Introduction to Linux

A Hands on Guide

Machtelt Garrels

Garrels.be

<tille wants no spam _at_ garrels dot be>

1.27 Edition

20080606

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Introduction

1. Why this guide?

Many people still believe that learning Linux is difficult, or that only experts can understand how a Linux system works. Though there is a lot of free documentation available, the documentation is widely scattered on the Web, and often confusing, since it is usually oriented toward experienced UNIX or Linux users. Today, thanks to the advancements in development, Linux has grown in popularity both at home and at work. The goal of this guide is to show people of all ages that Linux can be simple and fun, and used for all kinds of purposes.

2. Who should read this book?

This guide was created as an overview of the Linux Operating System, geared toward new users as an exploration tour and getting started guide, with exercises at the end of each chapter. For more advanced trainees it can be a desktop reference, and a collection of the base knowledge needed to proceed with system and network administration. This book contains many real life examples derived from the author's experience as a Linux system and network administrator, trainer and consultant. We hope these examples will help you to get a better understanding of the Linux system and that you feel encouraged to try out things on your own.

Everybody who wants to get a "CLUE", a Command Line User Experience, with Linux (and UNIX in general) will find this book useful.

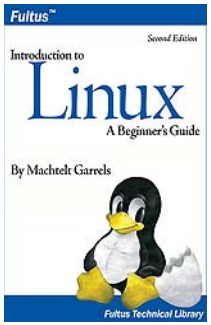
3. New versions and availability

This document is published in the Guides section of the Linux Documentation Project collection at http://www.tldp.org/guides.html; you can also download PDF and PostScript formatted versions here.

The most recent edition is available at http://tille.garrels.be/training/tldp/.

The second edition of this guide is available in print from Fultus.com Books as paperback Print On Demand (POD) book. Fultus distributes this document through Ingram and Baker & Taylor to many bookstores, including Amazon.com, Amazon.co.uk, BarnesAndNoble.com and Google's Froogle global shopping portal and Google Book Search.

Figure 1. Introduction to Linux front cover



The guide has been translated into Hindi by:

- Alok Kumar
- Dhananjay Sharma
- Kapil
- Puneet Goel
- Ravikant Yuyutsu

Andrea Montagner translated the guide into Italian.

4. Revision History

Revision History

Revision 1.27	20080606	Revised by: MG
updates.		
Revision 1.26	20070919	Revised by: MG
Comments from readers, license.		
Revision 1.25	20070511	Revised by: MG
Comments from readers, minor updates, E-mail etiquette, updated info about availability (thanks Oleg).		
Revision 1.24	2006-11-01	Revised by: MG
added index terms, prepared for second printed edition, added gpg and proxy info.		
Revision 1.23	2006-07-25	Revised by: MG and FK
Updates and corrections, removed app5 again, adapted license to enable inclusion in Debian docs.		
Revision 1.22	2006-04-06	Revised by: MG
chap8 revised completely, chap10: clarified examples, added ifconfig and cygwin info, revised network apps.		
Revision 1.21	2006-03-14	Revised by: MG
Added exercises in chap11, corrected newline errors, command overview completed for chapter 9, minor corrections in chap10.		
Revision 1.20	2006-01-06	Revised by: MG
Split chap7: audio stuff is now in separate chapter, chap11.xml. Small revisions, updates for commands like aptitude, more on USB storage, Internet telephony, corrections from readers.		
Revision 1.13	2004-04-27	Revised by: MG

Last read-through before sending everything to Fultus for printout. Added Fultus referrence in New Versions section, updated Conventions and Organization sections. Minor changes in chapters 4, 5, 6 and 8, added rdesktop info in chapter 10, updated glossary, replaced references to fileutils with coreutils, thankyou to Hindi translators.

5. Contributions

Many thanks to all the people who shared their experiences. And especially to the Belgian Linux users for hearing me out every day and always being generous in their comments.

Also a special thought for Tabatha Marshall for doing a really thorough revision, spell check and styling, and to Eugene Crosser for spotting the errors that we two overlooked.

And thanks to all the readers who notified me about missing topics and who helped to pick out the last errors, unclear definitions and typos by going through the trouble of mailing me all their remarks. These are also the people who help me keep this guide up to date, like Filipus Klutiero who did a complete review in 2005 and 2006 and helps me getting the guide into the Debian docs collection, and Alexey Eremenko who sent me the foundation for chapter 11.

In 2006, Suresh Rajashekara created a Debian package of this documentation.

Finally, a big thank you for the volunteers who are currently translating this document in French, Swedish, German, Farsi, Hindi and more. It is a big work that should not be underestimated; I admire your courage.

6. Feedback

Missing information, missing links, missing characters? Mail it to the maintainer of this document:

<tille wants no spam _at_ garrels dot be>

Don't forget to check with the latest version first!

7. Copyright information

```
* Copyright (c) 2002-2007, Machtelt Garrels
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
*     * Redistributions of source code must retain the above copyright
*       notice, this list of conditions and the following disclaimer.
*     * Redistributions in binary form must reproduce the above copyright
*       notice, this list of conditions and the following disclaimer in the
*       documentation and/or other materials provided with the distribution.
*     * Neither the name of the author, Machtelt Garrels, nor the
*       names of its contributors may be used to endorse or promote products
*       derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY
```



```
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE AUTHOR AND CONTRIBUTORS BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

The logos, trademarks and symbols used in this book are the properties of their respective owners.

8. What do you need?

You will require a computer and a medium containing a Linux distribution. Most of this guide applies to all Linux distributions - and UNIX in general. Apart from time, there are no further specific requirements.

The Installation HOWTO contains helpful information on how to obtain Linux software and install it on your computer. Hardware requirements and coexistence with other operating systems are also discussed.

CD images can be downloaded from linux-iso.com and many other locations, see Appendix A.

An interesting alternative for those who don't dare to take the step of an actual Linux installation on their machine are the Linux distributions that you can run from a CD, such as the Knoppix distribution.

9. Conventions used in this document

The following typographic and usage conventions occur in this text:

Table 1. Typographic and usage conventions

Text type	Meaning
"Quoted text"	Quotes from people, quoted computer output.
`terminal view`	Literal computer input and output captured from the terminal, usually rendered with a light grey background.
command	Name of a command that can be entered on the command line.
`VARIABLE`	Name of a variable or pointer to content of a variable, as in `$VARNAME`.
`option`	Option to a command, as in "the `-a` option to the **ls** command".
`argument`	Argument to a command, as in "read **man *`ls`*** ".
`prompt`	User prompt, usually followed by a command that you type in a terminal window, like in `hilda@home>` **`ls -l`**
command `options` *`arguments`*	Command synopsis or general usage, on a separated line.
`filename`	Name of a file or directory, for example "Change to the `/usr/bin` directory."
Key	Keys to hit on the keyboard, such as "type **Q** to quit".
Button	Graphical button to click, like the OK button.
Menu->Choice	Choice to select from a graphical menu, for instance: "SelectHelp->About Mozilla in your browser."

Terminology	Important term or concept: "The Linux *kernel* is the heart of the system."
\	The backslash in a terminal view or command synopsis indicates an unfinished line. In other words, if you see a long command that is cut into multiple lines, \ means "Don't press **Enter** yet!"
See Chapter 1	link to related subject within this guide.
The author	Clickable link to an external web resource.

The following images are used:

This is a note

It contains additional information or remarks.

This is a caution

It means be careful.

This is a warning

Be *very* careful.

This is a tip

Tips and tricks.

10. Organization of this document

This guide is part of the Linux Documentation Project and aims to be the foundation for all other materials that you can get from the Project. As such, it provides you with the fundamental knowledge needed by anyone who wants to start working with a Linux system, while at the same time it tries to consciously avoid re-inventing the hot water. Thus, you can expect this book to be incomplete and full of links to sources of additional information on your system, on the Internet and in your system documentation.

The first chapter is an introduction to the subject on Linux; the next two discuss absolute basic commands. Chapters 4 and 5 discuss some more advanced but still basic topics. Chapter 6 is needed for continuing with the rest, since it discusses editing files, an ability you need to pass from Linux newbie to Linux user. The following chapters discuss somewhat more advanced topics that you will have to deal with in everyday Linux use.

All chapters come with exercises that will test your preparedness for the next chapter.

- Chapter 1: What is Linux, how did it come into existence, advantages and disadvantages, what does the future hold for Linux, who should use it, installing your computer.
- Chapter 2: Getting started, connecting to the system, basic commands, where to find help.
- Chapter 3: The filesystem, important files and directories, managing files and directories, protecting your data.
- Chapter 4: Understanding and managing processes, boot and shutdown procedures, postponing tasks, repetitive tasks.
- Chapter 5: What are standard input, output and error and how are these features used from the command line.
- Chapter 6: Why you should learn to work with an editor, discussion of the most common editors.

- Chapter 7: Configuring your graphical, text and audio environment, settings for the non-native English speaking Linux user, tips for adding extra software.
- Chapter 8: Converting files to a printable format, getting them out of the printer, hints for solving print problems.
- Chapter 9: Preparing data to be backed up, discussion of various tools, remote backup.
- Chapter 10: Overview of Linux networking tools and user applications, with a short discussion of the underlying service daemon programs and secure networking.
- Chapter 11: Sound and video, including Voice over IP and sound recording is discussed in this chapter.
- Appendix A: Which books to read and sites to visit when you have finished reading this one.
- Appendix B: A comparison.
- Appendix C: If you ever get stuck, these tables might be an outcome. Also a good argument when your boss insists that YOU should use HIS favorite shell.

Chapter 1. What is Linux?

We will start with an overview of how Linux became the operating system it is today. We will discuss past and future development and take a closer look at the advantages and disadvantages of this system. We will talk about distributions, about Open Source in general and try to explain a little something about GNU.

This chapter answers questions like:

- What is Linux?
- Where and how did Linux start?
- Isn't Linux that system where everything is done in text mode?
- Does Linux have a future or is it just hype?
- What are the advantages of using Linux?
- What are the disadvantages?
- What kinds of Linux are there and how do I choose the one that fits me?
- What are the Open Source and GNU movements?

1.1. History

1.1.1. UNIX

In order to understand the popularity of Linux, we need to travel back in time, about 30 years ago...

Imagine computers as big as houses, even stadiums. While the sizes of those computers posed substantial problems, there was one thing that made this even worse: every computer had a different operating system. Software was always customized to serve a specific purpose, and software for one given system didn't run on another system. Being able to work with one system didn't automatically mean that you could work with another. It was difficult, both for the users and the system administrators.

Computers were extremely expensive then, and sacrifices had to be made even after the original purchase just to get the users to understand how they worked. The total cost per unit of computing power was enormous.

Technologically the world was not quite that advanced, so they had to live with the size for another decade. In 1969, a team of developers in the Bell Labs laboratories started working on a solution for the software problem, to address these compatibility issues. They developed a new operating system, which was

1. Simple and elegant.
2. Written in the C programming language instead of in assembly code.
3. Able to recycle code.

The Bell Labs developers named their project "UNIX."

The code recycling features were very important. Until then, all commercially available computer systems were written in a code specifically developed for one system. UNIX on the other hand needed only a small piece of that special code, which is now commonly named the kernel. This kernel is the only piece of code that needs to be adapted for every specific system and forms the base of the UNIX system. The operating system and all other functions were built around this kernel and written in a higher programming language, C.

This language was especially developed for creating the UNIX system. Using this new technique, it was much easier to develop an operating system that could run on many different types of hardware.

The software vendors were quick to adapt, since they could sell ten times more software almost effortlessly. Weird new situations came in existence: imagine for instance computers from different vendors communicating in the same network, or users working on different systems without the need for extra education to use another computer. UNIX did a great deal to help users become compatible with different systems.

Throughout the next couple of decades the development of UNIX continued. More things became possible to do and more hardware and software vendors added support for UNIX to their products.

UNIX was initially found only in very large environments with mainframes and minicomputers (note that a PC is a "micro" computer). You had to work at a university, for the government or for large financial corporations in order to get your hands on a UNIX system.

But smaller computers were being developed, and by the end of the 80's, many people had home computers. By that time, there were several versions of UNIX available for the PC architecture, but none of them were truly free and more important: they were all terribly slow, so most people ran MS DOS or Windows 3.1 on their home PCs.

1.1.2. Linus and Linux

By the beginning of the 90s home PCs were finally powerful enough to run a full blown UNIX. Linus Torvalds, a young man studying computer science at the university of Helsinki, thought it would be a good idea to have some sort of freely available academic version of UNIX, and promptly started to code.

He started to ask questions, looking for answers and solutions that would help him get UNIX on his PC. Below is one of his first posts in comp.os.minix, dating from 1991:

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

From the start, it was Linus' goal to have a free system that was completely compliant with the original UNIX. That is why he asked for POSIX standards, POSIX still being the standard for UNIX.

In those days plug-and-play wasn't invented yet, but so many people were interested in having a UNIX system of their own, that this was only a small obstacle. New drivers became available for all kinds of new hardware, at a continuously rising speed. Almost as soon as a new piece of hardware became available, someone bought it and submitted it to the Linux test, as the system was gradually being called, releasing more free code for an ever wider range of hardware. These coders didn't stop at their PC's; every piece of hardware they could find was useful for Linux.

Back then, those people were called "nerds" or "freaks", but it didn't matter to them, as long as the supported hardware list grew longer and longer. Thanks to these people, Linux is now not only ideal to run on new PC's,

but is also the system of choice for old and exotic hardware that would be useless if Linux didn't exist.

Two years after Linus' post, there were 12000 Linux users. The project, popular with hobbyists, grew steadily, all the while staying within the bounds of the POSIX standard. All the features of UNIX were added over the next couple of years, resulting in the mature operating system Linux has become today. Linux is a full UNIX clone, fit for use on workstations as well as on middle-range and high-end servers. Today, a lot of the important players on the hard- and software market each have their team of Linux developers; at your local dealer's you can even buy pre-installed Linux systems with official support - eventhough there is still a lot of hard- and software that is not supported, too.

1.1.3. Current application of Linux systems

Today Linux has joined the desktop market. Linux developers concentrated on networking and services in the beginning, and office applications have been the last barrier to be taken down. We don't like to admit that Microsoft is ruling this market, so plenty of alternatives have been started over the last couple of years to make Linux an acceptable choice as a workstation, providing an easy user interface and MS compatible office applications like word processors, spreadsheets, presentations and the like.

On the server side, Linux is well-known as a stable and reliable platform, providing database and trading services for companies like Amazon, the well-known online bookshop, US Post Office, the German army and many others. Especially Internet providers and Internet service providers have grown fond of Linux as firewall, proxy- and web server, and you will find a Linux box within reach of every UNIX system administrator who appreciates a comfortable management station. Clusters of Linux machines are used in the creation of movies such as "Titanic", "Shrek" and others. In post offices, they are the nerve centers that route mail and in large search engine, clusters are used to perform internet searches.These are only a few of the thousands of heavy-duty jobs that Linux is performing day-to-day across the world.

It is also worth to note that modern Linux not only runs on workstations, mid- and high-end servers, but also on "gadgets" like PDA's, mobiles, a shipload of embedded applications and even on experimental wristwatches. This makes Linux the only operating system in the world covering such a wide range of hardware.

1.2. The user interface

1.2.1. Is Linux difficult?

Whether Linux is difficult to learn depends on the person you're asking. Experienced UNIX users will say no, because Linux is an ideal operating system for power-users and programmers, because it has been and is being developed by such people.

Everything a good programmer can wish for is available: compilers, libraries, development and debugging tools. These packages come with every standard Linux distribution. The C-compiler is included for free - as opposed to many UNIX distributions demanding licensing fees for this tool. All the documentation and manuals are there, and examples are often included to help you get started in no time. It feels like UNIX and switching between UNIX and Linux is a natural thing.

In the early days of Linux, being an expert was kind of required to start using the system. Those who mastered Linux felt better than the rest of the "lusers" who hadn't seen the light yet. It was common practice to tell a beginning user to "RTFM" (read the manuals). While the manuals were on every system, it was difficult to

find the documentation, and even if someone did, explanations were in such technical terms that the new user became easily discouraged from learning the system.

The Linux-using community started to realize that if Linux was ever to be an important player on the operating system market, there had to be some serious changes in the accessibility of the system.

1.2.2. Linux for non-experienced users

Companies such as RedHat, SuSE and Mandriva have sprung up, providing packaged Linux distributions suitable for mass consumption. They integrated a great deal of graphical user interfaces (GUIs), developed by the community, in order to ease management of programs and services. As a Linux user today you have all the means of getting to know your system inside out, but it is no longer necessary to have that knowledge in order to make the system comply to your requests.

Nowadays you can log in graphically and start all required applications without even having to type a single character, while you still have the ability to access the core of the system if needed. Because of its structure, Linux allows a user to grow into the system: it equally fits new and experienced users. New users are not forced to do difficult things, while experienced users are not forced to work in the same way they did when they first started learning Linux.

While development in the service area continues, great things are being done for desktop users, generally considered as the group least likely to know how a system works. Developers of desktop applications are making incredible efforts to make the most beautiful desktops you've ever seen, or to make your Linux machine look just like your former MS Windows or an Apple workstation. The latest developments also include 3D acceleration support and support for USB devices, single-click updates of system and packages, and so on. Linux has these, and tries to present all available services in a logical form that ordinary people can understand. Below is a short list containing some great examples; these sites have a lot of screenshots that will give you a glimpse of what Linux on the desktop can be like:

- http://www.gnome.org
- http://kde.org/screenshots/
- http://www.openoffice.org
- http://www.mozilla.org

1.3. Does Linux have a future?

1.3.1. Open Source

The idea behind Open Source software is rather simple: when programmers can read, distribute and change code, the code will mature. People can adapt it, fix it, debug it, and they can do it at a speed that dwarfs the performance of software developers at conventional companies. This software will be more flexible and of a better quality than software that has been developed using the conventional channels, because more people have tested it in more different conditions than the closed software developer ever can.

The Open Source initiative started to make this clear to the commercial world, and very slowly, commercial vendors are starting to see the point. While lots of academics and technical people have already been convinced for 20 years now that this is the way to go, commercial vendors needed applications like the Internet to make them realize they can profit from Open Source. Now Linux has grown past the stage where it was almost exclusively an academic system, useful only to a handful of people with a technical background.

Now Linux provides more than the operating system: there is an entire infrastructure supporting the chain of effort of creating an operating system, of making and testing programs for it, of bringing everything to the users, of supplying maintenance, updates and support and customizations, etcetera. Today, Linux is ready to accept the challenge of a fast-changing world.

1.3.2. Ten years of experience at your service

While Linux is probably the most well-known Open Source initiative, there is another project that contributed enormously to the popularity of the Linux operating system. This project is called SAMBA, and its achievement is the reverse engineering of the Server Message Block (SMB)/Common Internet File System (CIFS) protocol used for file- and print-serving on PC-related machines, natively supported by MS Windows NT and OS/2, and Linux. Packages are now available for almost every system and provide interconnection solutions in mixed environments using MS Windows protocols: Windows-compatible (up to and includingWinXP) file- and print-servers.

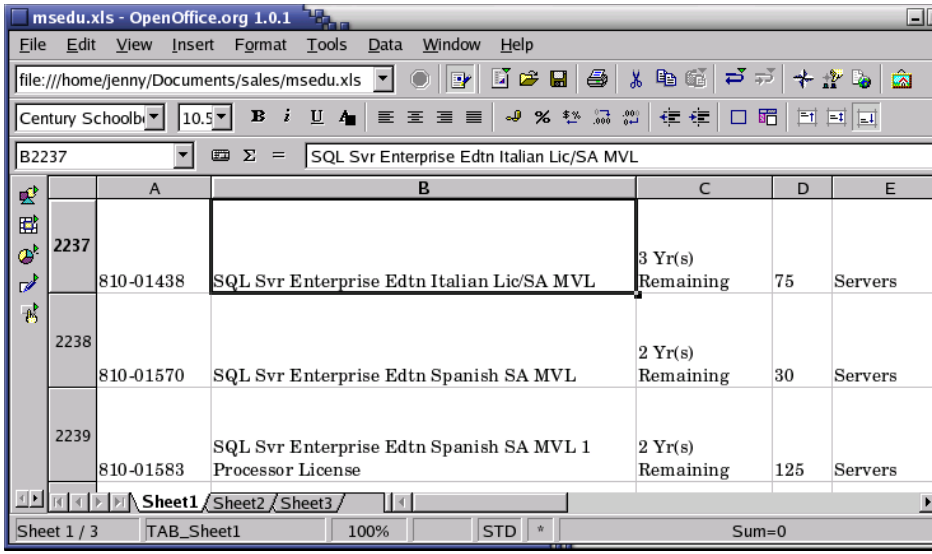
Maybe even more successful than the SAMBA project is the Apache HTTP server project. The server runs on UNIX, Windows NT and many other operating systems. Originally known as "A PAtCHy server", based on existing code and a series of "patch files", the name for the matured code deserves to be connoted with the native American tribe of the Apache, well-known for their superior skills in warfare strategy and inexhaustible endurance. Apache has been shown to be substantially faster, more stable and more feature-full than many other web servers. Apache is run on sites that get millions of visitors per day, and while no official support is provided by the developers, the Apache user community provides answers to all your questions. Commercial support is now being provided by a number of third parties.

In the category of office applications, a choice of MS Office suite clones is available, ranging from partial to full implementations of the applications available on MS Windows workstations. These initiatives helped a great deal to make Linux acceptable for the desktop market, because the users don't need extra training to learn how to work with new systems. With the desktop comes the praise of the common users, and not only their praise, but also their specific requirements, which are growing more intricate and demanding by the day.

The Open Source community, consisting largely of people who have been contributing for over half a decade, assures Linux' position as an important player on the desktop market as well as in general IT application. Paid employees and volunteers alike are working diligently so that Linux can maintain a position in the market. The more users, the more questions. The Open Source community makes sure answers keep coming, and watches the quality of the answers with a suspicious eye, resulting in ever more stability and accessibility.

Listing all the available Linux software is beyond the scope of this guide, as there are tens of thousands of packages. Throughout this course we will present you with the most common packages, which are almost all freely available. In order to take away some of the fear of the beginning user, here's a screenshot of one of your most-wanted programs. You can see for yourself that no effort has been spared to make users who are switching from Windows feel at home:

Figure 1-1. OpenOffice MS-compatible Spreadsheet



1.4. Properties of Linux

1.4.1. Linux Pros

A lot of the advantages of Linux are a consequence of Linux' origins, deeply rooted in UNIX, except for the first advantage, of course:

- Linux is free:

  As in free beer, they say. If you want to spend absolutely nothing, you don't even have to pay the price of a CD. Linux can be downloaded in its entirety from the Internet completely for free. No registration fees, no costs per user, free updates, and freely available source code in case you want to change the behavior of your system.

  Most of all, Linux is free as in free speech:

  The license commonly used is the GNU Public License (GPL). The license says that anybody who may want to do so, has the right to change Linux and eventually to redistribute a changed version, on the one condition that the code is still available after redistribution. In practice, you are free to grab a kernel image, for instance to add support for teletransportation machines or time travel and sell your new code, as long as your customers can still have a copy of that code.
- Linux is portable to any hardware platform:

  A vendor who wants to sell a new type of computer and who doesn't know what kind of OS his new machine will run (say the CPU in your car or washing machine), can take a Linux kernel and make it