issue

Check what distribution you are using. You can put your own message in this text file – it's displayed on login. It is more common to put your site-specific login message to the file /etc/motd ("motd"="message of the day").

free

Memory info (in kilobytes). "Shared" memory is the memory that can be shared between processes (e.g., executable code is "shared"). "Buffered" and "cached" memory is the part that keeps parts of recently accessed files – it can be shrunk if more memory is needed by processes.

```
df -h
```

(=disk free) Print disk info about all the filesystems (in human-readable form).

```
du / -bh | more
```

(=disk usage) Print detailed disk usage for each subdirectory starting at the "/" (root) directory (in human legible form).

```
cat /proc/cpuinfo
```

Cpu info – shows the content of the file `cpuinfo`. Note that the files in the `/proc` directory are not real files – they are hooks to look at information available to the kernel.

```
cat /proc/interrupts
```

List the interrupts in use. May need to find out before setting up new hardware.

```
cat /proc/version
```

Linux version and other info.

```
cat /proc/filesystems
```

Show the types of filesystems currently in use.

```
cat /etc/printcap |more
```

Show the setup of printers.

```
lsmod
```

(= "list modules". As root. Use `/sbin/lsmod` to execute this command when you are a non-root user.) Show the kernel modules currently loaded.

```
set |more
```

Show the current user environment (in full). Normally too much to bother.

```
echo $PATH
```

Show the content of the environment variable `PATH`. This command can be used to show other environment variables as well. Use `set` to see the full environment (see the previous command).

```
dmesg | less
```

Print kernel messages (the content of the so-called kernel ring buffer). Press "q" to quit "less". Use `less /var/log/dmesg` to see what "dmesg" dumped into this file right after the last system bootup.

```
chage -l my_login_name
```

See my password expiry information.

```
quota
```

See my disk quota (the limits of disk usage).

```
sysctl -a |more
```

Display all the configurable Linux kernel parameters.

```
runlevel
```

Print the previous and current runlevel. The output "N5" means: "no previous runlevel" and "5 is the current runlevel". To change the runlevel, use "init", e.g., `init 1` switches the system to a single user mode.

Runlevel is the mode of operation of Linux. Runlevel can be switched "on the fly" using the command `init`. For example, `init 3` (as root) will switch me to runlevel 3. The following runlevels are standard:

- 0 - halt (Do NOT set `initdefault` to this)
- 1 - Single user mode
- 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
- 3 - Full multiuser mode
- 4 - unused
- 5 - X11
- 6 - reboot (Do NOT set `initdefault` to this)

The system default runlevel is set in the file: `/etc/inittab`.

sar

View information extracted the system activity log file (`/var/log/sarxx` where `xx` is the current day number). `sar` can extract many kinds of system statistics including CPU load averages, i/o statistics, and network traffic statistics for the current day and (usually) several days backs.