THE EXPERT'S VOICE® IN OPEN SOURCE

Introducing Linux Distros

Choose the right Linux distribution for your needs

Jose Dieguez Castro

Apress®

Introducing Linux Distros

Jose Dieguez Castro

Apress®

Introducing Linux Distros

Jose Dieguez Castro
A Coruña
Spain

ISBN-13 (pbk): 978-1-4842-1393-3 ISBN-13 (electronic): 978-1-4842-1392-6
DOI 10.1007/978-1-4842-1392-6

Library of Congress Control Number: 2016943372

Managing Director: Welmoed Spahr
Acquistions Editor: Louise Corrigan
Development Editor: Gary Schwartz
Technical Reviewer: Brandon Scott
Editorial Board: Steve Anglin, Pramila Balen, Louise Corrigan, James DeWolf, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham, Susan McDermott, Matthew Moodie, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing, Matt Wade
Coordinating Editor: Nancy Chen
Copy Editor: Mary Behr
Compositor: SPi Global
Indexer: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springer.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Printed on acid-free paper

To those who share their time, effort, and work with all of us to make the world a better place, free software is only the beginning.

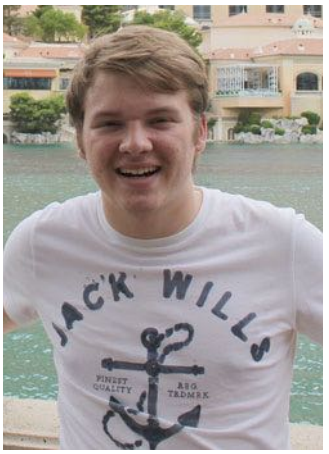
Contents at a Glance

Contents

About the Author

Jose Dieguez Castro is a senior systems administrator currently working as a freelance consultant. Joe has worked on a wide range of projects from small to large infrastructures in both the private and public sector. When asked about his specialty, he answers, "Getting the job done." He also likes to think of himself as a developer who cares too much about free software. Photography, sports, music, and reading are the way he frees his mind from work. He can be reached at `jose@jdcastro.eu`.

About the Technical Reviewer



Brandon Scott specializes in software engineering for desktop applications, software development kits, and distributed systems. He currently leads the development efforts for AspiraCloud Ltd., focusing on Microsoft SharePoint and Azure workstreams. Additionally, Brandon also partners with Razer Inc., aiding with the design of SDK products and open source libraries. He built his experience working for a variety of companies in different industries, such as JPMorgan Chase & Co. and Microsoft.

Acknowledgments

First of all, my thanks go to Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas Mcllroy, and Joe Ossanna for UNIX. Thanks to Richard Matthew Stallman for the GNU Project, the Free Software Foundation, Emacs, and the Free Software movement. Thanks to Linus Torvalds for the Linux kernel. And thanks to all of the corporations and community developers that for years have contributed to the development, documentation, and adoption of the Linux operating system and its distributions. These people not only helped to create the Linux ecosystem, but they also helped to make the world a better place for sure. Without all of these individuals, this book would never have been possible.

Thanks to my family and friends for their unconditional love and support.

Thanks also to the Apress team, especially Louise Corrigan, who convinced me to write this book; Nancy Chen and Jill Balzano, for their patience and hard work in coordinating this book; my editors, Gary Schwartz and James Markham, who were kind enough to deal with my broken English; and my technical reviewer, Brandon Scott, for righting my wrongs.

Most of all, my thanks go to you, the reader, for taking the trouble and using your valuable time to read this book.

Introduction

What is a Linux distro?
Why are there so many of them?
Which one should I choose?
Which one is the best?
Why are they so hard to use?
Why are they so ugly?
Do I have to manage all of this with text commands?

If you tried to answer these questions by yourself, or sought help from others, you probably did not get a clear answer—or maybe you received several contradictory ones. The truth is that no unique answer exists for many of these questions. If you want to deal with the huge jungle of Linux distros, you need to get some things clear first. You have to learn the basic concepts and then isolate some objective facts to make an informed decision.

In this book, I show you what makes up a Linux distribution. Before that, however, I guide you through a journey in time to explore the origins of Linux itself and of free software. It's easier to understand the concepts behind a Linux distro once you know how it all began. Next, I point you to the fundamental criteria that you must consider in order to compare and choose a Linux distro. I point out which of these criteria are based on relative or absolute concepts and which are simply personal preferences. Finally, to understand why Linux has a plethora of distros instead of just a few versions like other popular operating systems, I show you the family tree of Linux distros; there are only a handful of original distros but myriad relatives.

The core of this book is the detailed analysis of ten Linux distributions, from the most popular to the most innovative. I chose them to show you a variety of distros and how they can be so similar and yet so different, from the ones suited for Linux novices to the ones tailored for Linux experts. I show you the criteria that you need to consider for each distro, and a compilation of the pros and cons of each. Even when I try to be objective, it is inevitable that here and there I introduce some personal viewpoints, but you should take into consideration only the cold facts to make your own decision; your needs and mine might be completely different.

I also show you how to install and conduct basic maintenance tasks for each Linux distro. Don't get me wrong; I do not pretend here to instruct you on how to install and administrate a Linux distro in a professional setting; that would take another book. I have two purposes here. The first one is to give you a hint of what are you going to find in each distribution and how they differ, and a book like this one is a good way to gain that knowledge. The other purpose is to debunk the rumors that Linux is very hard to install, use, and maintain; in fact, there are many friendly distros. I provide you with an introduction to how to install those distros that are commonly considered too hard and appropriate only for experts. Finally, I briefly summarize other distros that may appeal to you so that you will have a wider view of the Linux spectrum.

Linux is not only an operating system for the desktop; there are other environments where it is used. Furthermore, there are many distros that were built with only a particular task in mind. I introduce you to the other side of Linux distros: the task-oriented side.

As it is beyond the scope of this book, I also offer you a bonus chapter, available only online, where I introduce you to other operating systems that are not Linux, but that share many similarities with Linux distros.

I hope that *Introducing Linux Distros* will answer many of your questions about Linux and will help you pick the best one that is right for you. To that end, I did my best and I hope that you enjoy reading this book.

—Jose Dieguez Castro

PART 1

Linux Distros

In the first part of this book, you get a good perspective on Linux distros and what you must consider when comparing the various distros in order to select the right one for you. You also learn about the origins of the Linux distros and the reasons why there are so many and how they are related.

In the first chapter, you learn about the nature of a Linux distribution. In order to do that, you need to understand what Linux is, and the best way to accomplish this is to learn about its origins and a bit of its development history. Basically, I present the minimal amount of information necessary for you to put Linux and its distros in context.

The second chapter is a compilation of points you must consider in order to compare the performance and features of two or more Linux distros. Some of those points cover subjective issues, such as personal preference of one aesthetic over another, or if you like one desktop environment in particular. Nonetheless, you will have a checklist to use as starting point.

The third and final chapter in this part of the book covers the family tree of Linux distributions. This information is provided so that you know how they are related to each another (at least the more relevant distributions) and to get an idea of why there are so many of them and why one particular distro has so many derivatives.

CHAPTER 1

Deconstructing a Linux Distro

One of the first things that surprises new Linux users who come from other operating systems such as Windows or Mac OS is that a single entity called Linux OS does not, in fact, exist. Linux is only a kernel, not an OS, and what people generally perceive as Linux OS is actually a Linux distribution, and more precisely a GNU/Linux distribution. The Linux distribution concept is very popular, and it is commonly known as a *Linux distro*. This is the term that I use throughout the rest of the book. Don't be put off by all of these terms: I cover them in detail in this chapter.

After learning that what you thought was Linux is actually a Linux distro, here's another surprise: there is not just one unique Linux distro. Rather there are many, perhaps a few hundred, Linux distros available. Thus the task of choosing the right Linux distro for your needs can easily become an overwhelming task. Nevertheless, this first step can be critical, and there are several approaches for accomplishing this this:

Note There are more than 700 Linux distros available at the time of this writing, but only about 300 are currently actively developed and supported.

- One way to choose a Linux distro is to follow the advice of a colleague, the recommendation of a professional, a review that you find on the Internet, a random choice, or because it is the only version you know about at the moment. If the experience is pleasant, you may wind up using that distro for a long time.
- Sometimes the experience of choosing a Linux distro is not pleasant at all. The problem could be the installation process, the desktop environment, the terminal, or the user documentation. Any of these issues might make a particular distro not right for you. From there, the process could follow one of these common paths:
    - You become angry and disappointed, so you give up on using this OS, possibly forever, or at least for a long time. You may take to the Internet to criticize the Linux distro that you tried, and you may warn your colleagues against it. Unfortunately, you might have turned into a long-term Linux user if you had only chosen the right distro at the start.
    - Let's say that you decide to try another Linux distro. After a few tests, you find the one that seems to address your expectations. You turn into a long-term, happy user of that particular distro, and you do not hesitate to test other ones.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-1-4842-1392-6_1) contains supplementary material, which is available to authorized users.

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_1

- Alternatively, you try several distros, maybe even a lot of them, but you never find the one with which you feel comfortable. The problem could be that you haven't defined your needs or wants specifically enough, and so you haven't discovered which distros can cover these needs and in which manner. You may end up as a long-term, not-so-happy Linux user, or you may switch to another OS completely.
- Finally, you may be someone who thoroughly enjoys testing new Linux distros. You may use one Linux distro on a regular basis, but test new ones on a different machine or VM, or you may switch Linux distros every few months or every few years. You are probably a long-term Linux user.

As you can see, the decision-making process for choosing a Linux distro is very important. The Linux experience can be radically different from one distro to another, and finding the right one for you may be hard. This book aims to help you in process of finding the Linux distro that best fits your needs. A lot of trouble, disappointment, annoyance, and unpleasantness can be avoided if you use objective selection criteria. Obviously, previous experience, expectations, and an open mind can help with the adoption of a Linux distro. The goal of this book is to help you make your ultimate decision, or at least take your first steps in selecting a Linux distro, and then make working with Linux OS a pleasant experience.

■ **Note** One of the easiest ways to choose a Linux distro is to use a service like `http://distrowatch.com`. The site provides rankings and a brief description of the most common distros. The differences among distros are very important. By the way, what distro do I use? I am writing this book using a virtual machine to run Windows 8 and Microsoft Word, but the underlying foundation is an Arch Linux installation managing it all.

Linux Distro

What exactly is a Linux distro? A *Linux distro* is a set of components that are required to achieve a working Linux system, and the processes needed to install these components to achieve a running system. Today, a Linux distro includes the Linux kernel itself (different versions of the kernel may be integrated into the same distro for different hardware architectures, for example), the omnipresent GNU tools (more about this later), a lot of small tools that are needed to provide different services, probably a windowing system, a desktop environment, and finally a package management system with a number of software packages that can be installed by default or according to user preferences. Even when the entire installation process can be completed (more or less) through the automated distro installation process, as opposed to other operating systems, the Linux OS installation is highly customizable. A particularly unique aspect of a Linux distro is the user documentation that is provided. Even when the documentation for the numerous individual packages that comprise a distribution is delivered, many distros provide nearly complete documentation about the distribution itself, primarily relating to the installation process or how to contribute personally to the distribution.

To choose a Linux distro, you must first learn exactly why they exist and why there are so many of them. To achieve this goal, we must first dig into the origins and history of Linux itself and the circumstances surrounding its creation.

The Origin of Linux

As is the case with many successful software projects, Linux started as a way to solve a particular problem faced by one software developer. In this case, in the early 1990s, a Finnish student at the University of Helsinki named Linus Torvalds (see Figure 1-1) wanted to be able to use at home an operating system similar to the one he used on campus. Alas, there was no option available that could satisfy his desire. He was using a version of MINIX at home, but it would not run on his PC 386 machine, so he decided to create his own OS. On August 25, 1991, he posted the following message on the `com.os.minix` newsgroup site, which would become one of the most famous quotes in computer history:

> *Linus Benedict Torvalds*
>
> *Hello everybody out there using minix -*
>
> *I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).*
>
> *I've currently ported bash (1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)*
>
> *Linus (`torv…@kruuna.helsinki.fi`)*
>
> *PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.*

Figure 1-1. *Linus Torvalds, creator of Linux, at the LinuxCon Europe 2014 in Düsseldorf*

■ **Tip** If you are interested in a first-hand history of Linux as told by its creator, read *Just for Fun: The Story of an Accidental Revolutionary* by Linux Torvalds (HarperCollins, 2001).

This is the beginning of the history of Linux, and you can be sure that nobody at that time, not even Linus Torvalds, could imagine that someday his pet project would become one of the most ubiquitous operating systems in the world. These days you can find Linux almost everywhere. It is installed on 97% of the 500 fastest supercomputers in the world[1], on the servers that run the Internet, and on smartphones and tablets (Android OS); it is also embedded in many consumer electronics products, cars, and so on. In fact, the of two most advanced, complex, and expensive objects/projects that humanity has produced, the International Space Station and the Large Hadron Collider, are also managed by Linux. This is ironic because if you look back to the time when Linux was born, you will discover an almost identical context (absent the yet-to-be-discovered technologies), but substituting UNIX (the OS that inspired Linus) for Linux.

■ **Note** The first name of the Linux OS was Freax, as a combination of "free," "freak," and "x" (for Unix). Later, a maintenance technician for the FTP server at the University of Helsinki decided to name the folder where the OS was allocated as Linux (for Linus' UNIX). Torvalds, who had considered and dismissed that name (it seemed much too egotistical to him) finally consented to keep the name. Clearly, Linux is a better name than Freax.

Let's go back to the early 1990s and talk about the operating system situation at that time. There were two clear separations in terms of operating systems: personal computers and everything else.

- The personal computer market was dominated by two players: Microsoft Windows (Windows 3.x and MS-DOS 5.0) and Apple Macintosh (System 7). There were, of course, other players, but they were minor in terms of market share: AmigaOS, Atari TOS, IBM DOS, DR-DOS, OS/2, BeOS, MINIX, Xenix, and so forth.
- In the remaining areas of computing (professional workstations, minicomputers, mainframes, and so on), there were a number of other contenders:
    - **UNIX-Based**: NeXTStep, IRIX, Xenix, AIX, SunOS, Ultrix, HP-UX, SCO, BSD, and UNIX. There were many operating systems based on UNIX, and the number was growing. Also, there was an increasing number of migrations and new installations of these operating systems in a variety of machines. This family of operating systems likely had the largest market presence at that time.
    - **Others**: Multics, OS/8, TOPS-X, OS/400, and so on. Today there are still machines running some of these operating systems; however, a lot of installations have migrated over time to a UNIX-based OS.

As you know, today the Microsoft Windows and Mac OS operating systems are the main players in the personal computer niche. In other computing areas, however, the UNIX family still has the biggest market share. Moreover, at this time, Linux is the most common UNIX variant, and it is growing faster and replacing a lot of older UNIX installations. So, why has Linux succeeded so well in the face of so many alternatives?

[1] www.top500.org/statistics/details/osfam/1

There are four main reasons for Linux's success:

1. **Linux was based on the existing UNIX OS**. Linus Torvalds was already using MINIX at home, and he wanted to use a similar OS to the UNIX-based one installed on the campus' mainframe at the university he attended. As mentioned, buying a new computer with an 80386 processor forced him to develop his own OS since he wanted to use a UNIX-based OS on that machine. Given the success and broad presence of the UNIX family of operating systems in the world, there were a lot of people willing to adopt a new alternative for their personal computers. There were only a few ones back then, MINIX and Xenix being the main ones, and each had its disadvantages.

2. **Linux was aimed at personal computers**. As mentioned, the main UNIX-based operating systems available for microcomputers at that time were MINIX and Xenix. (There were other minor actors, such as the IBM PC/IX, Venix, Coherent, and Idrix). But MINIX and Xenix had the following inconveniences, which opened the door to other competitors:

    - MINIX had a low-cost license, and the source code was made available by its developer, Andrew S. Tanenbaum (a computer science professor), who created it for educational purposes. However, it was only available for the 8086 PC and the 80286 PC AT machines, and its license prevented use of its source code to develop a new OS based on it.

    - Xenix, originally developed by Microsoft, was a Santa Cruz Operation (SCO) product at that time. Even when it was available for 386 systems, its license was very restrictive and expensive (it originally was an OEM product) and its source code was not accessible. This UNIX-based OS, however, was the most commonly installed one in the personal computer market.

    - Simultaneous with Linux development by Torvalds and others, there were a number of people working to port a UNIX version called the Berkeley Software Distribution (BSD) to 386 machines. This resulted in a series of BSD-based operating systems for personal computers. Torvalds didn't know at the time about the BSD project. If he did, perhaps he would never have started to develop the Linux kernel. (He has stated this many times in public.) I will cover this topic more in Online Chapter.

3. **The way in which Linux was distributed: free and open source**. Since Torvalds had few resources (and thought that $169 for MINIX was way too much), if he wanted others to use it, comment on it, and perhaps improve upon it, the best way to distribute Linux was to give it away for free. Torvalds developed his Linux kernel in a MINIX OS environment using the GNU C compiler, and later included the GNU tools, so you were obliged to use the GPL license (more on that later) and distribute its source code. This fact, plus free distribution, helped propel the early adoption of Linux by a bunch of "hackers" (in the original meaning) who were willing to play with the kernel and the code. Heretofore, you could not play with the Xenix code, and even if you could contribute to MINIX or make your own version or port of it, you could not distribute it freely and legitimately. (There were, in fact, several unofficial ports of MINIX to various machines.)

4. **The Internet**. The Internet was essential to the success of Linux. Without the Internet, e-mail, FTP, and newsgroups, it would have been impossible to join so many people together and to be able to work collectively on a project like this. People from all over the world became enthusiasts of Torvalds' pet project, and they started to collaborate voluntary in its development. The number of Linux supporters grew at an astounding pace, and today there are thousands of developers and numerous enterprises working together to make Linux a better OS with each new release. Shortly after its introduction, the original assembly code for Linux was ported to C by Torvalds and others, and soon ports would appear for machines other than the 386.

■ **Note** If you are curious about what the first release of the Linux code looked like, you can download it from `www.kernel.org/pub/linux/kernel/Historic/linux-0.01.tar.gz`.

Torvalds initially began by developing a few utilities for MINIX OS, including a terminal emulation program to connect to the university's mainframe, a disk driver, and a file system drive. Soon he realized that if he continued along this path, he would wind up making his own OS, so he made the decision to get rid of MINIX and make a better UNIX-based OS himself.

On October 5, 1991, less than two months after he started work, Torvalds made the following announcement about version 0.02 of his kernel in the `comp.os.minix` Usenet newsgroup[2]:

> *Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-)*
>
> *As I mentioned a month (?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.*
>
> *Linus*

Soon many other programmers joined to the cause. The quality and fast pace of the kernel development was achieved in a naïve and very unusual way—simply by releasing a new version on a weekly basis and getting feedback from users and volunteers. Thus it evolved in the strictest definition of the word: by selecting only the changes and improvements that worked and discarding the rest. As a result, by the end of 1993, the Linux OS was stable and reliable enough to compete with almost all of the commercial versions of UNIX, and it even began to serve as the foundation of a lot of commercial applications. In a short period of time, Linux became the predominant UNIX alternative.

As mentioned, the rate of growth of Linux was amazing, and by March 1994, Linux version 1.0 was released. In a brief span of time after the release of the first Linux distributions (1992-1994), it became a component of the toolbox of numerous enthusiasts, professionals, and academics.

---

[2] `https://groups.google.com/forum/#!topic/comp.os.minix/4995SivOl9o`

Torvalds continued to lead the development effort, and he is still making the final decisions on the shape of the Linux kernel today. He now works at the Linux Foundation, a non-profit institution created to promote, care for, and standardize the development of the Linux OS. It is supported by a number of leading Linux and open source companies around the world as well as by individual developers globally.

It's important to remember that Linux is only a kernel, not a complete OS, so Torvalds put together this kernel and the tools from the GNU project to realize that first version of the GNU/Linux OS. But what was the GNU Project?

■ **Note** A kernel is the core, or central, component of an operating system. It basically connects the hardware and the application software. There are two main architectures involved in constructing a kernel: the microkernel (a lot of small pieces) and the monolithic kernel (a unique large piece).

Without getting too deep into those concepts, a curious historical aspect of the Linux kernel is that was developed as a monolithic kernel, while MINIX was based on a microkernel. This was the reason for a comment made about Linux by A. S. Tanenbaum (the original developer of MINIX) in 1992, which stated that Linux was obsolete by concept. That spurred a debate between Tanenbaum and Torvalds, and demonstrated Torvalds' obstinacy, which has become legendary, as he continued to support the monolithic kernel architecture in his OS.

Tanenbaum also maintained that Linux would become obsolete in a few years, and the GNU Hurd kernel (a sort of UNIX-based microkernel) would be its replacement. It turned out that he was completely wrong about this prediction. Linux is more alive than ever and GNU Hurd is still under development and never delivered on its promise.

The Linux mascot, shown in Figure 1-2, has a funny history behind it. The community was looking for a logo/mascot, and after several suggestions, Linus talked about being bitten by a "ferocious fairy" penguin in an Australian zoo in 1993. Thus it was settled. Larry Ewing made the original drawing. The name Tux came from **T**orvalds **UNIX,** and it's also the beginning of the word tuxedo (frequently associated with penguins).

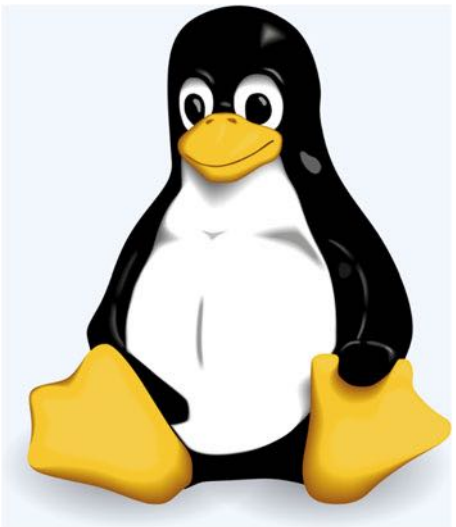


Figure 1-2. *Tux, the official mascot of the Linux kernel and its de facto logo*

GNU Project

Earlier I mentioned that Torvalds is kind of stubborn, and his obstinacy was in play in retaining the kernel architecture of Linux, a major factor in its success. The GNU Project exists because of the obsession and determination of another legendary character, Richard M. Stallman of MIT, who resisted culture changes in software development, deciding to turn back the clock to its origins.

Hacker is a word that has been abused and distorted from its original meaning. Almost no one outside of the computing world knows the real concept behind that term, and even a lot of people in the IT sphere don't realize its true meaning.

The hacker subculture was born in the 1960s at MIT (Massachusetts Institute of Technology) by a very clever group of people (engineers and physicists) who found solving day-to-day problems with the computer systems of that time to be fun and intellectually rewarding. They discovered workarounds to avoid and circumvent the limitations and failures of those systems, and they challenged themselves and others to extend the limits of those computer systems and achieve smarter paths to doing so.

The goal was to always achieve excellence and have fun in the process, and the pranks among them were a great part of that culture. The quintessential part of the hacker's subculture, however, was the sharing of information and achievements, rather than keeping these to themselves. This was done in part for pride and glory, and in part because they wanted to help to improve the world. They saw themselves as a sort of a mix between rebel heroes and mythical Robin Hoods, willing to give away their spoils. They had a strong sense of ethics and responsibility, even when they seemed to act as children in the eyes of computer corporations and institutions. This subculture expanded to other areas as well. In Silicon Valley, for example, one of the icons of this movement was the Homebrew Computer Club, a computer hardware hacker's club, where later the first Apple computer was presented by member Steve Wozniak.

> *"What they had in common was mainly love of excellence and programming. They wanted to make their programs that they used be as good as they could. They also wanted to make them do neat things. They wanted to be able to do something in a more exciting way than anyone believed possible and show 'Look how wonderful this is. I bet you didn't believe this could be done.'"*
>
> —Richard Stallman in the 1985 TV documentary, *Hackers: Wizards of the Electronic Age*

It was this subculture that gave birth to some big technological achievements like the UNIX OS, the TCP/IP protocol (the backbone of the Internet itself), and the GNU Project. You can honestly say that Linux was a byproduct of the last strains of that original movement, before it evolved into a somewhat different and wider approach, the Free Software initiative. Going back to the GNU project, however, Stallman is sometimes called "the last of the true hackers" (he even saw himself as that for a period of time), and by that I am referring to original group of hackers at MIT.

■ **Tip** If you want to know more about the origins of hacking and computers as we know them today, you may be interested in an excellent book called *Hackers: Heroes of a Computer Revolution* by Steven Levy (Anchor Press/Doubleday, 1984). It is the best and most accurate chronicle about those early and interesting times.

Richard Stallman (see Figure 1-3), a mathematical prodigy and student at Harvard in 1970 (he would graduate from Harvard *magna cum laude* in Physics in 1974), soon joined MIT's Artificial Intelligence (AI) Lab and became part of this subculture. He had grown up as a professional systems programmer in that environment, and he had internalized those ethics so much that they become part of his way of living.

Figure 1-3. *Richard Stallman, founder of the GNU project, at Oslo, Norway on February 23, 2009*

In the early 1980s, however, a series of events would break down that environment and destroy the idealism and freedom that had so influenced Stallman's life. So in 1983, Stallman decided to start a new project called the GNU's Not Unix (GNU) Project to try to create a completely new operating system that could be freely distributed, inspired by the hacker subculture and ethics. His initial goal soon became a full-time job, so he abandoned MIT in 1984 and began writing the GNU software. He also wanted to avoid the possibility that MIT could claim any rights to his code. Two of the first programs he wrote were the multi-language and multi-platform gnu compiler, or gcc, and the Emacs text editor. These two programs are still in use today. gcc is one of the most used software compilers in the world, and Emacs is still the favorite text editor of a lot of computer professionals (myself included).

To attract people to help him develop the GNU OS, he published the GNU Manifesto in 1985, which would become one of the pillars of a software revolution, six years before the initial Torvalds message to the `com.os.minix` newsgroup. This manifesto stated some fundamental liberties that should apply to all software:

- Freedom to run a program for any purpose.
- Freedom to study the mechanics of a program and modify it to suit your needs.
- Freedom to redistribute copies free or for a fee.
- Freedom to improve and change modified versions to use by everybody.

As more people read the manifesto and began embracing its principles, a set of new needs appeared: a way of sharing and distribution and a way to preserve the hacker's ethics that Stallman so much appreciated. FTP and traditional mail solved the distribution problem, and the creation of the GPL solved the ethical issues and legal matters. *GPL* stands for General Public License, and Version 1, written in 1989, was a compendium of similar attempts made in the early versions of GNU Emacs, GNU compiler, and GNU debugger. The license basically solved two problems:

- **The distribution of the software as a binary**. The GPL prevented this by forcing the distribution of the software under this license, which is always accompanied by the source code in a human-readable format.
- **Avoiding additional restrictions**. Under the GPL, the software can be combined with other software programs that have license agreements with minor restrictions in them unlike the GPL itself, but never with software programs with more restrictive licenses unless the whole package is distributed under the GPL.

The GPL was a complete success, and later the Linux OS would adopt the GPLv2 and become one of its major beacons throughout the world. This license would eventually be complemented with the LGPL in order to support the software libraries. The third, and current, iteration of both licenses was adopted in 2006. Some believed that it was too restrictive; Linus Torvalds and other notable Linux developers were some of the most vocal critics. However, it's difficult to calculate the exact number of programs developed under this license; estimates are that the GPL family of licenses cover about 55% of all free software available, so it has clearly had a huge impact.

■ **Tip** If you want to read the original GNU Manifesto, you can do so at `www.gnu.org/gnu/manifesto.en.html`. You can read the original version of the GPL License at `www.gnu.org/licenses/old-licenses/gpl-1.0.html`.

As stated earlier, the GNU Project (see Figure 1-4) was a main contributor to the Linux OS, as a complement of the kernel, in the same way that the Linux kernel covered the lack of one in the GNU project, because the Hurd kernel was still unavailable. (The first "usable" version would appear in 1996.)

Figure 1-4. *The logo of the GNU Project is obviously a gnu*

In the early days of Linux, these GNU tools made up a large percentage of the OS (the GNU Project produced thousands of utilities like the shell, Emacs, the GIMP, GNOME, and the initial file system). Stallman always remembered this, and he claims that the proper name for Linux is GNU/Linux (some distros, such as Debian, still maintain this name) to recognize the priceless contribution of the GNU Project. The name suggested by Stallman and the Free Software Foundation never had any great impact. I personally just call the OS Linux, as do most people today.

However, it wouldn't be fair not to recognize that without the GNU Project, the Linux OS may have never existed. But it would also be reasonable to state that the GNU tools are today a minor part of the Linux OS, and with the great changes that the cloud, virtualization, and the systemd project (a new init system for Linux that is slowly replacing a lot of tasks heretofore done by a collection of small tools) are making in the OS, perhaps in the future the GNU Project elements will be completely gone from the Linux OS.

Thus the GNU project was critical for the development of Linux, and it was the key to arriving at a new way of developing software. It is certainly the reason why we had so many software projects and developers in those days. It was clearly a revolution, but it is sometimes is forgotten, underestimated, and even criticized. I'm absolutely sure, however, that if the GNU Project had not existed, the software world today would be radically different from what we have now.

The Birth of Linux Distros

Once the Linux kernel became a reality and the GNU tools formed the missing piece to build the entire operating system, you still had to assemble, compile, and package them. It was a difficult, complex, and large job, well suited to enthusiasts ("geeks," as we call them today). However, if your intention was to introduce the OS to a larger audience, it was clearly necessary to simplify this process. With the development of a number of new programs built with the new OS in mind by an ever-increasing army of new developers, this need became an urgent one. Some of those enthusiasts developed the first Linux distros initially for their personal use, but they soon started to share them because they quickly realized that the job required a large number of interested individuals.

Thus, in 1992, the first Linux distributions were created. None of these distributions (MCC Interim Linux, Yggdrasil, H. J. Lu's "Boot-root," and Softlanding Linux Systems) still exist, but in 1993, the first two of the great Linux distributions were born: Slackware and Debian. Slackware was based on the SLS (Softlanding Linux System) and Debian was a new creation by Ian Murdock, because he was dissatisfied with SLS distribution. New distributions kept appearing, such as Red Hat, SUSE, Caldera, Conectiva, Mandrake, and so on, until the "big expansion" at the start of the new century when a large number of new distros were introduced.

I cover this history in a detailed manner in Chapter 3 and also in the subsequent chapters dedicated to the major Linux distros. Those distros were a convenience to less-experienced users, but they also met the basic requirements of the most experienced hackers, who adopted them even if they chose to work with more complex versions specifically suited to their needs.

As stated at the beginning of the book, there are an astounding number of distributions, and many of them offer different versions for different architectures, purposes, tastes, and so forth. You may ask yourself, "Why so many distros?" Read on! The answer to this question follows.

Why So Many Distros?

The reasons behind the awesome set of different Linux distributions are in the roots of the Linux OS itself and the GNU Project. The conditions that cultivated the success of the Linux OS also nurtured the huge ecosystem that we have today with Linux. The possibility of changing nearly anything within the OS and distributing those changes for free contributed to the tremendous growth in the installed base of Linux users. Anyone who didn't like how something worked could simply change it, assuming they had the knowledge, motivation, and means to do so. Add to this the tweaker mentality of the hackers and you have what is available today: a distro for every taste.

New distros appear every year. And a vast number of nonconformist Linux users still like to tweak every minor aspect of their Linux installations. As if that isn't enough, some of them take it a step further: it is entirely possible to make a one-off Linux distro (and there are many examples).

While the majority of Linux distros are sustained by a group, or perhaps even a community of developers, most of the major distros are supported by corporations with a number of employees dedicated to its development.

> *The GNU Project and the GPL represented just the beginning of a new, larger, and more important movement: Software Libre, which translates to "Free Software." When Stallman started the GNU Project, he also introduced the concept of "copyleft" (as opposite of "copyright"), which would become the major component of the GPL and some other licenses with similar concepts of more or less restrictive use. As Stallman described it, "The central idea of copyleft is that we give everyone permission to run the program, copy the program, modify the program, and distribute modified versions — but not permission to add restrictions of their own. Thus, the crucial freedoms that define "free software" are guaranteed to everyone who has a copy; they become inalienable rights. For an effective copyleft, modified versions must also be free. This ensures that work based on ours becomes available to our community if it is published."*
>
> —Richard Stallman in *Opensources: Voices from the Open Source Revolution* by Chris Dibona (O'Reilly Media, 1999)

In 1985, as a part of the plan to support the GNU Project, Stallman founded the Free Software Foundation (FSF), a non-profit organization with the goal of supporting the development of free software in general. The GPLs that FSF created soon went viral in the programming community, and the movement surrounding free software began to grow. This movement quickly gained mindshare in the software world, and those who generally liked the idea but not the restrictions imposed by the GPL created other licenses such as Apache, BSD, MIT, Public Domain, Mozilla, and so on. In 1989, a group of people, including Bruce Perens and Eric Raymond, tried to unify the BSD and GPL licenses under the umbrella of a new movement called the *Open Source Initiative* (OSI). Others used the term FOSS, for *Free and Open-Source Software*, as a means to unite both movements. In any event, the fact is that the mindset of sharing and opening up source code is here to stay, and the number of developers writing code under these terms will not cease to grow. Software forges[3] and central repositories like GitHub, Bitbucket, GNU Savannah, and others are testimonials to the vitality of the Free Software movement. A large part of the Internet itself is built on the foundation of Free Software projects.

Although Free Software today has not achieved the high level of ethics and the culture of the early hacker's movement, and although software patents are a constant menace to its goals, it is true that it provides us with the privilege of using and enjoying a great number of software projects that we can fork[4] and modify at any time.

This is the environment in which a great number of communities continue to develop Linux distros, creating new ones daily, and without the Free Software movement, and if all software were closed-source, doing so would literally be impossible. Contrast this with the Microsoft and Apple operating systems. Linux represents more than an OS; it is also about freedom. Even though most people refuse to use the term GNU/Linux, the truth is that the success of Linux is also the success of the GNU Project, the FSF, and Free Software.

---

[3]A web-based platform designed to host and share computer program source code, generally FOSS (Free and Open Source Software). You can learn more about it at `https://en.wikipedia.org/wiki/Forge_(software)`.

[4]Fork in the sense of bifurcation. It's a common term used in programming when a copy of a source code is made to work on its own as an independent project. You can learn more at `https://en.wikipedia.org/wiki/Fork_(software_development)`.

■ **Note** You may have noticed that the term "Software Libre" was used before. This term is preferred by some people, and Richard Stallman himself has also used it, because the word "libre" in Spanish avoids the ambiguity of the same word in English (free). Those who prefer Software Libre simply do not want you to confuse free, as in a free beer, with free, as in free speech. Software Libre is always linked to freedom, but it could be commercialized and sold, as it frequently is. You can learn more about this at https://en.wikipedia.org/wiki/Gratis_versus_libre.

Do I Have to Choose One? Aren't They All the Same?

This is a common question. Why choose a particular Linux distribution? Can't you achieve the same goal with all of them? The answer is clear and simple: NO**.**

There are many different reasons for choosing a Linux distribution, just like there are many different reasons and ways to create one. Yes, you can tweak and customize almost any distro, but the effort to do so can be huge in some cases, and sometimes it's nearly impossible to achieve your goal.

If you need to embed a Linux distro in an electronic device with scarce resources, for example, you cannot simply install the last Ubuntu distro for desktops because it does not work that way. Similarly, if you try to use the elementary distro as a Linux server, it probably will not end well, and you will get a huge headache for your trouble.

Summary

I hope that you now have a clear idea about what a Linux distro is and the reasons leading up to the current multitude of available Linux distributions. Perhaps you were wondering why I was spending so much time talking about the GNU Project and Software Libre when this is book about Linux. However, by now you should understand that Linux is linked by birth, nature, and definition to the GNU Project and the Software Libre movement.

The more important point, and the theme of this book, is that it is impossible to choose just one Linux distribution for all situations. This is precisely the reason behind this whole book; you would have nothing to read and I would have nothing to say about this topic if the choice of a Linux distro did not matter.

I cover the decision criteria for selecting a Linux distro in the next chapter in order to help you understand the variety of purposes, tasks, tastes, and so forth that form the basis for choosing a Linux distro. The rest of the chapter explores the different factors that have to be taken into account when choosing the right distro for your needs.

CHAPTER 2

Linux Distro Selection Criteria

Now that you know what a Linux distribution is and why there are so many, the next step is to find a way to compare and screen them. The best way to do this is by choosing a series of factors that you can use to measure and classify the various Linux distros objectively, without bias in your reasoning. It is OK to listen to the opinion of a colleague or a Linux expert, or to read an article or review, and so on, but it is very rare to find a source that is completely unbiased or that matches your particular situation. I strongly advise you to take these factors into account and weigh the most important ones based on your needs. Then you can choose the distribution that is the strongest based on those needs. This book will help you make an informed decision.

In this chapter, I name some distributions in each category, but don't take this as a way to easily choose the right distro (also, I don't name all of the distributions that fit for a particular point). You should choose your Linux distro on the basis of a number of factors, not just one (except in very particular cases like task-oriented ones). This goal of this chapter is to help you evaluate each distribution by knowing what factors you should consider when doing so.

In the next chapters I introduce you to many distros in detail. I have chosen a sample of the most popular, reliable, and secure, but also a few ones to illustrate some points like the current state-of-the-art Linux distributions or ones that fit particular needs. In certain chapters I refer back to the factors that I mention here to see how each distro rates. Although it is impossible to cover all of the alternatives available to you, I try to cover as much ground as possible.

Also, note that I do not take a deep dive technically. I think that is beyond the scope of this book. There are some technicalities and concepts that I must cover to give you precise information, but if you are new to Linux, I don't want you to get distracted or intimidated by me being too technical. When I analyze the different distros, I will get as technical as necessary to explain how they perform.

Note I mention many factors to consider when choosing a Linux distro. By the end of the chapter you may have come up with some factors that I did not mention. Please note that I covered the most critical ones for making the right decision, and I discarded some because I think they overlapped ones that were mentioned or were less specific.

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_2

Factors to Consider when Choosing a Linux Distro

Figure 2-1 shows the important factors to consider when choosing a Linux distro. The first step in choosing a system, whatever it is, is to ask yourself how you are going to use it. Using your new operating system to compose letters and track your finances is not the same as using one to serve a marketplace on the Internet. You could use the same distro for both purposes, and some people do, but it most cases this would be a very bad decision. Even if you are new to computers, you probably know that is generally not a good idea to use the same tools for domestic purposes and for professional work.

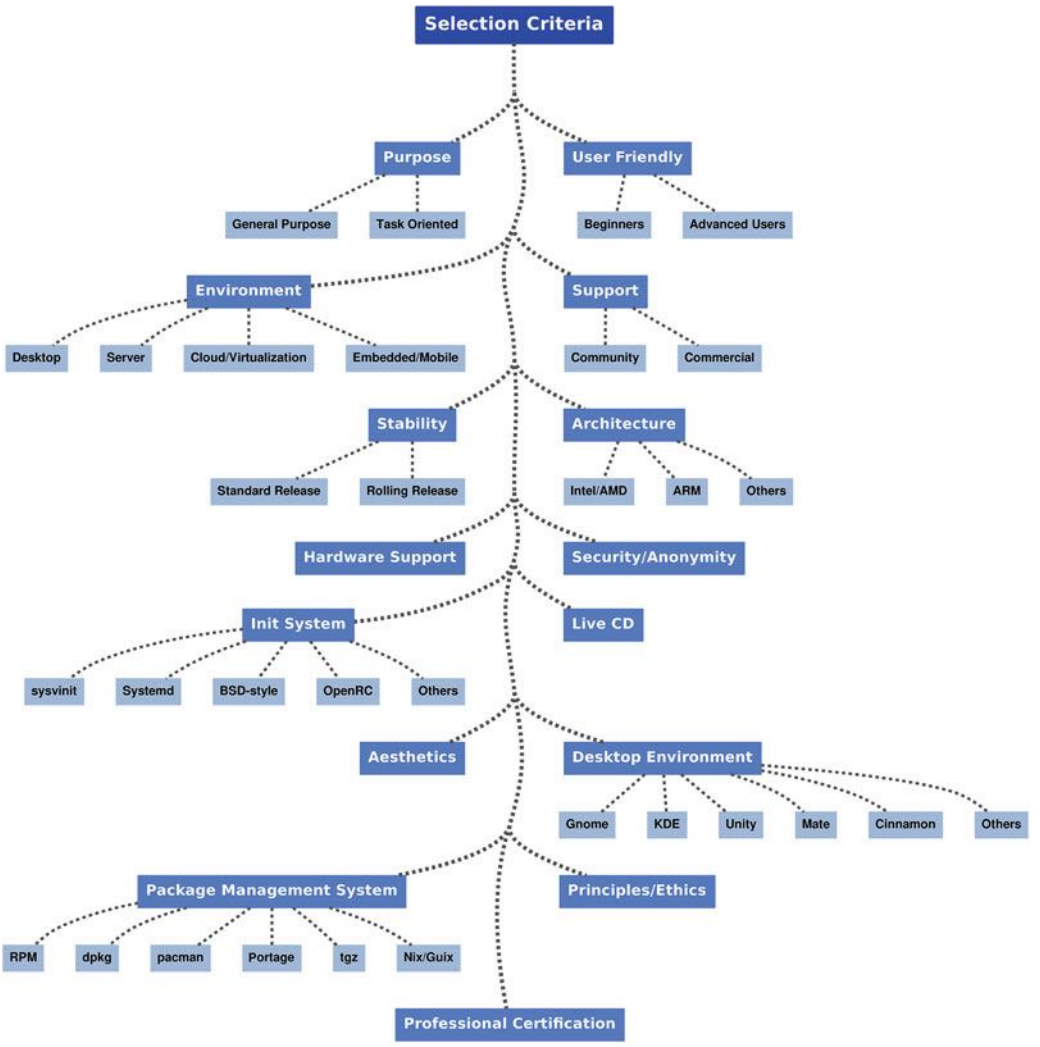


Figure 2-1. The Linux distribution selection criteria graph

Let's divide the category of basic workflow into two large categories: general purpose and task-oriented.

General Purpose

We can include in this category the majority of the Linux distros and we can think about them as the equivalent of the other big competitors of Linux: MS Windows and Apple OS X. They are not oriented to doing any specific task; rather they try to fulfil common and diverse jobs from writing documents to editing videos, etc. These tasks are what everyone thinks of when they think about an operating system. The majority of people who want a Linux distro for their own use also want to perform such tasks.

You expect a distro to be able to use whatever category of software you need for your daily work at home or work. In this category are the most popular distros: Ubuntu, Fedora, Debian, and Mint.

Task-Oriented

Why would you need a specialized Linux distro? This is an interesting question, and there is no simple answer to it. The most common cases are the following:

- You want to use the distro for something specific and the general Linux distros lack a necessary feature. It could be as simple as a heavy customization and the ability to add specific software repositories. Educational and children-oriented distros are good examples.
- You want to focus on only one task and eliminate all features and software that are not needed. There are several reasons for this, such as the following:
    - Hardware constraints: It could be for space on disk or in memory, or architecture, like distros oriented for networks devices like routers or other embedded devices. You could need to reduce the software in the distro to the minimum required; often the kernel is also customized. A distro suited for a mobile device is another example, like Android.
    - Security: If you want to run a Linux distro in a firewall or similar device, you want the minimal software overhead possible for non-vulnerability and performance reasons. And you only want to use software and libraries that you can trust. Alpine is a good example, as is Tails.
    - Ease-of-use: A distro to manage a home theater PC does not need a word processor or a spreadsheet, but an easy-to-use interface oriented for use with a remote is a desirable feature. Likewise, the Chrome OS distro for notebooks is focused on apps and storage in the cloud.

There are a lot of tasks and distros in this category; so many that I devote all of Chapter 17 to this topic. In advance, I will enumerate here some of the tasks and distributions for the sake of completeness.

- Multimedia/Arts: Ubuntu Studio, ArtistX
- Media PC/Home Theater: Kodi (formerly known as XBMC), Mythbuntu
- Security/Forensics/Anonymity: Kali, Tails, DEFT
- Storage: FreeNAS
- Mobile: Android, Tizen, Ubuntu Phone
- Cloud Apps: Chrome
- Routers: OpenWrt
- Gaming: SteamOS
- Cloud/Data Center: CoreOS

I think you get the idea. You can jump to Chapter 17 if you are looking for a distro of this type but a better idea is to keep reading and see what other factors you should consider when choosing a distribution that fits you.

Environment

Another question that you should consider when looking for a Linux distro is the environment where you are going to run it. I've broken it into four main environments where you could run a Linux distribution (I tried to keep it simple): desktop, server, cloud/virtualization, and embedded.

Desktop

The desktop environment is what most people think of when they think about a Linux distro; ironically, it is precisely this one on which Linux has the least success. Although there are some estimations of tens of millions of Linux OSes installed on the desktop (and some estimates go as high as over one hundred million), the truth is that there is a huge gap between Linux and the leaders in this field, Windows and Apple (in that order). Some people believe that the Linux desktop user count is very similar to Apple worldwide. The majority of Linux distros are oriented to this environment (the big ones have specialized versions for each environment), and ironically many people think that this is the reason behind the lack of success of Linux in this area. Anyway, Linux started as a desktop OS and it remains the first Linux experience for most users.

Here we could talk about SoHo users (Small Office, Home) and about corporate and workstations users. Usually a Linux distro oriented to this environment is a general purpose one that is focus on ease of use and offers a familiar graphical interface. These distros also contain administrative tools and services more proper for a server purpose, but they are not suited to this task like servers are, and when people use them in environments other that the desktop it usually requires a lot of customization and knowledge to make them work properly. Debian is a classic example of this.

In this category are almost all of the most popular distros: Ubuntu, Fedora, Mint, Debian, OpenSUSE, etc.

Server

When you need to serve a database or a web site, you use a Linux distro oriented to the server environment. These distros are especially suited for this task, incorporating tools, services, security, customizations, and specialized kernels. They usually lack a graphical environment. Security, performance, and stability are the main features in this category and usually these distros are a bit outdated compared to the desktop distros. These Linux distros are meant to be running on a 365/24 basis and they usually are not rebooted for years at a time. Considering the critical aspects of this category, to rely on a Linux distro not especially designed for this task may work for a short period (usually with great effort and many headaches) but it could be a nightmare in the long run. I would rather use a server distro on my desktop (which is also an unpleasant experience) that use a desktop distro in a server. (Debian is a notable exception).

In this category are popular and well-recognized distros like Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS, and Oracle (the last two are forks of Red Hat) as well as the corresponding server versions of the most popular desktop distros like Ubuntu Server or Debian.

Cloud/Virtualization

This is a subcategory or a super-specialization of the server category, but its importance and how is shaping the future of Linux is enough to claim its own place. You will find here distros that are suited to work in the cloud where they need to be reliable, secure, scalable, easy to automate and deploy, etc.

Most of the big and popular distros are used in this environment, like Red Hat, CentOS, Ubuntu, and Debian. Some offer specialized versions to fit new needs, like Ubuntu Cloud and Amazon AMI (one fork of Red Hat suited to work with AWS), and some even offer their own cloud services like Red Hat OpenStack or OpenShift. There are also extremely light and minimal Linux distros created especially for this environment like CoreOS, which makes use of new technologies like containers (Docker, rocket and Kubernetes). Other new initiatives like that are at last beginning to appear point, together with systemd, to the future of Linux and servers in general.

And although virtualization is supported by default nowadays in the kernel, there are some distributions that focus only on it, like Oracle VM Server or Proxmox VE (Debian based), which work like bare-metal hypervisors.

Embedded/Mobile

The distros in the embedded/mobile category are very specialized, aimed to fit in mobile devices, consumer electronics, network appliances (routers, firewalls), machine controls, industrial automation, road vehicles, medical instruments, etc. The best known ones are employed on mobile devices, such as Android, Ubuntu mobile, and Tizen. Others like OpenWrt and Alpine are focused on network devices. And of course, there is a Debian distro for embedded devices.

Support

For some users, like professionals, the level of support can be a threshold point to filter which Linux distros are suitable for them and which are not. Good documentation or support by the community of users (wikis, irc, forums, mailing lists, etc.) can be enough for experienced or domestic users, but for the corporate world you may desire commercial support to keep the systems running without trouble. So, as you maybe already suppose, there are two categories in this group: community and commercial support.

Community

Traditionally, because of the origins of Linux itself, the help documentation needed to install and troubleshoot the OS is created and maintained by the community of users and developers. If the distro has a big community, the quantity and quality of the documentation and support is better. When the support is better, it attracts new users to the community. A traditional example of this is Debian. However, even some distros that are oriented to advanced users and are hard to install and administrate to newcomers have excellent documentation and a very supportive community, like Arch Linux or Gentoo.

What you can expect from the community in the majority of distros is some documentation (in a webpage, a wiki, or a CD) and several ways in which the users can communicate with each other in order to answer questions. Almost every Linux distro has a mailing list, and most of them also have irc channels and forums; the bigger ones also have some system for users to submit bugs to the developers. There are often separate resources of the same type to drive developers and normal users into sub-communities that sometimes overlap on some topics.

Commercial

When corporations and institutions began to replace their UNIX installations with Linux, some people saw the opportunity to do more than sell and distribute the software; they could also offer commercial support from experts to assist in the process of migration and maintenance of Linux. Thus the first commercial distributions of Linux supported by a company were born. Today two of them still represent a major number of corporate Linux installations: Red Hat and SUSE.

There are different approaches to offering commercial support. Companies like Red Hat have an entirely commercial distribution. Red Hat also has a separate distro named Fedora that is supported by the community; some Red Hat programmers contribute to its development, and it is where they introduce the most recent technologies and test a lot of changes that are later incorporated into the commercial distro. The company Canonical has a main distribution, Ubuntu, that is essentially the same for all users and is partially supported and developed by the community, but it also offers commercial support to SoHo and corporate users and some tools especially fit for the corporate world, like an Ubuntu Server version.

This commercial support can be offered in different channels, such as the community one, from personal support offered by experts directly on the site (by direct employers or consulting firms that are official partners) to telephone hotlines or web sites offering help.

Usually this commercial support is linked to a maintenance contract paid annually. And some companies, like Red Hat and SUSE, also offer certification programs and exams to provide external professionals the knowledge and competence to maintain these distro installations. I'll explain this in detail later in this chapter.

User Friendly

I'm going to split the user friendly section into two groups: beginners and advanced users. Yes, this is a little unfair: using an "easy" Linux distro does not imply that the user does not have profound and advanced knowledge about Linux. Take Apple, for example; their OS X is famous for its ease of use even for beginners but it is also the OS chosen by a lot of IT professionals, even Linux Kernel developers. Some people don't want their OS to get in the middle of their work; they want an OS that simply works, so they choose a distribution that is very easy to install and maintain, and one that requires very little time to keep current. Others are just the opposite; they want to tweak every minor detail of their OS and feel that they are in total control of their machines. Thus, choosing a distro based on its ease of use is not always a question of knowledge or capability; most of the time it is a question of personal preference.

So, when talking about the friendliness of a Linux distro, there are two questions to consider: your level of experience with Linux and how much time you want to spend maintaining, customizing, and learning about your OS.

Beginners

Is this your first rodeo with Linux, or do you have a little experience with it? Do you want to spend very little time tweaking your distro? Do you want to avoid as much as possible the use of the terminal? Do you have little interest in learning about Linux? If you simply want to install the OS and then spend very little time keeping it updated, you should consider a "beginner" Linux distro. Keep in mind that ease of use does not mean inferior quality, stability, or capabilities. In fact, the bigger distros try to make the experience with the OS as pleasant possible, hiding all of the nuts and bolts from the user, as a way to attract the biggest number of users possible.

Saying what is easy to use is not always as clear as you might think. There is a large amount of subjectivity here. However, there is some common ground that allows us to establish a certain agreement on which Linux distros are friendlier than others, such as how easy it is to install the distro and how easy it is to update and install new programs and hardware.

Some examples of this kind of distro are Ubuntu (which, in fact, had a great impact in this aspect), Mint, and elementary OS.

Advanced Users

As in the previous category, there is no direct relation between experience and this kind of distro; some users who are new to Linux begin with an "advanced" distro because they want to learn as much as they can about the OS and want to have an OS suited to their needs. I strongly discourage this approach because desire and willingness do not always come together, and this approach usually ends in frustration. If you are coming from Windows or OS X and you decide to start your Linux experience with a distro focused on advanced users, you must be ready to learn a lot, and I mean a huge amount, of concepts that cover a big span, from the internals of your machine to the internals of your OS. And Linux is not like Windows in that you will not have a GUI to tweak any aspect of your OS (in fact, this is true for some things in Windows); you must use the terminal to write some complex, arcane, and often very long commands as opposed to editing simple text files. It is not a walk in the park. But it could be a very funny and pleasant experience if you are willing to do it. So, it is up to you, but if you have little experience with Linux I strongly recommend that you start first with a "beginners" distro to get a glimpse of how the OS works and to learn the general concepts.

On the other hand, all of the distros (well, most of the desktop ones) allow you to tweak them at a very deep level through the terminal. This is counterproductive in many "beginner" distros; they work very well as installed. If you customize them too much at critical points, you may end up regretting it. This is one of the main reasons why distros for advanced users exist. If you want to deeply customize your Linux OS, you must use a distro that allows you to do so without breaking things in the long term. Anyway, distros like Debian have a friendly way to customize the distro as you wish (and sometimes break it too).

Some of the distributions in this category are Arch Linux, Gentoo, NixOS, and Slackware.

Stability

Stability is another crucial factor for professional users who needs the MTBF (mean time between failures) of their OS to be minimal. For professional users, time is money. Failures in their Linux installation could cause other serious problems like data loss/corruption or security breaches. When I talk about stability here, I'm talking about how often a distro updates the packages and kernel.

Traditionally this factor is divided into two categories: standard release distros and rolling release distros, but in some distros this division is more like a blurred line where no clear model exists and both can be used at the same time.

Standard Release

The standard release model is the desirable model when you need a Linux distro for a professional environment. This release cycle is based in versions (with a fixed period or not) and the core base of the distro is kept stable until the next release, which only contains bug fixes or security patches. When a new version is launched, the kernel and packages are not fresh ones; they are stable versions, well tested and arguably secure. This is the model of Red Hat, for example. Versions are normally linked to the development or upgrading of important features and do not have a specific interval of time between releases; they are released when they are ready. The Linux kernel itself follows a similar model with its version releases.

Others distros like Ubuntu have a time-based release, which is currently two times a year (a six-month period). Every fourth release (a two-year period) is a special one named LTS (long-term support) that aims to be a stable release maintained for a long time (five years versus the nine months in normal releases). The LTS is the one oriented to professional environments.

Debian goes a step further. It has three different models: Stable, Testing, and Unstable. The Stable model follows the standard mode, the Unstable model is a rolling release distro, and the Testing model covers some ground in the middle. Debian Stable is a very common Linux distro used in many web servers and other reliable services.

With this model, once a new version is released you usually have two options:

- Upgrade your distro to the new version. This is the most comfortable option, when all goes well. However, even when some distros take special care to make this process go as smoothly as possible, it does not always work perfectly. Sometimes this is due to errors in the process, but most of the time it is due to heavy customizations from users. Remember to always make a backup of your data before any distro upgrade (you should make backups regularly).
- Make a backup of your data, make a fresh install, and restore your backup. This requires more work and time compared to the upgrade option (if all goes well, obviously), but it does minimize the problems that you could find. Some users end up doing this due to experience gained after various broken distro upgrades.

Rolling Release

Rolling release distros try to keep pace with the development of the Linux kernel and the different software packages so they always have latest version available. In order to do so, some distros also tend to minimize the customization of those packages and kernel, so they are almost vanilla packages. They are geared to advanced users and those who are eager to have bleeding-edge technology. Once you install one of these distros, if all goes well, you usually only have to do periodic updates.

These distros are considered not as stable and harder to maintain, but the truth is not as severe as it sounds. You can work on a daily basis with one of these distros without problems; some people even use them in servers. Many of these distros offer solutions for keeping a certain stability, like different repositories separating the latest software versions from the ones that are a little outdated (frequently a matter of days or weeks) but are more reliable.

There is a controversial opinion that a rolling-release distro is always more reliable in terms of security terms because the latest version has more holes fixed and is less prone to having zero-day vulnerabilities because there is no time to discover them. I think that you should take this theory with a grain of salt: some vulnerabilities exist in version after version because nobody noticed them.

There are many distros that follow this model; the most popular ones are Debian Unstable, Arch Linux, and Gentoo.

■ **Tip** I am trying to be as objective as possible in this book. If you are curious about my software choices, I use a rolling release distro on a daily basis on my laptops and workstations but I always use a standard release distro on my servers and critical machines.

Hardware Support

The question of hardware support is usually as simple as how many resources are available to the team responsible for maintaining and developing the Linux distribution. Much hardware is supported by the kernel directly, but the distro controls how good the automatically detection and enabling of this hardware is. Also, there are binary drivers that some distributions do not add to their distros for ethical or licensing reasons, which makes it very hard to support certain hardware. Printers, graphic cards, network cards, and wireless devices are usually problematic devices, and even some laptops are poorly supported by Linux.

Happily there has been a huge improvement in this field over the last few years, and often there is no problem at all with common hardware. But specialized devices usually require the company behind them to develop a driver for Linux. Although this is common today, there is another problem: usually the companies

only develop drivers in .rpm and .deb packages, forcing the user to make conversions or to compile from the source in other distros. Sometimes this does not work very well because they are customized for a certain distro, normally Red Hat/Fedora and Debian/Ubuntu.

Thus you must know before you buy any hardware if it is completely compatible with Linux; this is especially important with laptops. There are some websites dedicated to covering this issue. The best way to ensure that you will have minimum problems with hardware in Linux is to do your research and choose one of the big distros (Ubuntu, Fedora, Debian, Mageia, OpenSUSE) that normally have a lot of resources or a company behind them. You may think that the derivate distros of these big ones will offer the same support, but this is not always the case; and if it is supported, it is not always as simple as adding the same hardware in their mother distro.

Aesthetics

Aesthetics is a very subjective topic, but here is one thing that you can take from a completely objective point of view: whether the distro takes care of its design or not. Usually, as in the previous factor, this is directly related to the level of resources available to the team behind it. Having some community developers try to make a distro look pleasant is not the same as having a team of designers focus on it. There is one notable exception: elementary OS, a community-driven distro that aims to offer the best design experience possible with only a small team behind but tons of good taste (it is an Ubuntu-based distro; they focus mainly on usability and design). It has the same goals as OS X: ease of use and being a pleasure to the senses (in fact, it's been criticized for taking heavy inspiration from OS X). There is a new project called Ozon OS that looks to follow the steps of elementary OS but with its own sense of design; it is still in the early stages but it seems promising. Other distros that have a very good design (I do not have to like them, but I recognize the effort) are Ubuntu, OpenSUSE, Mint, Mageia, and Fedora.

But to be honest, once you have chosen one of the main desktop environments, KDE or Gnome, you do not have too much room for improvement. Even if you customize it and the main programs that are installed by default, there is so much software in the repositories with its own design and aesthetic, and this software does not use the API of those common desktops. So it is hard to keep a homogenous aesthetic within the OS. This problem is common to Windows too, and also in very minor way to OS X, but in Linux it is more obvious. It does not matter if elementary OS is beautiful and homogenous in all of its (few) native apps if you install an awful software that comes with its own design. If design matters very much to you, keep in mind that a Linux distro like Mint that has a really good aesthetic but with minor customizations would be always more homogenous in its design if you are willing to install more software that the few developed apps that elementary OS offers.

It would be stupid for me, even when I love and support Linux, to not recognize that a nice design and a homogenous look is still far from being ideal for the most of the Linux distros out there.

Desktop Environment

The desktop environment is also a question of taste. Some people strongly prefer a desktop environment and even dislike others. The most commonly used and known are Gnome and KDE, but there are some alternatives like Unity, Mate, Cinnamon, Xfce, LXDE, and Enlightenment.

Usually you can have several different desktop managers installed in your distro, even at the same time (choosing the one you want to work in at login time), but is normal for each distro to have a predefined one by default. Some people choose OpenSUSE over Fedora, for example, because they prefer to work with KDE instead of Gnome. But some of the big distros let you choose what DE you want to use when installing them.

To measure the impact that a desktop manager can have in a distribution, consider Linux Mint. When the Gnome project released the version 3 of its desktop environment, there were huge changes; as a result, many detractors wanted to keep working on version 2. Mint decided to develop two desktop environments to please those users: Mate is based on Gnome 2 and Cinnamon is a fork of Gnome 3. The distro gained a

lot of users because of this decision. Later, Ubuntu developed its own desktop environment, Unity, which also created a lot of controversy and a great number of people who did not liked it, so they migrated mainly to Mint. Those two great migrations of users and the excellent work made in both environments means that today Linux Mint is the most popular distro on distrowatch. So, clearly the DE matters a lot to a huge number of users.

I will cover this in more detail in the chapters dedicated to the distros, showing the DE available (and predefined options) for each distro.

■ **Tip** I do not use any desktop environment. I use a windows manager instead, a tiling window manager to be more precise (Awesome WM in particular). This is a very trendy option for a lot of advanced users and there are plenty of possibilities, the most well-known being OpenBox, FluxBox, xmonad, i3, Awesome WM, and dwm.

Init System

Until very recently I did not consider the init system as a factor in a decision because almost all of the distros were using the same init system, init (SysV). But in 2006, the Ubuntu distribution (and all of its derivatives) implemented a new init system, Upstart, that was also adopted as the default by Red Hat and Fedora. And five years ago, a new init system was born to become a revolution in the Linux ecosystem, and a very controversial one. This new init system, systemd, created by Lennart Poettering at Red Hat, not only was a revolutionary init system, it soon become something more that an init system, in fact much more than that. And that made a lot of people angry, creating many polarized opinions on the matter.

The systemd init system broke some of the traditional UNIX principles, and that is not acceptable to some people, both distro developers and system administrators. The current reality is that almost all of the big Linux distributions adopted systemd, and others are totally opposed to taking that step. Two cases became famous and controversial: Ubuntu discarded its own init system in favor of systemd, and Debian adopted systemd after a very contentious debate among the members of the Debian Technical Committee. This created an outburst in the Debian community and resulted in the resigning of some members. The situation is still hot, and recently a new distribution forked from Debian called Devuan was created to allow to those who hate systemd to still continue to use Debian (only as Devuan). In 2014, some users started a boycott campaign against systemd but it has not had a huge impact.

Thus, as mentioned, some distros are reluctant to abandon sysvinit or adopt systemd, and as a result you can now choose between different alternatives. Since this topic is so controversial, the type of init system has clearly become a factor for many people when choosing a Linux distro.

sysvinit (Traditional Init)

Some distros are still using the traditional init system, also known as SystemV init, and others offer it as an option. Devuan is one of them, as are PCLinuxOS and LFS. There are a few more, but basically none are very popular distributions. We are at the end of an era for sysvinit.

systemd

Almost all of the popular distros and derivatives are supporting the systemd init system as the default option these days. It is included by default in Red Hat, Fedora, Ubuntu, Debian, Arch Linux, and OpenSUSE.

BSD-style

The BSD-style init system was used by Arch Linux for many years, but it was abandoned in favor of systemd. CRUX is still using it, however, as is Slackware.

OpenRC

Created by a former Gentoo developer, Roy Marples, OpenRC is still the init system used by that Linux distribution. Gentoo also allows systemd as an option, the same way that you can still continue to use sysvinit if you want. That's the very nature of the distro itself: do whatever you want. Other than the ones based on Gentoo and Alpine, I do not know of any other Linux distros using the OpenRC init system.

Others

There are a few other init systems, but they are merely testimonial, such as GNU dmd for Guix, Mudur in Pardus, BootScripts in GloboLinux, or busybox-init suited to embedded systems.

As you can see, there are not many options if you are a systemd detractor, but it's Linux, so you always have a few options.

Package Management System

In Linux, you install/update software using a package manager, usually from official repositories on the Internet or directly from a package. These days it's rare to have to compile a package by hand (but it happens). A package manager is a fantastic, centralized, and secure way to manage the software of your operating system. But, as usual in Linux, there are many different package managers.

I'm not going to get into my opinion of whether one is better than the others; all of them have advantages and disadvantages. The truth is that it is very difficult, if not impossible, to use one as a complete replacement for the one that comes by default with the distro. So we can divide the Linux distros by their package management systems. Also, distros that share the same package management system often have different tools to manage them; Mageia uses urpmi to manage its rpm packages, while Fedora uses DNF (previously YUM).

RPM

RPM is the acronym for Red Hat Package Manager and is obviously the one used by that distribution and its derivatives. It was also adopted by SUSE and Mandriva and thus their derivatives. It's the de facto official packaging system, and it's the one most frequently supported by corporate developers. These packages have the .rpm suffix.

Some of the Linux distros that support this system are Red Hat, CentOS, Oracle Linux, Fedora, SUSE, OpenSUSE, Mageia, and PCLinuxOS.

dpkg

The package management system created by Debian is called dpkg, and it is the second most used after rpm. The suffix of these packages is .deb. When corporate developers release a version of their software for Linux, they normally offered .rpm and .deb packages, and sometimes also the source.

Some of the many distros that use this system are Debian, Ubuntu, Mint, and all of the Debian/Ubuntu derivatives.

pacman/AUR, Portage/emerge, and tgz

pacman/AUR, Portage/emerge, and tgz are the package management systems of Arch Linux, Gentoo, and Slackware, respectively. The first two are sophisticated and advanced systems because they support advanced and rolling release distros. The last is used almost exclusively by Slackware.

Pacman/AUR is used in Arch Linux, Manjaro, Parabola, Antergos, Chackra, and ArchBang. Portage is used in Gentoo, Sabayon, Chrome Os, and Funtoo.

Nix/Guix

Nix/Guix is a new type of package management system. It is based in a purely functional model, which allows a series of unusual features like multiple versions of the same package, atomic upgrades and rollback, garbage collection, etc. Nix is the original concept; Guix has a similar approach and comes from the GNU Project.

Currently, as it is the state of the art in this field, there are only two distros that support these systems: NixOS with Nix and GuixSD with Guix.

Architecture

Since the decline of the RISC processors (except ARM), there are not very many different architectures in modern machines. Intel processors are the clearly dominant actor in the market, followed by AMD. The ARM processors are getting more and more popular, thanks to mobile systems. But there are many other architectures out there, such as folks who still carry around old Apple PowerBooks with a PowerPC processor, servers with Intel Itanium cores, workstations such as SUN UltraSPARC, IBM mainframes, etc.

The Linux kernel by default supports a lot of different hardware architectures, but not all of the distros support them. In fact, the majority of the distros only support the major ones: Intel and perhaps ARM.

Intel

In the Linux kernel, the denomination i686 (a.k.a. 586) is for the 32-bit processors after the 386 (deprecated in the kernel), and x86-64 (a.k.a. amd64 or intel64) is for the 64-bit processors from Intel and AMD. Almost all of the distros currently support the x86-64 architecture and i686 as well, but some of them are abandoning the i686 architecture, like Sabayon Linux.

Ubuntu, Debian, Fedora, openSUSE, Mint, Mageia, … as said, practically all of them support both architectures.

ARM

The ARM processor is the king of consumer electronics and mobile devices; its omnipresence is astounding. However, ARM is not as well supported as the Intel/AMD processors, and currently only a few distributions have ports for it. Some of these distros are Ubuntu, Debian, Fedora, Arch Linux, Gentoo, Slackware, Kali, and of course Android and Chrome OS. Perhaps the fact that ARM is becoming as an interesting actor in the low-consumption server world will change the adoption by the big actors in this field (Red Hat and SUSE are the first ones making some movement in that direction).

Others

Other architectures are still less supported, as expected. The commercial ones have good support for mainframes and PowerPC. If you are looking for other architectures, even exotic ones, your decision is a clear one: Debian. Debian supports a lot of different architectures, about ten officially and more as unofficial ports. You can even find images for older architectures and install them as if were the 1980s.

Security/Anonymity

If you are concerned about security (and you should be), you must know that some Linux distros are more conscious about this topic than others. Usually the big ones that have more resources, especially the commercial ones, are the most dedicated to this matter. Distros like Red Hat, CentOS, SUSE, and Fedora are the most secure by default. Linux is a reasonably secure system by default, but usually the Linux distros do not come with a default hardened security; you can make it this way if you have the knowledge, but still you must rely on the security of the distro for things like the software repositories. For example, you can make Arch Linux a very secure system, but until recently the software packages not were signed, and the AUR packages can be a security concern if you don't know what you are doing.

There are a few distros that are focused entirely on security from different perspectives. Some of these are

- Tails (The Amnesiac Incognito Live System): It is based on Debian and its perspective is to provide security through anonymity, preserving your privacy.
- Pentoo: It is a Live CD Gentoo-based distro hardened by default.
- Kali (formerly BackTrack): It is also based in Debian and is focused on digital forensics and penetration testing.
- Alpine: A lightweight distro that focuses on being secure by default. A tiny and very secure distro, it is primarily designed for x86 routers, firewalls, and similar.
- Qubes: Its approach provides a secure system based on the isolation of its components via virtualization. It is based on Fedora.

Principles/Ethics

In the previous chapter you learned how the GNU Project took an important view on principles, and how the FSF created a new way to license software based on a series of solid principles and ethics. The FSF has a series of guidelines (which you can read at `www.gnu.org/distros/free-system-distribution-guidelines.en.html`) that establish when a Linux distribution is a Free system.

> *This means these distros will include, and propose, exclusively free software. They will reject nonfree applications, nonfree programming platforms, nonfree drivers, nonfree firmware "blobs," nonfree games, and any other nonfree software, as well as nonfree manuals or documentation.*

Free Software Foundation: About Free Linux Distributions

There are a few distros that embrace these guidelines and thus are endorsed by the FSF. If you want to support these guidelines, you may want to choose one of those distros. But I must advise you that in this case (when not?) freedom comes with a price, and you are probably going to be very restricted in the hardware that you can use. The major problem here is with graphic cards, network cards, and printers.

Some of the distros endorsed as Free by the FSF are Trisquel, gNewSense, Dynebolic, and Parabola. You can get the complete list at www.gnu.org/distros/free-distros.en.html.

Live CD

A Linux Live CD is a Linux distro contained on a CD-ROM (or DVD) that can be booted and run as if was installed on the system but without writing anything to the hard disk. There are three main reasons why you would want to use a Linux distro in a Live CD:

- To test a Linux distro without installing it.
- To use it like a rescue tool to recover a system. There are Live CD distros that specialize in this approach.
- For anonymity/security reasons. A Live CD does not leave any footprint in the system and you can be sure that the distro was not altered (e.g. trojans) because of the only-read medium. Distros like Tails are a perfect example of this use-case.

Not all distros have a Live CD version, but Ubuntu, Fedora, Mageia, Debian, OpenSUSE, and Mint do. Laptops and workstations often lack a DVD reader these days, so these Live CDs are also usually available as an image to install on a USB flash drive and boot from it.

Professional Certification

Some commercial distros have a program to certificate professionals with the necessary skills to perform administrative and engineering tasks on their operating systems. There are also certifications offered by third parties that are not linked to any distribution of Linux in particular. A Linux professional certification is often a mandatory requirement to work in companies with a Linux infrastructure, so it could be a good investment for any System Administrator that wants to works with Linux installations.

There are currently five major Linux certifications, three tied to a specific distro and the others not (Ubuntu offered one in the past, but was discontinued in 2010):

- LPIC by the Linux Professional Institute is a very well recognized certification not linked to any distro, but their exams and questions are based both in Red Hat and Debian. They currently offer three levels of certification:
  - LPIC 1: Linux Server Professional - Junior Level. It requires you to pass two exams: 101 and 102.
  - LPIC 2: Linux Server Professional - Advanced Level. It also has two exams (201 and 202) and requires you to have a LPIC 1 Certification.
  - LPIC 3: Linux Server Professional - Senior Level. It consists of one exam from one of the three available specializations:
    - 300: Mixed environment (Linux and Windows)
    - 303: Security
    - 304: Virtualization and High Availability

- Linux Certification by the Linux Foundation. The Linux Foundation recently began to offer two different certifications (these certifications are based on CentOS (Red Hat), Ubuntu, and openSUSE):
  - Linux Foundation Certified System Administrator (LFCS) - Advanced Level
  - Linux Foundation Certified Engineer (LFCE) - Senior Level
- Novell offers certifications to administer systems with SUSE Linux. The current Novell certifications are the following (each one of them requires one exam):
  - SUSE Certified Linux Administrator (CLA) - Junior Level
  - SUSE Certified Linux Professional (CLP) - Advanced Level
  - SUSE Certified Linux Engineer 11 (CLE) - Senior Level
- Red Hat has three levels of certifications to qualify as an expert system administrator of their distro. This is the most recognized Linux certification (and one of the most in all of the IT world qualifications too) and their owners are in much demand.
  - Red Hat Certified Systems Administrator (RHCSA) - Advanced Level. It requires one exam.
  - Red Hat Certified Engineer (RHCE) - Senior Level. It requires one exam.
  - Red Hat Certified Architect (RHCA) - Master Level. It requires five exams from within nine different specializations.
- Oracle, which has a Linux distro that was originally a fork of Red Hat, offers their own certification programs at two levels: OCA and OCP.

As an IT professional, you should pursue any of these certifications and therefore choose any of the distros involved in one of these certifications. As a company IT manager, you may want to find a Linux professional that has one of these qualifications to be sure of her competence. Earning these certifications can be expensive (about $4,000 for the RHCA exams) and can require a high level of experience. The most valuable certifications are the Red Hat and LFC ones because they are hands-on exams based on real experience, not just theory.

Linux Standard Base

By now you should realize that not all Linux distros are equal. Obviously they have a lot in common, but at the same time they have many particularities that make them different from each other. In order to make this "mess" something bearable and manageable to software developers (and ultimately the users), the LSB[1] was created.

The Linux Standard Base is a project supported by several Linux distros that aims to offer some common ground between the different distributions. It provides standards for things like the file system hierarchy, libraries, printing system, run levels, and so on.

[1] www.linuxfoundation.org/collaborate/workgroups/lsb

> *The goal of the LSB is to develop and promote a set of open standards that will increase compatibility among Linux distributions and enable software applications to run on any compliant system even in binary form. In addition, the LSB will help coordinate efforts to recruit software vendors to port and write products for Linux Operating Systems.*

The project originated in 2001 and the current version of its standards is 5.0. But its impact is very small; only a few Linux distros follow it, like the commercial ones: Red Hat, SUSE and Ubuntu. So it is currently only a beautiful idea embraced by a small percentage of all Linux distributions. Also, it has been criticized by some distros like Debian over some controversial decisions. As always, Linux gives you freedom at the cost of its big diversity.

Summary

In this chapter, I showed you several factors that you should consider before choose any Linux distro. Use the factors that most concern you to compare distros. Also, a cruel reality was revealed to you in this chapter: not only is there a huge number of distros, but they are very different. And although there are some people working to "unify" Linux, there is still a long, long way to go. Maybe there is no need to unify Linux; maybe things like containers, Nix, and systemd are going to make that unnecessary in the future.

In the next chapter, I introduce you to the genealogy of the current "families" of distros.

CHAPTER 3

The Linux Distro Family Tree

The first Linux distributions were created from scratch, having only the Linux kernel and a bunch of dispersed tools (like the GNU Project ones) with which to work. It was necessary to create a series of new tools and scripts to compose what we consider a Linux distro. Over time they would become more sophisticated, advanced, and useful, but the very first distros were created in this way, from zero.

But once there were some distros that were good enough, and because some of them were Free Software, it began to make sense to create new distros based on these early ones, rather than going through all of the work to recreate the required nuts-and-bolts software. Thus, soon there were new distros based on prior ones, but with various tweaks to achieve the goal(s) of the creators. As with the kernel, the user communities and the free licenses allowed for the fast adoption of this model.

This model is still the most used, and it has given birth to a series of "families" of Linux distros. In fact, only a few Linux distros are created and maintained based on the original approach from the kernel. The majority of distros today are derived from ones that existed before, sometimes from the "original" ones and sometimes from ones that were derivatives themselves.

The reasons to create derivatives are obvious:

- You already like most of a specific Linux distro, but you want to change some things. The best way to do this is to create a fork of the distro and make the changes you need. If you maintain synced all of the things that you left intact, it helps to develop your distro through the years.
- You have a different or personal view of what a Linux distro should be, and you want to create a personal distro, but you (or your team) do not have enough resources to perform all of the necessary work. Taking a previous distribution and selecting what fits your project is an excellent starting point.
- You want to create a task-oriented distribution but you don't want to deal with the hard work that creating an entire distro from scratch entails, so you pick a distro that you like because most of the job is already done.

Sometimes forks originated within the distro's community itself because of a difference of opinion or irreconcilable disagreements between members. There are some famous examples of this in the history of Linux distros.

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_3

Linux Distro Genealogy

In this chapter, I want to give you a brief idea of how the Linux distro genealogy evolved from the time of the birth of the Linux kernel itself. You can see the popularity of a distro by how many forks it has or how old it is. Also, the genealogy explains why some distros do the things the way they do. I don't pretend to include all of the current distros; that would be an astounding task and, except from an academic point of view, not a great help; it might even be boring and confusing.

I have divided this genealogy into three periods to help you assimilate the information, and I created three graphical timelines to show a quick view of this evolution. In those graphics, the distros are represented as originals or as derivatives. Some of the distros that were created as a fork of one may switch over time to a completely original approach or become based on a distro different from the original one. The same happens with names; I use the current name (or the last one, if it has been discontinued), but sometimes the name changed more than once during its lifetime. The date you see is the date of the first release of each distribution; I do not list the end date of the distribution for the sake of clarity.

Also, I don't get into a detail discussion on every distro because this would be pointless given the quantity of distros. Some distros did not have a significant impact on history or were not different enough to rate discussion. As for the historical distros, there is so little information available. Profound research into this topic is beyond the goal of this book and not my intention anyway.

By the way, there is very little information (and it is often contradictory) about the history of Linux and the distros in particular. Somebody should take up the torch, do the research, and publish it[1].

Let's begin with the early Linux distro timeline shown in Figure 3-1.

[1]I only know about one book onto this topic, *Rebel Code: Linux and the Open Source Revolution* by Glyn Moody for Basic Books, but it only cover in detail the first years of Linux.

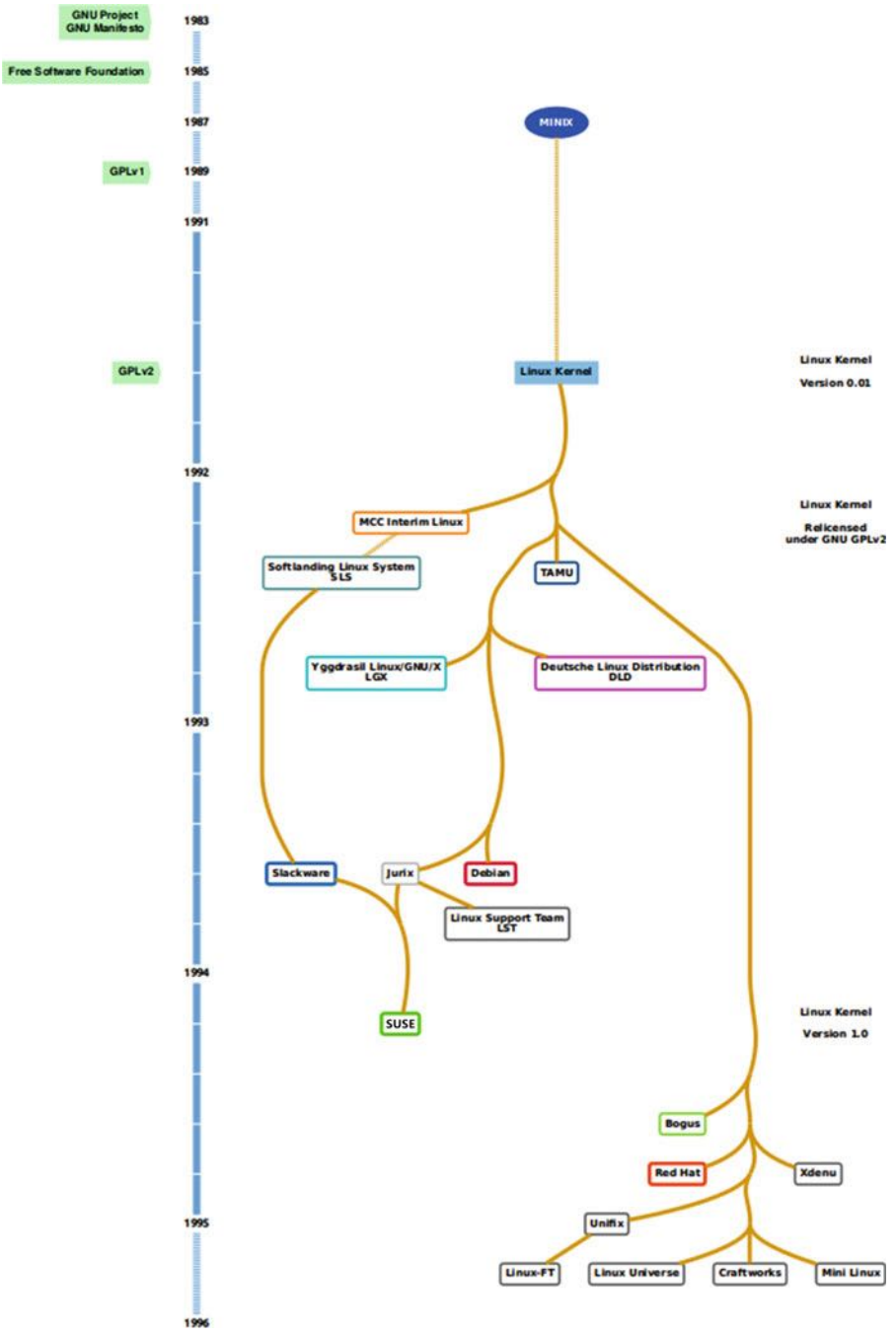


Figure 3-1. *The Linux distro timeline: 1991-1995*

1991-1995: The Big Bang

The first five years of Linux history represents the Big Bang in the Universe of Linux Distros, from none to the first ones appearing. This is also the crucial time when four of the most important distros in history were born: Debian, Slackware, SUSE, and Red Hat. And although there were some derivatives even then, this is the period when distros based on an original concept were the majority.

The first distros appeared in 1992, the first year after the beginning of the Linux kernel. All of them were pioneers, and all of them are extinct today. The first distros were released by one or two developers. They were the foundation of the main four (Debian, Slackware, SUSE, and Red Hat), either as derivatives or as a source of inspiration. Let's learn a little about these pioneers.

- **MCC Interim Linux**: This was not exactly the first distro to try to achieve something similar to the goals of a Linux distribution, but it's the first one that we can name properly as a Linux distribution. It was released in February of 1992 at the University of Manchester (England), more specifically at the Manchester Computer Centre, hence the name. A set of end-user and programming tools plus the kernel could be installed through a primitive menu-driven installer. Thanks to the work of its developer, Owen Le Blanc, non-experts in Linux could install the OS in their systems for the first time. It is currently discontinued.
- **Softlanding Linux System (SLS)**: This name is very representative of the intentions of the first distros. It was released by Peter McDonald in Canada in May of 1992. It was the most popular one for a brief time but was criticized for its numerous bugs. It was the first distro to use a graphical environment, the X Window System (still used today and coming from the UNIX world) and was inspired by the MCC distro. It ceased to exist.
- **TAMU**: This distro was released at the same time as SLS by the Texas A&M University Linux Users Group. Some sources claim that this distro was the first one to include the X Windows System. It was also, like SLS, a buggy one. It is not active at the moment, but there was an attempt to resuscitate it as late as 2010. It is discontinued.
- **Yggdrasil Linux/GNU/X (LGX)**: The first distro that had a company behind it, Yggdrasil Computer Incorporated, making it the first commercial distro. It was named after the World Tree of the Norse mythology. It was developed by Adam J. Ritcher in California and released in 1995. It's notable for three big achievements: it's the first one to be distributed as a Live CD, the first one to auto-configure and detect hardware, and the first one that was compatible with the UNIX Filesytem Hierarchy Standard. It's also another one that did not survive those days.
- **Deustche Linux Distribution (DLD)**: An early distro from Stuttgart, Germany, released in 1992.

■ **Tip** If you want to explore more and see how those early distributions really looked, you can still download old versions of some of them at `www.oldlinux.org/Linux.old/distributions/`.

Now let's look at the big four. Almost at the same time (the summer of 1993) but in two different parts of United States, two developers, both annoyed by the lack of stability of the SLS distro, had the same idea: to begin their own distribution.

- **Slackware**: Created by Patrick Volkerding in 1993 and originally a fork of the SLS distribution, it is the oldest distro still maintained actively. Perhaps with the intention of avoiding the errors of SLS, this distro focused on stability and simplicity. It was created with no further intentions and even its name was based on a joke, but is the most mature, one of the most popular, and the basis for another big distro. It was the first one that was really popular. It is probably the most representative of how those early distros were at that time, because of its conservative nature.
- **Debian**: Ian Murdock created this distro in 1993 and named it after his then-girlfriend, Deborah, and himself. It was originally inspired by SLS but it was not a fork. Its big contribution was the Debian Manifesto, which was included very early in the first release and was aimed at maintaining and developing the distro in a free and open way, like Linux, creating the first Linux distro community. This idea is still at the core of the distro and it continues to have strong values about freedom and sharing through the Debian Social Contract. Over time it would become one of the most forked distro in history.

Not too much later the other two big ones were born:

- **SUSE**: A German distribution, still active, that was first released in September of 1992. The first big, commercial one, SUSE was one of the most used distros in Europe for a long time. It was based originally on Slackware but soon it began its own distro based on another one, Jurix (now discontinued), also a German distro.
- **Red Hat**: Red Hat started by selling Linux and UNIX software accessories, and then it released its own distro in 1994. It was the big commercial one in United States (as SUSE was in Europe) and is still the most important distro in the USA, and also worldwide. It was the first to adopt the RPM package management system, based on a previous development in the Bogus distro.

The middle era in Linux distribution history is explained in the next section. I have split it into two groups visually, Figures 3-2 and 3-3.

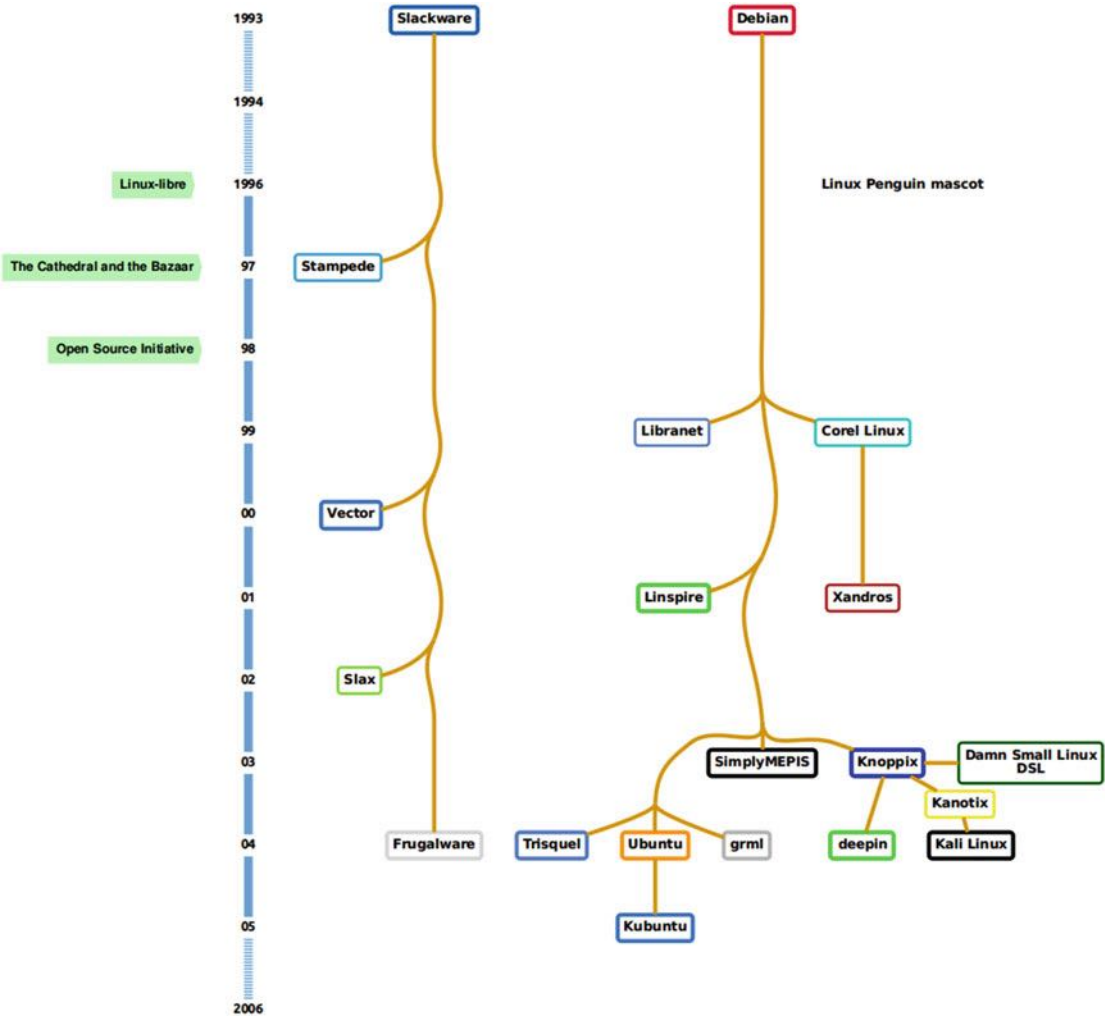


Figure 3-2. *The Linux distro timeline: 1996-2005, Part I*

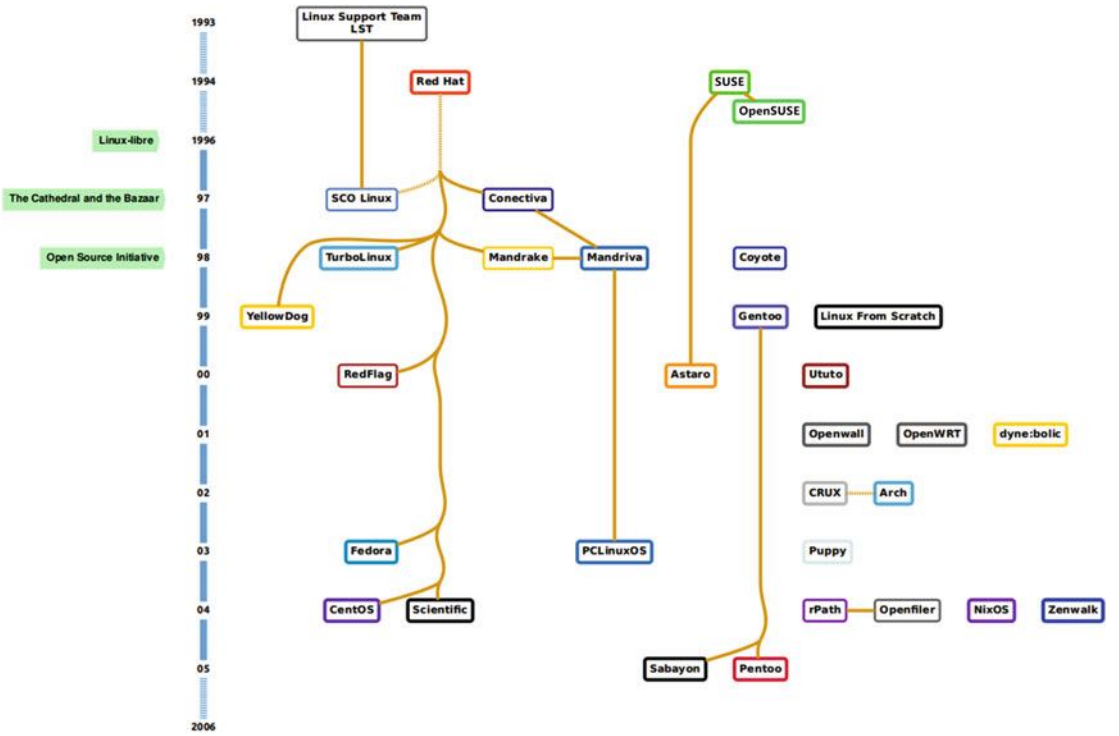


Figure 3-3. The Linux distros timeline: 1996-2005, Part II

1996-2005: The Expansion of the Linux Universe

The period of 1996-2005 is characterized by the crowning of the big four as the most popular distros and the expansion of their family trees. This is the era in which the fork mode was popularized, and the big four were the most forked ones, especially Red Hat and Debian. It's also notable as the time when three distros, which would become the source of many forks, were born: Gentoo, Arch, and Ubuntu.

Red Hat

One of the most popular distros in this era is also one of the most forked. Some of the most relevant distros based on Red Hat (RH) are

- **SCO**: Originally named Caldera, it was the first distro fully oriented to the corporate world. It was based on the German LST distro but also much inspired by Red Hat. It's not properly a fork of RH, but it adopted much of RH's solutions. It showed other distros how to be an alternative to the corporate desktop, making it very popular for years. It's currently defunct.
- **Mandriva**: A French distro released in 1998, Mandriva was very popular in Europe and was considered very friendly and easy to install and manage. It was greatly used as an alternative to the Windows desktop. It is currently discontinued.

- **Fedora**: The community-developed alternative to Red Hat, Fedora is supported by the company. It was released in 2003. Currently it's a sort of community lab for developing/testing some of the new technologies of the mother distro, but originally it was created as a community project to provide an alternative to the defunct Red Hat Linux, the full open source distro of RH.
- **CentOS**: Similar to Fedora, CentOS is a fork of the commercial distro of RH, Red Hat Enterprise Linux. In fact, it is almost like the last one, without the proprietary parts. Originally it was created as a clone of RHEL in 2004, and it's probably the distro most installed on corporate servers around the world.

SUSE

SUSE did not have many forks in these days but it had a very significant one, OpenSUSE, which is the equivalent of Fedora in Red Hat. It was really the successor of the SUSE Linux Personal distro. SUSE was the most professional approach that a Linux distro had in those days, but that implied that much of the code was private and only available as free after a few months, that affected seriously on his adoption and therefore a small number of forks. The OpenSuse distro was a change of that paradigm and it gained some popularity through the years, but never enough to be a rival of Red Hat and other distros.

Debian

Debian had many forks in this period, but the three most significant ones were Xandros, Knoppix, and Ubuntu.

- **Xandros:** Originally named Corel Linux, Xandros was a commercial distro based on Debian and first released in 1999. The most significant fact was the availability of Corel WordPerfect Office on Linux. It was a clearly an attempt by Corel to compete with the Windows/Office duo. It's discontinued.
- **Knoppix:** This is a German Live CD/DVD distro that was distributed with many computer magazines, a popular trend at the time. It was released in 2000 and soon become very popular because of its approach. Given that it didn't need to be installed to run/test it, it was tested and used by many users that were new to Linux.
- **Ubuntu:** Clearly the big revolution of this era, Ubuntu was the distro that helped the most to popularize the Linux OS. To a great number of people it is a synonym for Linux. It was the most forked one of this era and still is. Originally released as a Debian fork in 2004, many Debian users switched to it.

Originals

This period also had some new, original distros, and some of them would become very popular years later and the base of many forks.

- **Gentoo**: It was released in 1999 by Daniel Robbins. Its goal was to create a distro without precompiled binaries that could be specifically tuned to the current hardware and tailored by the user's needs. Advanced users received this new distro very well.
- **Linux from Scratch (LFS)**: This was never a popular distro, but its unique approach made it a significant input in this era. It is not properly a Linux distro; instead it is a kit to make your own Linux distro. It was originally released in 1999.

- **Arch**: Inspired by CRUX (another distro of this period), Arch became very popular among advanced users and was later a base for many forks. Its goals were minimalism and simplicity as well as power and the ability to stay up-to-date. It was released in March 2002.
- **NixOS**: A distro built around a revolutionary package manager, Nix, it is still a state-of-the-art distro. It was not well known, and it is without forks, but it still was a significant event of that era.

Slackware

Slackware continued to have fans who were very loyal to the distro, but it was not as popular as Debian or Red Hat. It had some forks, like Slax and Vector, but far less than the number of forks from Debian/Red Hat.

Figures 3-4 and 3-5 show the last period in the history timeline of Linux distros. This period is explained in the next section.

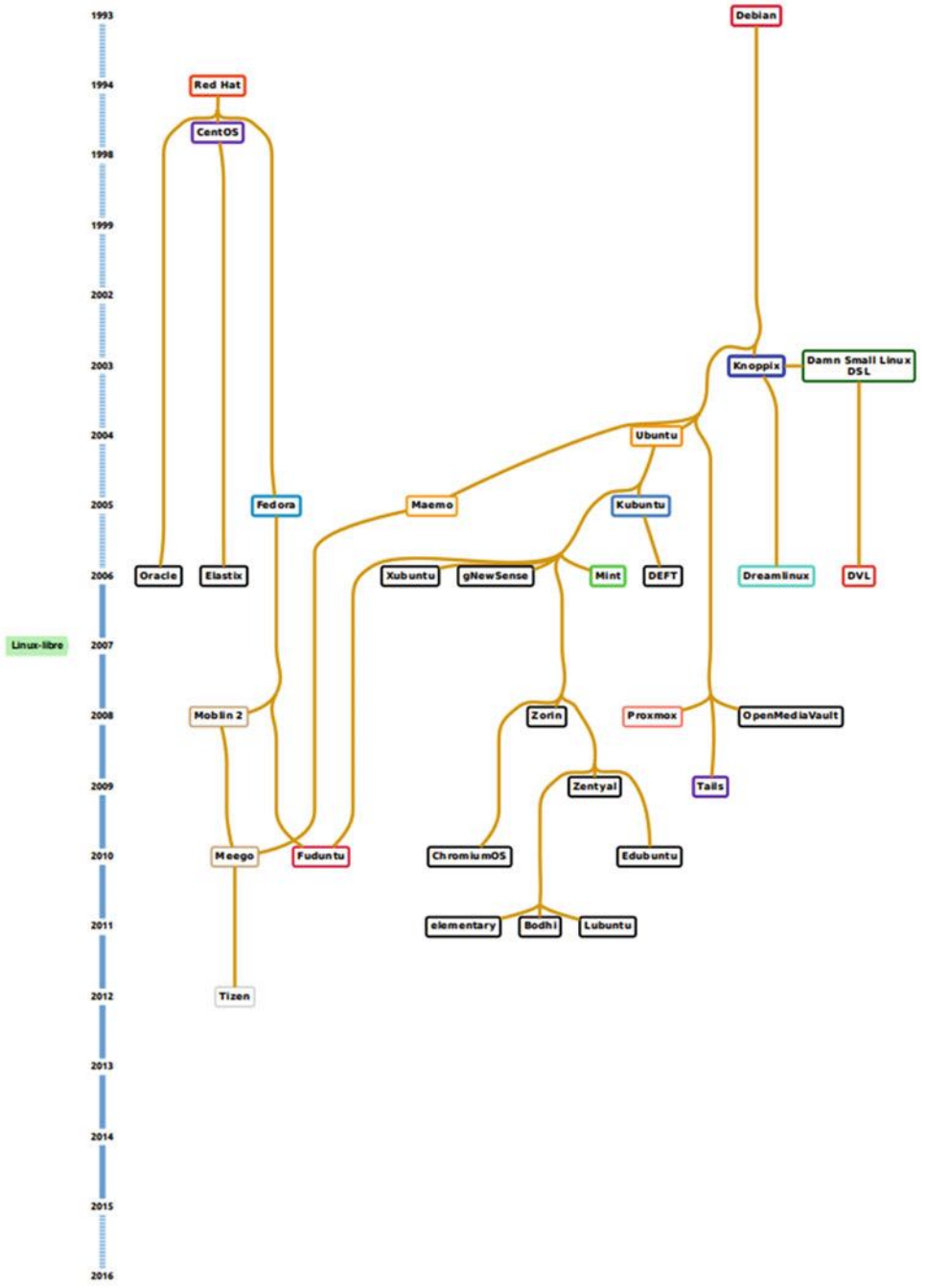


Figure 3-4. *The Linux distro timeline: 2005-2015, Part I*

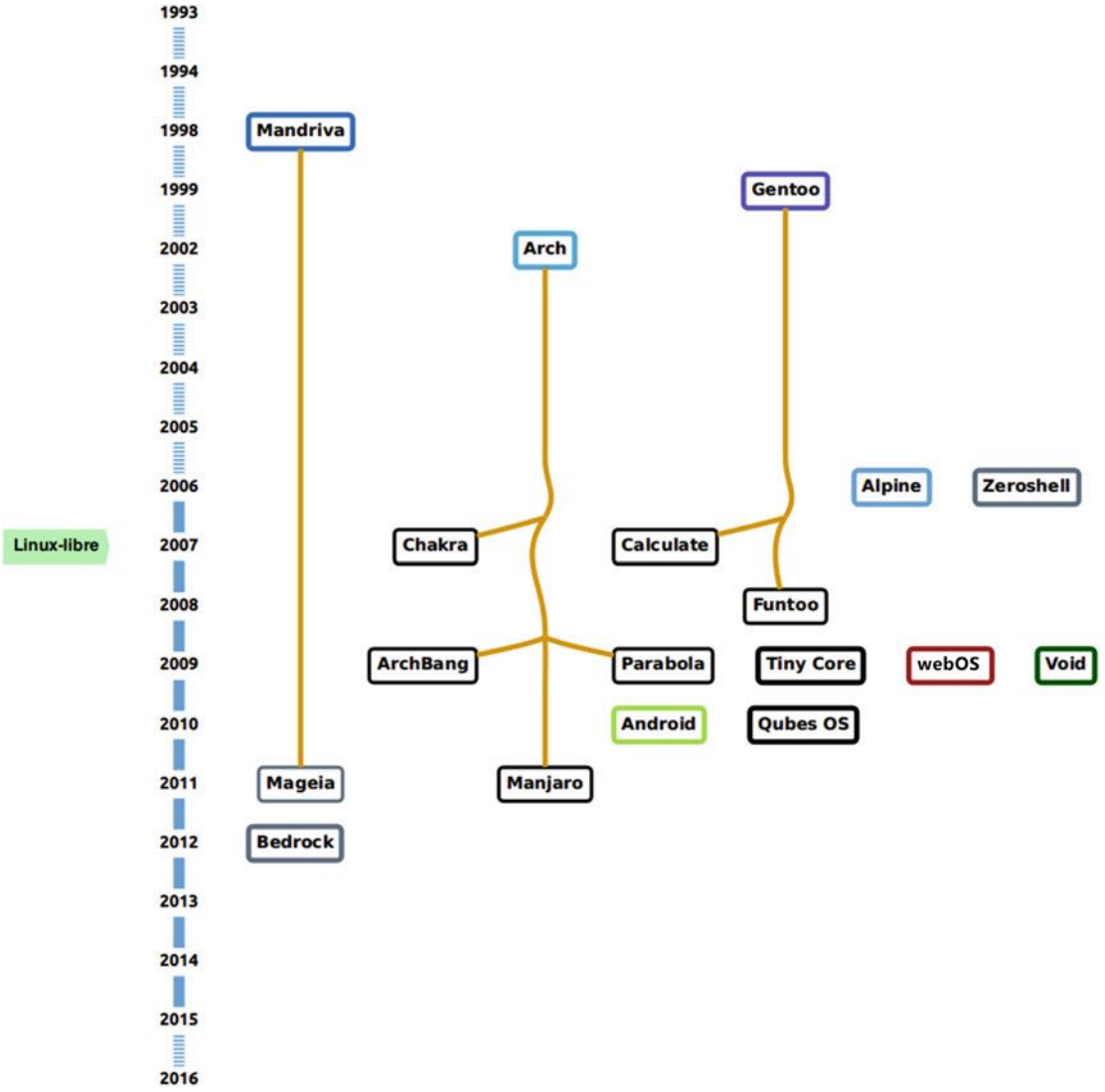


Figure 3-5. *The Linux distro timeline: 2005-2015, Part II*

2006-2015: The Consolidation—A Few Stars, Many Planets

The period of 2006-2015 is a time of consolidation of a few stars, namely Ubuntu, Debian, and Red Hat, each of which have many forks (especially Ubuntu) and continue to grow in popularity. Slackware and SUSE continue to have a significant weight but their eras of active forking are long over. On the other hand, Arch and Gentoo get established as popular distros and have many forks. And there is still room to appear new original distros, some as revolutionary as QubesOS (not popular) or Android (currently the most-used Linux distro in the world).

- **Mint**: The most popular fork of Ubuntu, it's so popular that it surpassed the fame of Ubuntu recently on distrowatch (but I doubt that it has as many installations). Only two years after Ubuntu was released, Mint was born to provide a different "flavor" than the original distro. Currently, it has alternative versions that are based on Debian instead of Ubuntu.
- **elementary OS**: It was born in 2011 to provide a nicer and simpler Ubuntu. In fact, the origins of the distro are in a Gnome theme.
- **Mageia**: Born from the ashes of Mandriva in 2010, its goal is to continue to provide the same experience to the orphan users of the previous distro.
- **Android**: It is not properly a Linux distro in the strictest definition, but it is a Linux-based OS, and many consider it a distro. It was released in 2008 and is focused on mobile devices like tablets and smartphones.
- **Tails**: The distribution for those who care about privacy and anonymity, Tails is based on Debian and was originally developed in 2009. It's a Live CD distro, and it had a significant impact in the security field because of the Snowden revelations.

Summary

This chapter presented a brief review of the evolution of Linux distros and how they can be grouped as "families" based on their origins. Now imagine how long this chapter would have been if I charted the about 700 distros currently available (more than half of which are discontinued). I hope that this chapter helps you understand the origins of the Linux distributions and how they are related.

In the next chapter, I show you the Ubuntu Linux distribution.

PART 2

General Purpose Distros

In this part of the book, I cover ten Linux distributions in detail. All of these are general-purpose distributions, which is the most common type. This is the core of the book; these chapters give you a better perspective on each distro and provide a good demonstration of how one distro compares to another. These chapters also provide real-world examples of the crucial decision points presented in Chapter 2. Plus, these chapters save you a lot of time testing distros in order to pick the right one for you.

The following chapters provide a brief description of each of the distros as well as some of the history and philosophy of them. You also learn how to install a Linux distro, maintain it, and discover its pros and cons. Finally, I include a brief compilation of all of these points for each particular distro.

Why Only These Particular Distros?

I cannot analyze in detail all of the available Linux distributions if I want to keep this book at a reasonable length and price. Thus I had to be selective: I chose to cover the main distributions in depth, but I also wanted to show you a wide spectrum of what is available. To that end, I picked the popular distros plus two advanced ones and even one "experimental" distribution. The Linux distributions covered in detail in this book are (in order):

1. Ubuntu
2. Fedora
3. Debian
4. OpenSUSE
5. Mint
6. Mageia
7. elementary OS
8. Arch
9. Gentoo
10. Slackware
11. NixOS

After covering these distributions, I dedicate one additional chapter to some of the other distros in a summary fashion; the idea here is to give you a glimpse of these distributions without going into the amount of detail provided on the others.

CHAPTER 4

Ubuntu

Ubuntu is one of the most famous Linux distributions (and the most used, too), so it's probably the first one to come to mind among people who are not advanced Linux users. In fact, many people think Linux and Ubuntu are the same thing. This is particularly remarkable because Ubuntu is a relatively recent distribution (only 11 years old as compared to its "parent" distro, Debian, which is 22 years old). Obviously, Ubuntu has done something very well in order to achieve such recognition. I can safely say that when it comes to Linux history, you can divide it into the era before Ubuntu and the one after it. I think this is reason enough to start my distro analysis with Ubuntu.

History

In the mid-1990s, a South African entrepreneur named Mark Shuttleworth (see Figure 4-1) founded a digital certificate authority and Internet security company called Thawte, which would become the second largest company on the Internet (until its main rival, VeriSign, purchased it in 1999 for several hundred million dollars). Suddenly rich, Shuttleworth decided to use the money to achieve some of his dreams. After founding a non-profit organization (The Shuttleworth Foundation, or TSF), a venture capital firm (HBD for "Here Be Dragons"), and becoming one of the first space tourists (all in the first three years after the sale), he took a step in 2004 that would make him one of the most recognizable figures in the Linux world, and the catalyst of some of the actions would shape how Linux would be seen by the rest of the world in the future.

Figure 4-1. *Mark Shuttleworth at Linuxtag 2006 at Wiesbaden, Germany*

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_4

As a long-time Debian user, maintainer, and developer (and general open source advocate), Shuttleworth firmly believed that Linux (as a distro) could be brought into the mainstream and thus compete with the major operating systems of the time, meaning Windows and Mac OS. He had a strong commitment to open source and Software Libre, and wanted to give back to the community, as it was the basis of his former company and its success. However, he thought that bringing big and radical changes to Debian would be a huge enterprise given its community management nature. So, the easiest and best solution was to use Debian as the base for a new distribution.

In April of 2004, he met with about a dozen developers from the Debian, GNU Arch, and GNOME projects to outline a better Linux distribution. They called themselves "the Warthogs" and in six months they would shape what would become the first Ubuntu release, the "Warty Warthog."

Shuttleworth also founded and funded Canonical Ltd., a UK-based company to provide the commercial support and services for Ubuntu. The name of the company is a clear message: they want to be the canon for all other. The name chosen for the distribution, Ubuntu, is also a statement. Ubuntu is word of Zulu origin (one of the South Africa tribes) that means "humanity to others;" it also means "I am what I am because of who we all are." This is the spirit behind Ubuntu.

Ubuntu soon won several prizes and recognition from the specialized media, and then it quickly became the most famous distro. I personally used the distro as my main OS from 2005 to 2012, and I remember it was already on the tongue of almost every advanced Linux user. Today, Canonical estimates that there are 40 million Ubuntu desktop users and counting. Ubuntu is also very prominent in the server market, especially in the cloud, where it is one of the most installed distros. Moreover, Dustin Kirkland from Canonical recently claimed that there are over a billion people using Ubuntu, both directly and indirectly, through servers, cloud instances, virtual images, phones, and so forth[1].

Criticism and Controversy

Ubuntu has had to confront a lot of criticism and controversy. Shuttleworth and Canonical have a particular vision of the path Ubuntu must follow to be a competitor in the desktop market and other areas, and they have not hesitated to make innovations and changes that move away from the Linux and Free Software tradition of other distributions. Some users are very conservative and they do not like abrupt changes, especially if they love Ubuntu.

As a result, Ubuntu has gone from being the one of the most loved Linux distro to one of the most criticized, and it has lost many users. I suppose this is the price you pay when you are the king of the desktop Linux distributions. I feel that in the past few years, since the decision to make Ubuntu a ubiquitous distro in search of convergence, it seems to have lost its way a little. I hope that in the future Ubuntu gets back to being the amazing and revolutionary distro that changed the Linux world forever.

Philosophy

The original philosophy of Ubuntu was to create the best Linux distro available, and to compete with the other major actors in the OS arena: Windows and OS X. Today the philosophy has not changed too much, but these days Canonical wants to conquer the desktop and server markets as well as mobile devices, the cloud, and the Internet of Things. Ubuntu wants to be everywhere and to be the best.

Originally Canonical wanted to offer a distro that was easy to use, accessible, localized, and driven by the community. It also wanted to be predictable, so frequent releases were a crucial point. This is still true today, and this goal has expanded to offering a ubiquitous experience where you can have the same distribution in almost every device.

[1] http://insights.ubuntu.com/2015/12/22/more-people-use-ubuntu-than-anyone-actually-knows/

Distro Selection Criteria

Now that you know its origins, let's look at how Ubuntu does against the distro selection criteria discussed in Chapter 2.

Purpose and Environment

Ubuntu is mostly a general purpose distribution, but lately Canonical is looking to be a distro that can be installed on all types of hardware far beyond the traditional laptop/desktop arena, like mobile devices (smartphones and tablets), the cloud, servers, and the IoT (the Internet of Things, meaning electronic devices connected to the Internet). To achieve this goal, it has created different versions of the same distribution, fitted for particular tasks. Thus we can say that Ubuntu is a general purpose distro and a task-oriented one as well, depending on which flavor you use.

Currently Ubuntu has "versions" of its distribution for different environments, but not all are available for download.

- **Desktop**: This is the traditional version of the distro, which is general purpose and oriented to workstations, desktops, and laptops.
- **Server**: This version is installed in servers, so it is a task-oriented one.
- **Cloud**: Available as images for the most common public clouds such as Amazon AWS, Google Cloud, or MS Azure, this version also has a complete OpenStack product. As part of its orientation to the IoT, there is also Ubuntu Core, which can be used in cloud environments.
- **Kylin**: This is an adaptation of the desktop version for the China market and it complies with the Chinese government's procurement regulations.
- **Phone (Mobile)**: This version is not available for download from the web site but it comes with some smartphones and tablets as an OEM OS. This is a recent adventure for Canonical and it is still in its first stages, so the devices available are limited.

Support

Ubuntu is a well-supported distribution, offering both commercial and community support (even when they have the same distro for both cases). For enterprises, there is Ubuntu Advantage service, which includes tools for management, automation, deployment, and assistance like Landscape (an exclusive technical library), 24/7 telephone support, and optionally an on-site dedicated Canonical support engineer. For that, you have to pay a quota annually based on your installation size. You can learn more about Ubuntu Advantage at `www.ubuntu.com/management/ubuntu-advantage`.

As for the free support from Canonical, you can use two resources:

- **Documentation**: `https://help.ubuntu.com/`
- **Technical Answer System**: `https://answers.launchpad.net/ubuntu`

Also, you can get free community support from various channels:

- **Wiki**: `https://wiki.ubuntu.com`
- **Forums**: `http://ubuntuforums.org/` or `http://discourse.ubuntu.com`
- **Ask Ubuntu**: `https://askubuntu.com`

- **IRC**: https://wiki.ubuntu.com/IRC/ChannelList
- **Mailing lists**: http://community.ubuntu.com/contribute/support/mailinglists

Of course you can contribute to the community and give back as much as you want/can to help to sustain it at http://community.ubuntu.com. Ubuntu is so widely used that you can also contact and participate in one of the many local communities (LoCo) all around the world; go to http://community.ubuntu.com/help-information/meeting-other-ubuntu-users/local-communities.

This level of support, huge not only in size and means also in quality, is something that you cannot find in other distributions. This is one of the strongest points about Ubuntu.

User Friendliness

User friendliness was always a core goal of the distribution. Ubuntu is a very easy-to-use Linux distro; it is especially recommended for newcomers to the Linux world. It has its ways of doing things, and not all people like the Unity desktop environment, but once you get accustomed to it, it is very easy to use and intuitive. Also, it tries to be transparent and automate as much as possible, from hardware detection to maintenance updates. This plus the great support make it one of the distros that I recommend to beginner Linux users.

Stability

Out of the box, Ubuntu is a very, very stable Linux distribution. You can install a normal release (not a LTS one), and use, maintain, and upgrade it without problems for several years without needing to do a fresh install. There's also the option of using a LTS release to get an extra assurance of stability and security. The LTS (long-term support) versions are oriented to those who need a headache-free OS install for their systems because they depend on it for their job or simply because they don't like to spend too much time maintaining their OS.

Canonical uses a variant of the standard release scheme for their distro. There are normal releases, with a period of six months, and there are LTS releases with a longer period (about four standard releases in two years). The standard releases have a maintenance period of nine months, after which they cease to be officially supported. The LTS releases are supported for five years, and are generally more stable because they lack experimental or partially finished functionalities. You can see a calendar and list of releases at https://wiki.ubuntu.com/Releases.

The Ubuntu releases have a double denomination of a number scheme and a codename. The number scheme is in the format YY.MM, corresponding to the year and month of the release. Usually the releases happen in the 04 and 10 months (April and October). The codename is always composed by an adjective and an animal, usually funny, like the current one, the 16.04 or "Xenial Xerus" release.

Hardware Support

Ubuntu's hardware support is probably one of the best. The most common hardware is detected automatically; even when it is a graphic card you can usually choose between installing an open source or a private driver for it. When the hardware is not known, it is often easy to find an alternative source for a driver from the hardware's company or from the community because Ubuntu is such a popular distro; smaller distros don't usually get specific drivers made for them. And in a worst-case scenario, you can often use a Debian driver.

When a computer company decides to sell its machines with a Linux distribution as their OS by default (or optional), usually Ubuntu is the chosen one. This is true for companies as big as Dell, HP, Asus, and Lenovo to the specialized ones such as System76.

As a whole system, laptops are the most problematic hardware, but there are companies that make laptops that work particularly well with Linux, such as Lenovo, HP, and Dell. Canonical has a site and a program to certify hardware that runs without problems with Ubuntu; it's called Ubuntu certified hardware, and the site for desktops is http://www.ubuntu.com/certification/desktop.

Aesthetics

Shuttleworth tried from the first version to create a good-looking distro, and Canonical continues this effort. The logo is a very recognizable icon in the Linux world and even outside it, and it is also a clear symbol of its principles. It is easy to see that Canonical takes care of the aesthetics in every aspect of the distro, from the web site to the distro itself. In the past, when the company freely sent CDs to your home, the CD envelope was always very well designed. The design has changed over time, and it's obvious that it continues to evolve. But this is a very subjective topic and many people don't like the color palette, which is several tones of orange and aubergine. This color palette forms part of the corporate image and it is strongly linked to Ubuntu, so it will probably never change, but it has more than one or two detractors.

Desktop Environment

From the very beginning, Ubuntu used the Gnome desktop environment. The first derivatives, Kubuntu and Xubuntu, used KDE and Xfce for the desktop by default as the main difference. But since the 11.04 release, Ubuntu uses its own Unity desktop interface. It was conceived initially for the purpose of having a unique interface that could be used on all of the devices available (originally notebooks but now laptops, tablets, and smartphones).

The launch of Unity was a great controversy in the Ubuntu community. Many users migrated to other distros (like Mint) because they disliked the new interface. This is still a problem today. Two new official flavors, Ubuntu Gnome and Ubuntu MATE, exist only for those who love Ubuntu but hate Unity.

If you want to see all the different official flavors available, go to https://wiki.ubuntu.com/UbuntuFlavors (not all of them are based on a different desktop environment).

Init System

Until very recently, in fact until the 15.04 release, Ubuntu used its own init system named Upstart, which was also adopted by other distros like Fedora. (Actually, in the first few years, it used the classic sysv init, System V).

But in the last releases Ubuntu decided to join to the majority of distros and adopt the new kid on the block, systemd. You could find Upstart in their penultimate LTS release (14.04) but it's now deprecated in the current LTS release (16.04).

You still can use Upstart in the last release by installing it and switching to use it permanently, but this is not something that I recommend to any non-advanced user. In the end, the future seems to be a systemd one in almost every Linux distro out there (with honorable exceptions).

Package Management System

Because it was initially a derivative of Debian (it still depends a lot on this distribution), the package management system is the same as the one used there, dpkg. Ubuntu, of course, has its own package repositories and its own graphical tools, but it uses the classical shell tools of Debian, apt-get and aptitude. The typical user will use the graphical tool for this job, Software (actually gnome-software), but is not unusual to need to use the shell tools when you have to install a very rare package or when you want to use a PPA (you can do this graphically, too).

The PPAs (Personal Package Archive) are a unique characteristic of Ubuntu (there are similar things in some distros, but not exactly like this). They are a way to keep your own packages maintained by you in a personal repository. This is frequently used to add software that is not in the official repositories, but it is also a big source of headaches and problems when upgrading, and it could cause big security risks. You have to trust in the owner of that PPA if it is not yours and vice versa. Usually people use PPAs to run the latest versions of software not yet available in the repositories; it's a way to emulate rolling releases in a certain way, but it's wrong.

Another innovation in package management that Canonical introduced recently is Snappy; it was previously only supported in Ubuntu Core and Phone, but now is part of the Desktop versions, since the 16.04 release. Snappy is sort of a mixture of the Nix (and Guix) package managers, and the Apple OS and Android ones. Basically Snappy uses transactional updates (which can be rolled back) and atomic packages, reducing the complexity and the conflicts between packages.

Architecture

The Desktop version of Ubuntu Linux officially supports only the 32 and 64-bit Intel/AMD architectures. The Server version offers releases for the ARM and IBM Power8 platforms. This covers the majority of use cases for the normal user, but if you need to use Ubuntu in other platforms you must rely on older or unofficial versions. Obviously tablets/smartphones that use the Phone version are supported but there it comes as a bundle item (Ubuntu as OEM).

Security/Anonymity

Ubuntu is a reasonably secure Linux distro out of the box. It is not a paranoid-level security distro, but it uses the AppArmor security module and its defaults are enough for the majority of users. Of course, you can always make it more secure; you can install the tools and configure them, but you need to know what you are doing because you can easily make it worse. You can take some extra measures during the installation itself, like encrypting your home directory or the entire disk and always using a password (and a strong one). The official documentation offers great advice; go to https://help.ubuntu.com/lts/serverguide/security.html.

In fact, the major risks that an Ubuntu user could suffer in terms of security result from user behavior. Things like not using a password to log in, using a weak password, or using unofficial repositories or PPAs are the most common threats.

As for the anonymity part, Ubuntu is generally respectful in this matter, leaving it up to the users whether they send their information to the company to help to improve the distro. However, over the past few years there has been a great controversy because of the integration of Ubuntu Search with Amazon and the inclusion of a direct link to the store. Later there was an option to deactivate the Amazon search; finally, in the current release of 16.04, the online searches are deactivated by default. From a financial point of view, I suppose Amazon's financial support for this feature helped Canonical to balance its books.

Principles and Ethics

From the beginning, Shuttleworth and Canonical had a strong sense of ethics and principles based on Software Libre and open source. The name and the logo are a strong declaration of their principles. But as the years went by, Canonical and Shuttleworth adopted a more flexible and pragmatic approach to these matters in order to achieve its main goal, to be an alternative in the desktop market and gain a bigger share of users. This flexibility meant that the distro could ship both binary and proprietary drivers for things like graphics and network cards. Also, the user was given the option of installing proprietary or even commercial software from the installation (like mp3 support) or from the Software Center.

Last year there was some criticism about how Canonical was moving away from its core community. There is some truth in this because canonical did lose a bit of focus when it shifted to making a ubiquitous OS for all types of devices. But Canonical is a company and it has to make money.

Live CD

The desktop version of Ubuntu is also a Live DVD that you can use to test Ubuntu without needing to install it. You can also use it for security reasons, such as accessing the Internet without leaving any footprints in the system or if your current OS (for example, Windows) is infected or insecure. Advanced users can also use it to fix/repair problems with a current Ubuntu installation. Older versions usually fit onto a CD but current versions require a DVD.

Professional Certification

In the past, Canonical offered a professional certification for their systems, but this has been discontinued. That certification was equivalent to the current LPIC-1 from the Linux Professional Institute and was a very basic one. As Ubuntu is based on Debian, there are some certifications that are generic and valid for this distro, such as the previously mentioned LPIC series or the offerings from the Linux Foundation.

Installation

Installing this distro, or any other, is never a straight path. There are multiple options that allow you to personalize your installation to your needs or preferences. I am going to cover the most common installation for this particular distro, but I will show you other ways in other distros throughout this book, with the purpose of giving you a wider spectrum of what Linux (as a distro) is; I won't just stick to the almost automatic "push next ➤ push next" procedure.

The first thing you have to do is download an installable ISO image of the desktop version of the distro from `www.ubuntu.com/download/desktop`. You can also purchase a USB stick from the shop at `http://shop.canonical.com`, but the only option there is to get the last LTS release of the distro for 64-bit Intel/AMD architectures, as you can see in Figure 4-2.

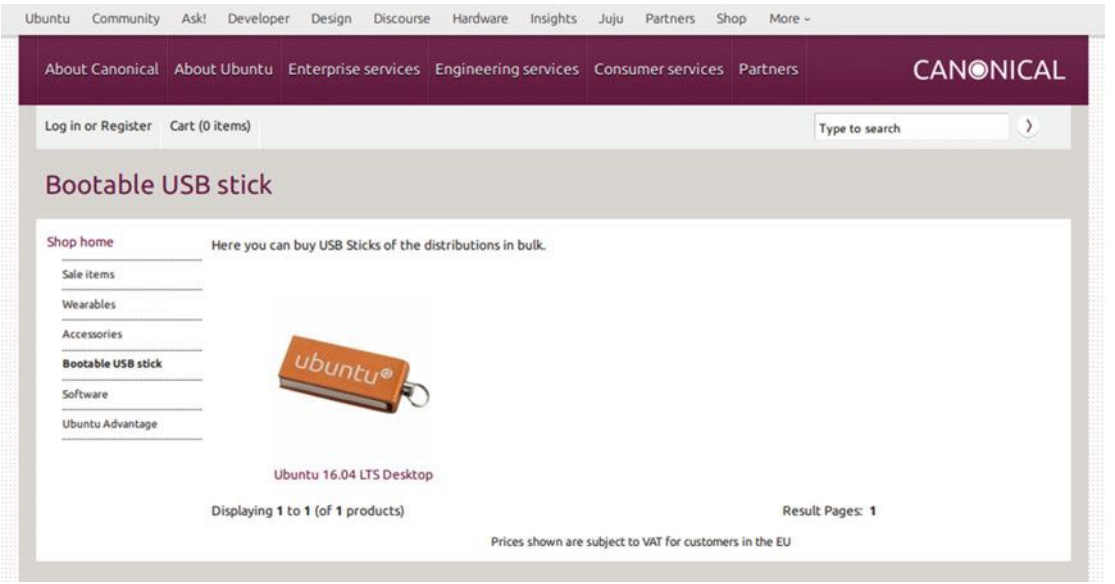


Figure 4-2. *The Canonical shop, where you can purchase a Ubuntu LTS release on an USB stick*

Figure 4-3 shows the download page. There are usually two main options to choose from: the latest LTS release and the latest normal release (they are called "versions" instead of "releases," but I will stick to the term "release" for consistency). If you use your computer for work or you do not need to have the latest versions of the software, I strongly recommend that you install the LTS release; in other cases, choose the latest normal release and you are good to go. At the time of writing this book, the corresponding release is the 16.04 LTS.

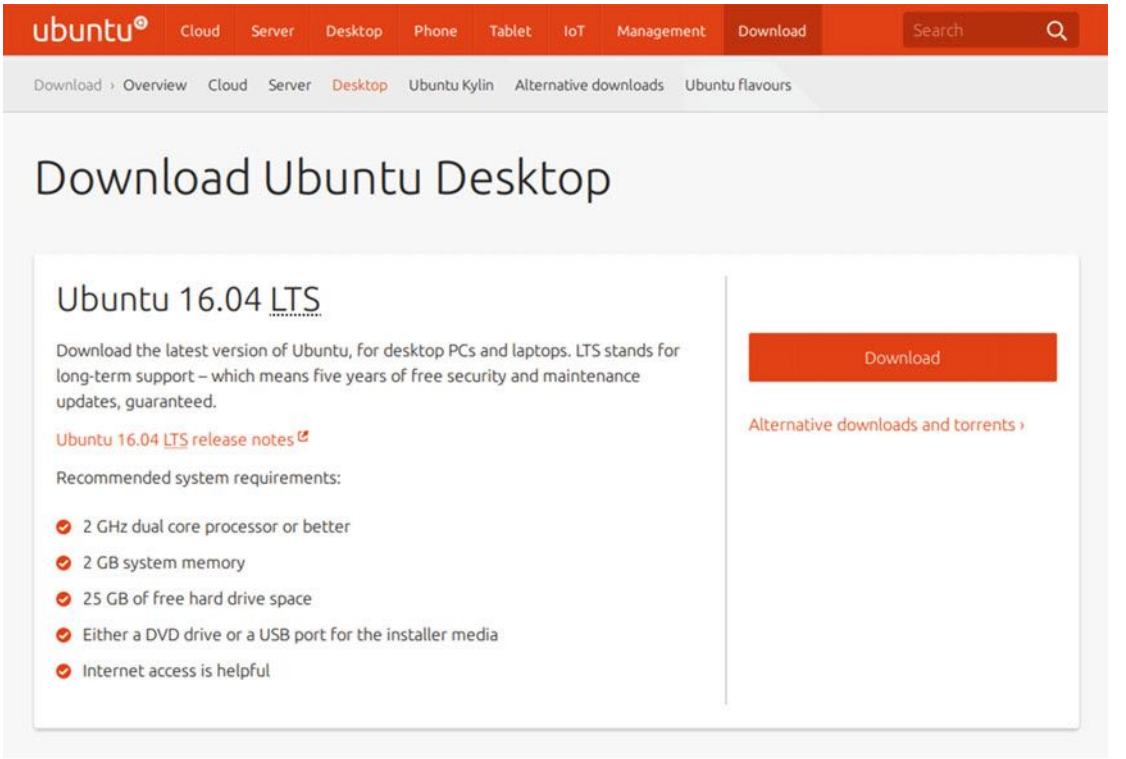


Figure 4-3. *The Ubuntu desktop download page*

I have chosen the latest LTS release, 16.04 for 64-bit, to show you the Ubuntu install process because it is the one that the majority of people would choose. After you press the button to download the distro, Canonical will show you an intermediate page to contribute to the maintenance of the distro (see Figure 4-4). You can make a donation (you can purchase merchandise from their shop for the same purpose) or you can skip it and continue the download.

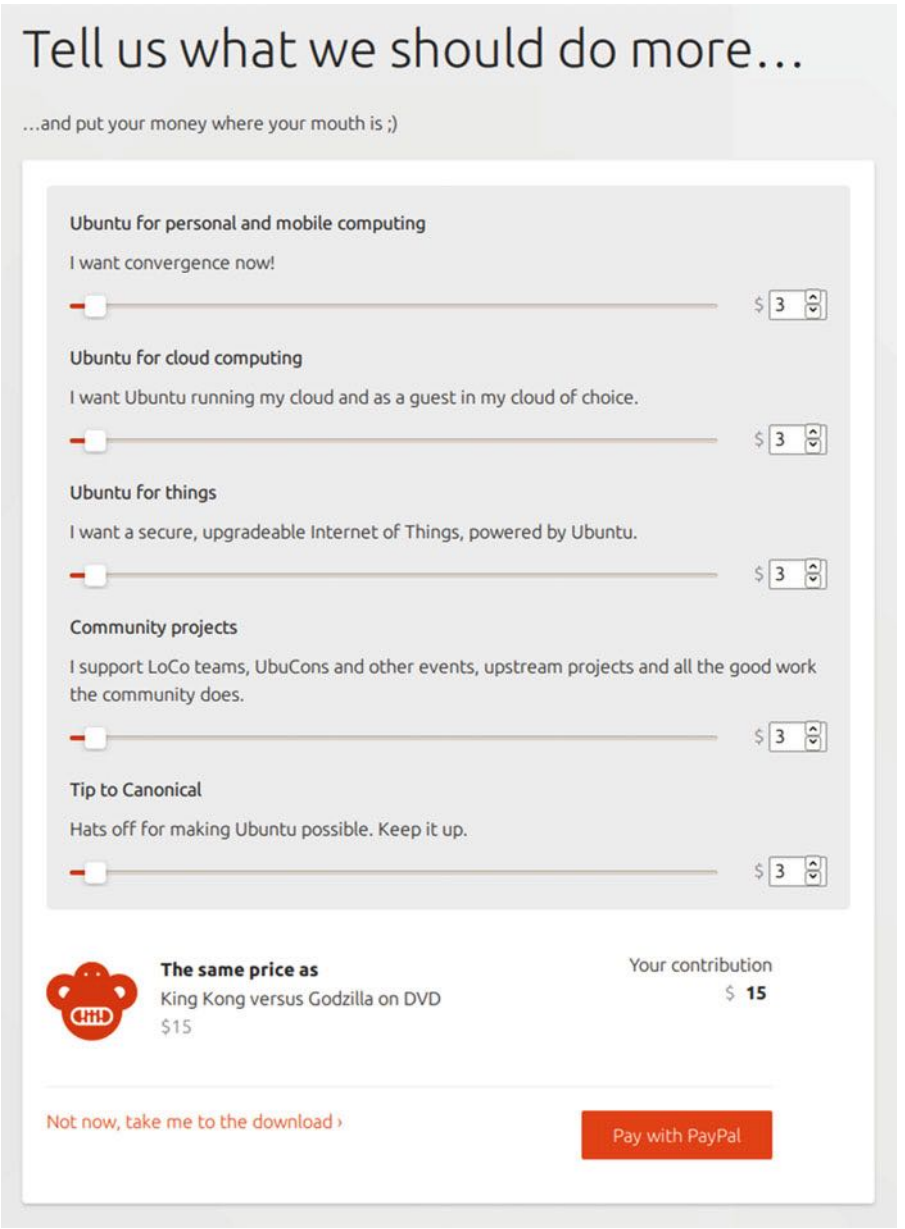


Figure 4-4. *The intermediate page where you can donate to the project*

You can also choose an alternative download (there is a link on the download page) in case you need an older release, a network installer (to download all you need from the Internet at install time; this is only recommended for fast online connections), or you want to use the BitTorrent protocol to download the ISO image (see Figure 4-5).

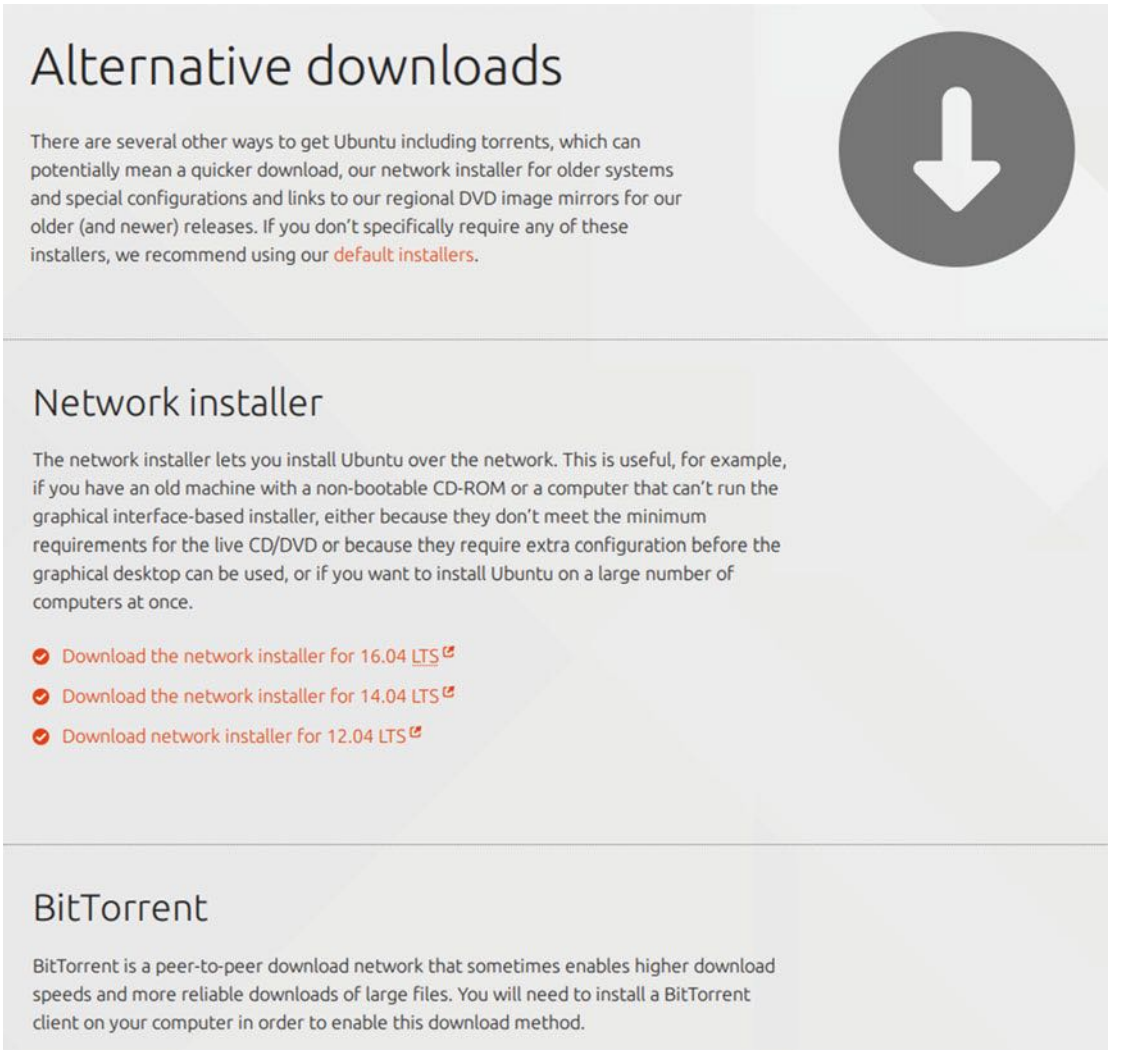


Figure 4-5. *The Ubuntu alternative downloads page for the desktop version*

After downloading the ISO image (the 16.04 release size is about 1.4GB), you can burn it onto a DVD or put it on a USB drive. After you boot from that Ubuntu image, you get the first screen, which is a black screen with an image at the bottom like the one shown in Figure 4-6. This screen only lasts for several seconds.

Figure 4-6. *The first screen that appears in the boot process*

If your press any key in this screen before it disappears, you will jump to a screen like the one shown in Figure 4-7.

Figure 4-7. *The text installation menu of Ubuntu*

This screen is a text menu that allows you several options that are not available in the regular graphical installation process. It allows you to test your DVD disk or your memory, usually to diagnose some problem that happened in a previous installation attempt. But you can also set other advanced options using the function keys, as shown in the bottom menu, like making an OEM or an expert installation, or avoiding certain hardware detection problems, or changing the default language of the installation. These options are intended for intermediate/advanced users and I don't recommend trying any of them, so do not press any key in the previous screen and let's continue the booting process.

The next screen that appears is another black screen with an image in the center with an animated graph that shows you the activity while it is booting (see Figure 4-8).

Figure 4-8. *Image shown on the boot progress screen*

At the end of the boot process you get your first screen in the graphical environment, which is shown in Figure 4-9.

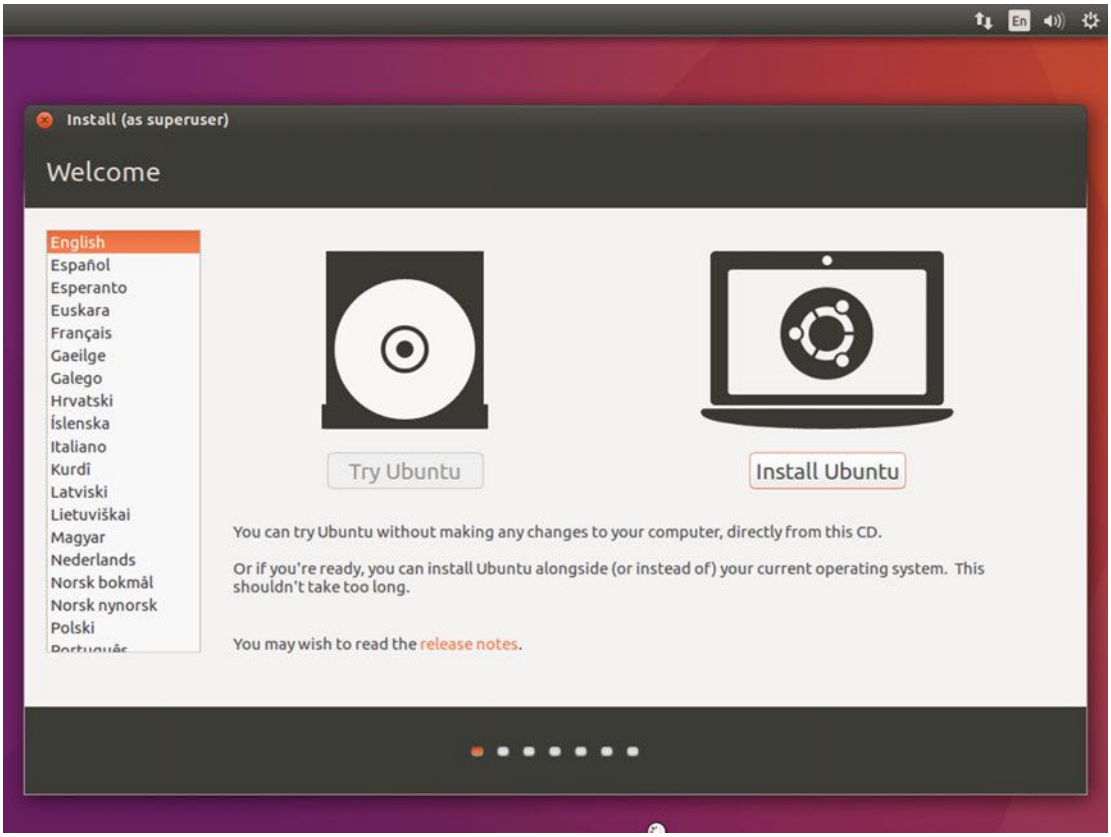


Figure 4-9. *The Welcome screen after booting the Ubuntu ISO image*

This screen offers two main options: "Try Ubuntu" and "Install Ubuntu." You can also set the language used in the install process. In the menu bar at the top you can also set other options (some of them are equal to the text menu I showed you before) like network or accessibility options. If you choose "Try Ubuntu" you are going to initiate a Live session of Ubuntu in which you can test a lot of things without writing anything in your hard disk, and you can always continue the install process from there. Skip that process. Instead, press the "Install Ubuntu" button.

The next screen (Figure 4-10) checks if your computer has enough disk space and a connection to the Internet. An Internet connection is not necessary to install the distro, but if you have one available and you check the option "Download updates when installing," the distro is going to install the available updated packages instead of the old ones in the ISO image. I recommend checking this option for two reasons:

- If you want to keep your Linux up-to-date (and I strongly recommend that you do so) you must spend extra time to download the packages after the installation when you make your first update anyway. So, if you're not in a hurry and you have a good Internet connection, do it now.

- If any of the updates are security updates, the first time that you boot up your new Linux and go to use the Internet you will avoid any security holes that could affect your system/data because you hadn't updated the system yet.

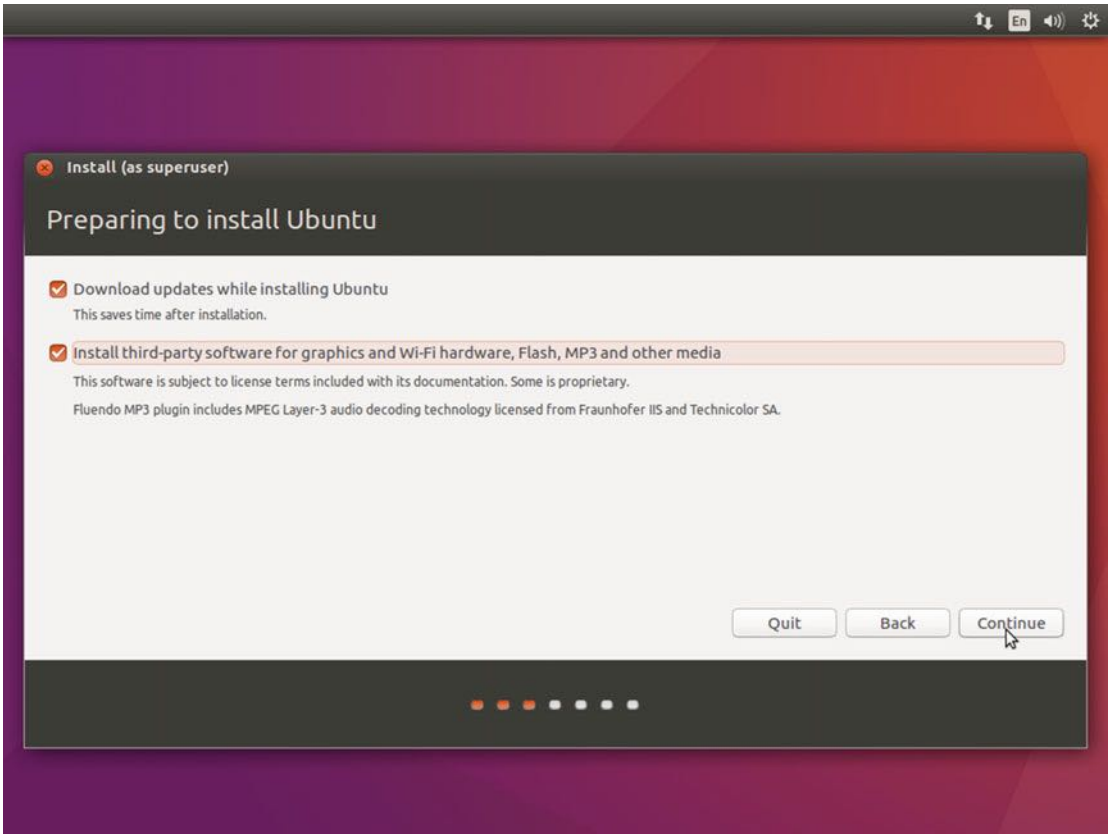


Figure 4-10. *The first step in the Ubuntu installation process*

After releasing a new version of the distro, even the day after, there are often new package updates. In Linux and generally in open source, it is very normal to have regular updates, even daily ones. If you use a distribution like Ubuntu perhaps you won't have updates every day, but if you use a rolling release distro you can count on one hand the days without any updates (of course it depends on the number and variety of packages that you have installed in your system).

The last option on this screen is to install proprietary software like graphics cards and wireless drivers, Adobe Flash, or the mp3 Fluendo plug-in (to allow several applications to reproduce .mp3 files). The reason why this is optional is that this software is not under a free license and some people do not agree with that. It is up to you to install it or not; however, for example, due the prevalence of the mp3 format in audio files, I chose to install the plug-in so I can play those files.

Press the Continue button. The next screen that appears is the one in Figure 4-11. This is the most complex part of the installation because you have to make a few critical decisions. Basically this part is deciding how you are going to manage your hard drive(s) to install Ubuntu. There are two main options here: the automatic one and the manual process.

- **Erase disk and install Ubuntu**. This is the automatic process. Ubuntu is going to erase your entire hard disk (and all of the data) and make new partitions with an automated calculated size. This is only recommended when installing on a fresh machine. In this option you have two other choices:
  - **Encrypt the disk**. This encrypts your entire hard disk, and only you can access its contents with a password. CAUTION! With this, if you forget your password, you lose all of your data. I only recommend this for intermediate/advanced users with experience with Linux. Also, always make backups of your data! Another thing to note is that this option affects the performance of your hard disk because your CPU must encrypt/decrypt your data continuously. If you do not have a modern machine, a good amount of memory, and a fast disk (better, a SSD), I do not recommend activating this option.
  - **Use LVM**. This is a modern way of managing disk volumes that is more flexible and powerful than the traditional partitioning scheme. I only recommend this for intermediate/advanced users because if you are not going to change your disk volumes, it is not necessary. If you are planning to do so in the future, activate it now.
- **Something else**. This is the complete manual procedure. You have to choose the partitions (number, size, type, mount points) and whether to encrypt them or not. This option does not allow you to choose LVM; you must use the shell and a series of commands to do it. It's only recommended for those who know what they are doing.

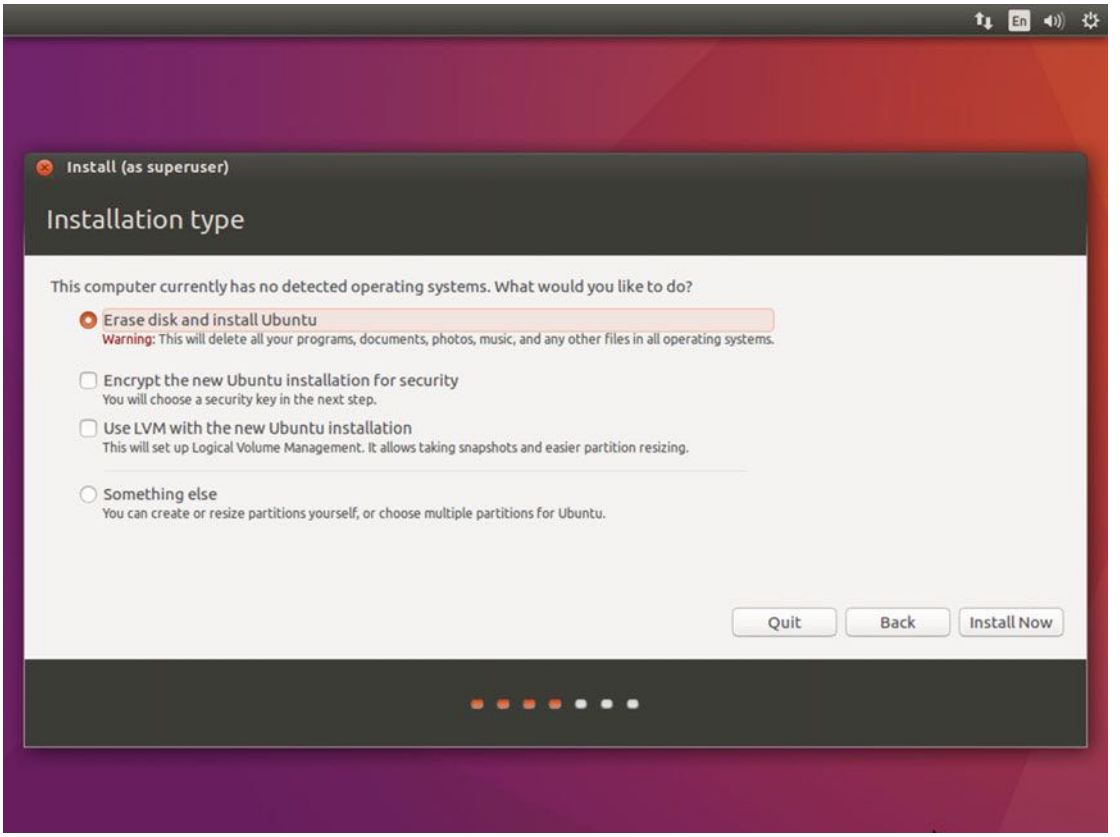


Figure 4-11. *The installation type screen where you choose how to manage your disk*

In this first review of a distro, my intention is to show you how easy it can be to install a new Linux distribution in a fresh system. Later, in different distros, I introduce you to other ways to install a distro, to avoid being boring and repetitive, and to show you how to do certain things in Linux. Because of that I'll skip any advanced options here.

If you press the Install Now button, it will show you a resume (Figure 4-12) of the changes that are going to be made to the hard disk(s). This operation is irreversible and that's reason enough to show you the information. Usually it will create two partitions: a root one in the ext4 format and a swap one. Because I am installing Ubuntu in a fresh system, I can press the Continue button without worry.

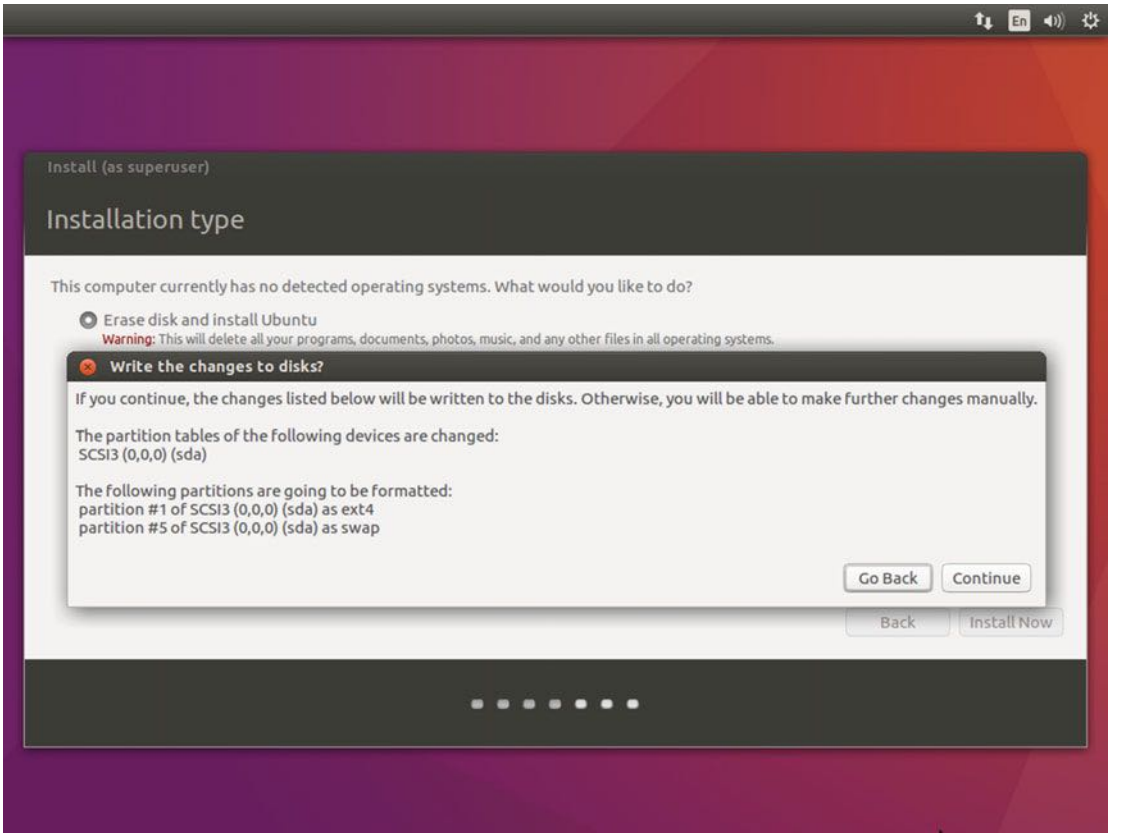


Figure 4-12. *The confirmation step before making changes to the hard disk*

Figure 4-13 shows an interactive map for choosing your time zone to correctly adjust the date and time. Usually Ubuntu will automatically detect your current time zone, but you can always choose it manually by navigating to the appropriate area in the map.

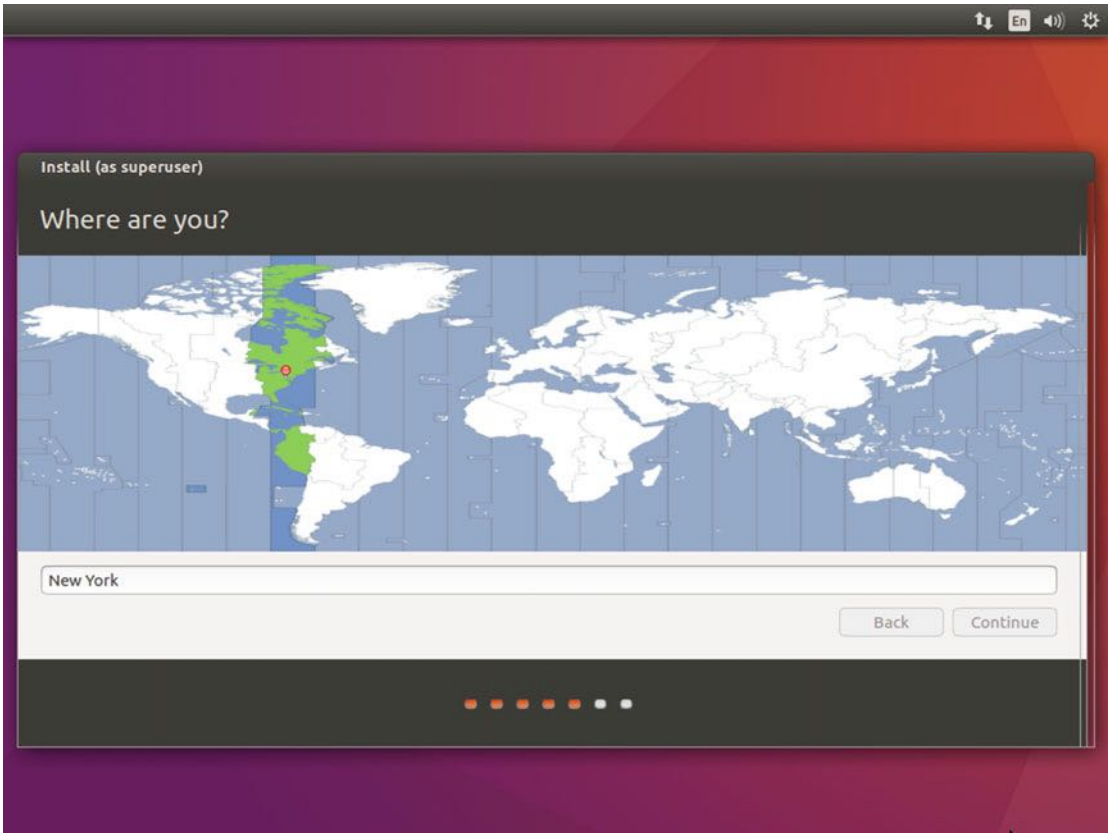


Figure 4-13. *Interactive time zone selection*

After selecting your time zone and pressing the Continue button, you must choose your keyboard layout and language (Figure 4-14). Usually this is automatically detected, too. Otherwise, you can push the button to try to detect it again interactively (it is going to ask you to press some keys and maybe ask some questions) and then test the results in the interactive text field. Once you have finished, you can press the Continue button.

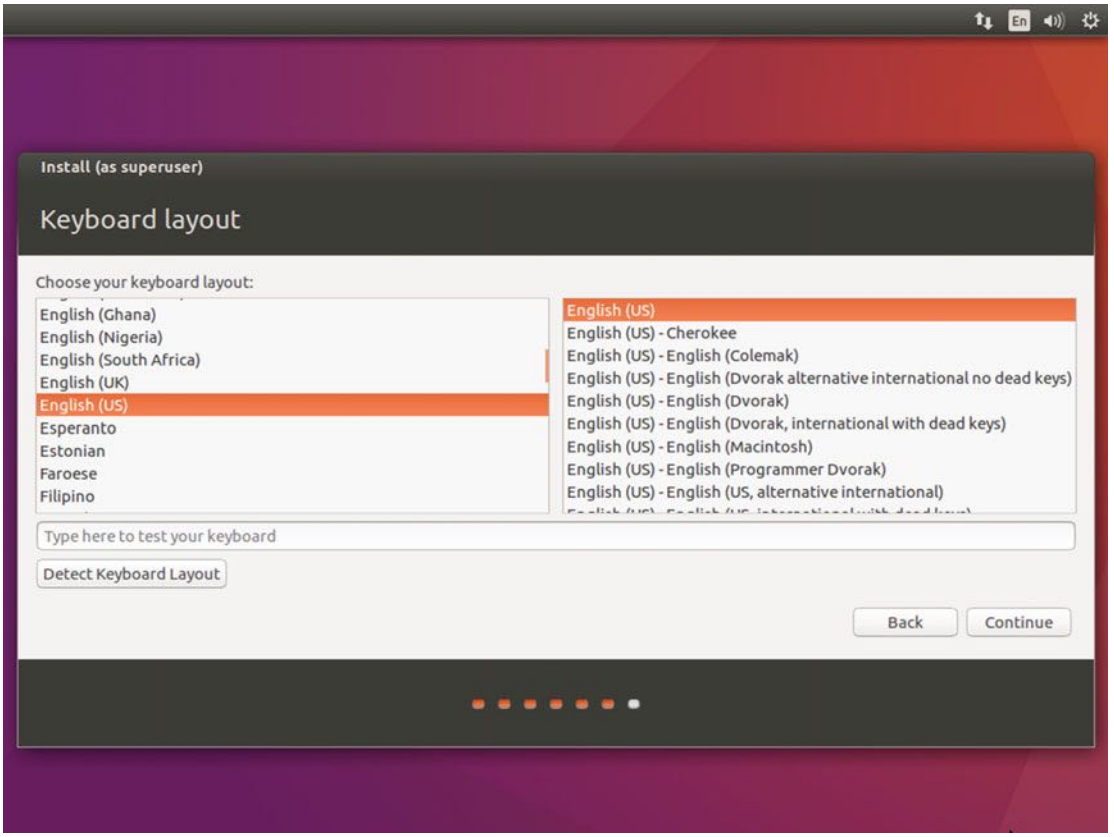


Figure 4-14. *Keyboard layout and language selection screen*

Figure 4-15 shows the last screen where you have to make any decisions. You basically have to decide your name and password. Actually, I recommend deciding these things BEFORE you begin your installation. I suggest you pick a username (put your real name, if you want, in the first text field) that is easy to type in any keyboard, so avoid any non-English alphanumeric characters. If your keyboard ever breaks or your installation fails and does not detect your keyboard layout, you are going to thank me for this, believe me.

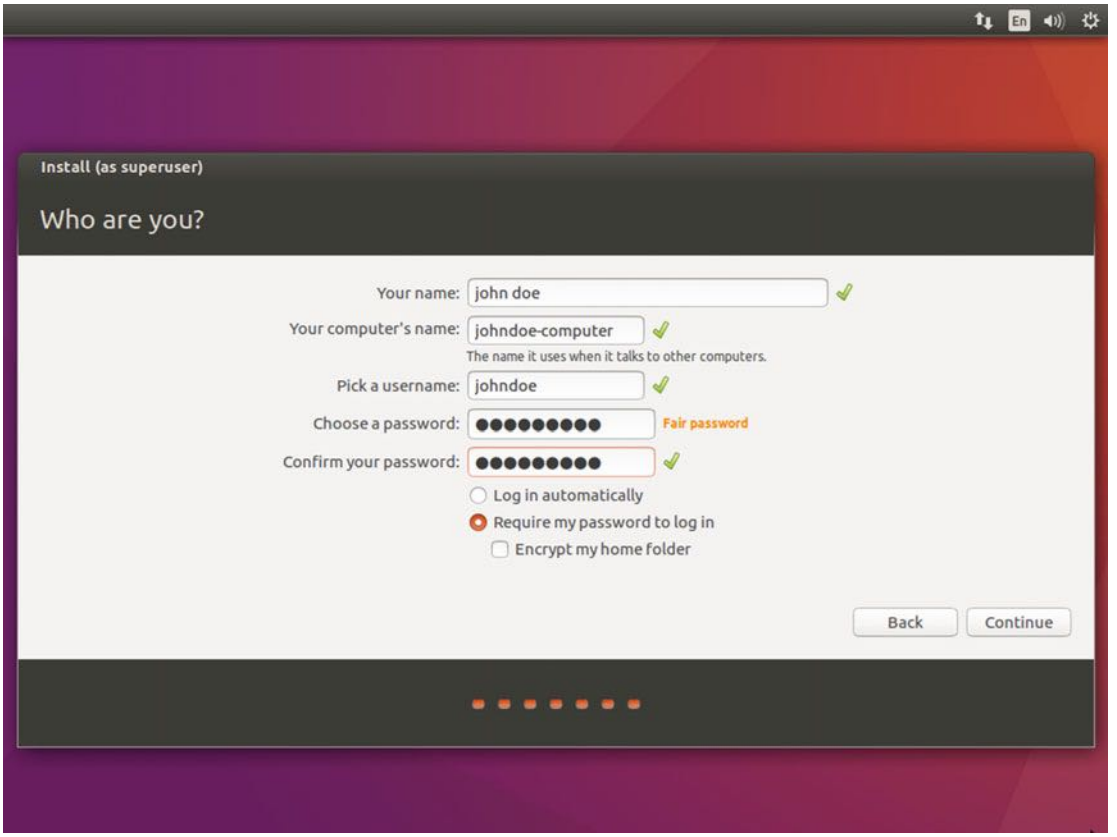


Figure 4-15. *Identity screen where you enter your user name and password*

Next you must choose a password. It's best to choose a strong one, and this means using lowercase and uppercase letters, numbers, and symbols. But there is a way to do this easily. I suggest the following: pick a sentence from a poem, a song, or a book that you like and you can remember easily. Then pick the first letter of each word, add a pair of numbers and symbols, and you have a very strong password that is easy to remember. For example, consider this famous sentence:

> *O Captain! my Captain! our fearful trip is done.*
>
> —Walt Whitman, Leaves of Grass, 1891

You can easily compose a very strong password by taking the first letter of each word, adding two different symbols to separate the different parts (make them easy to find on any keyboard), adding the first letter of the last name of the author and two numbers from the year when the poem was first published.

`Ocmcoftid.W-91`

Now you have to decide if you want to log in automatically when you turn on your computer or use your password. If you are installing Ubuntu on a desktop computer that only you can access now or in the future, you can log in automatically. Likewise, if this computer is for an elderly person who may not be able to remember a password or is open to the public, you can pick this option. Otherwise, I strongly suggest against this option.

As for encrypting the home folder, I give you the same advice as with the encryption of the hard disk. And again, remember, you must always do backups periodically. (I learned by experience that you can never recommend this too many times.)

Pressing the Continue button begins the real process of the installation. The screen is now a carousel that introduces you to the possibilities of your new OS at the same time that is doing the install process (see Figure 4-16). At the bottom is a progress bar and the step where you are at the moment (in text). The amount of time this process takes depends of your computer and your Internet connection speed because it is going to download several packages from the Internet.

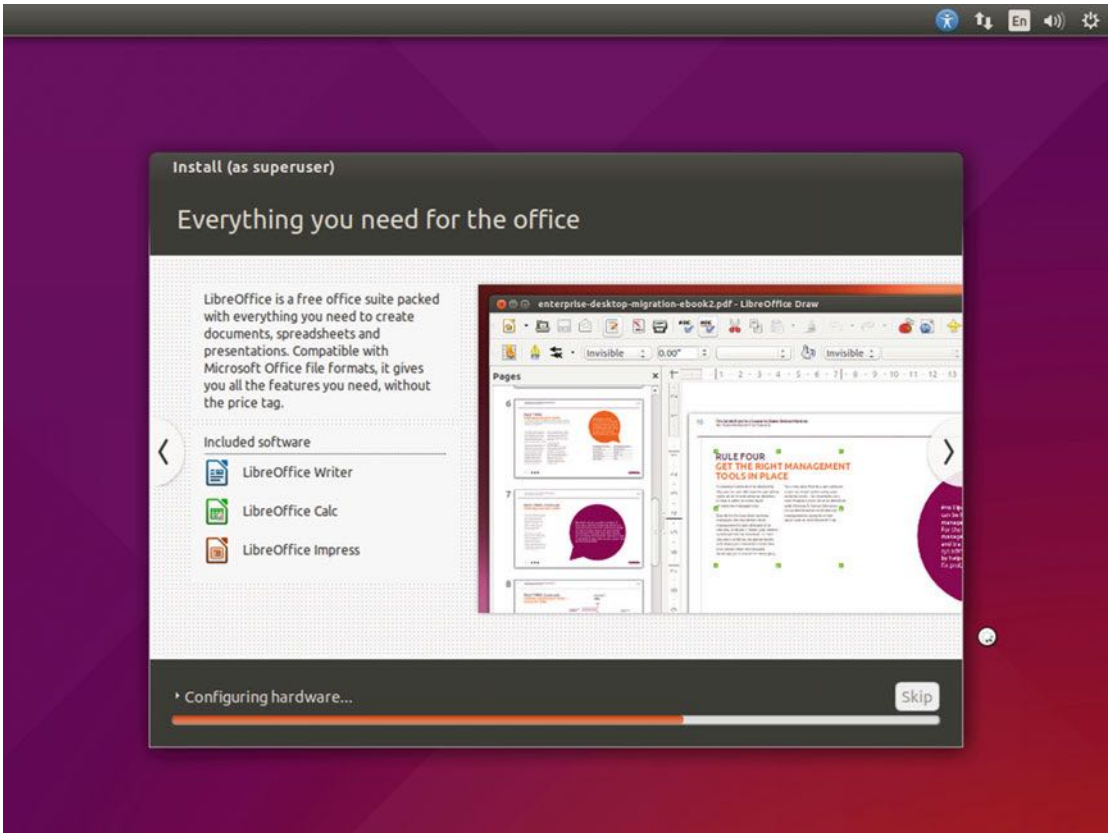


Figure 4-16. *Copying the files to the hard disk and finishing the setup*

At the end of the installation of Ubuntu on your hard disk, you will see a dialog that informs you that the installation is complete and you need to restart your computer to be able to enjoy your new Ubuntu Linux OS (see Figure 4-17).

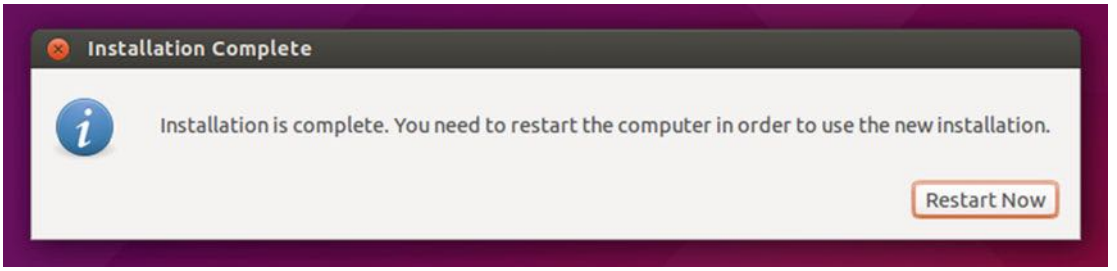


Figure 4-17. *The installation is complete. Restart your computer*

So restart your computer. After booting, the first screen that appears is the login one to start a new session (see Figure 4-18). Here you need to introduce your password (I hope that you did not forget it. You are going to need this password for every administration process). You could log in as a guest if you want to do a few things in the system (like browse the Internet or use any already installed app) but all of the data from this session will be temporary and you won't have access to your data on your hard drive. This is an ideal option, however, for any visitor to your system.

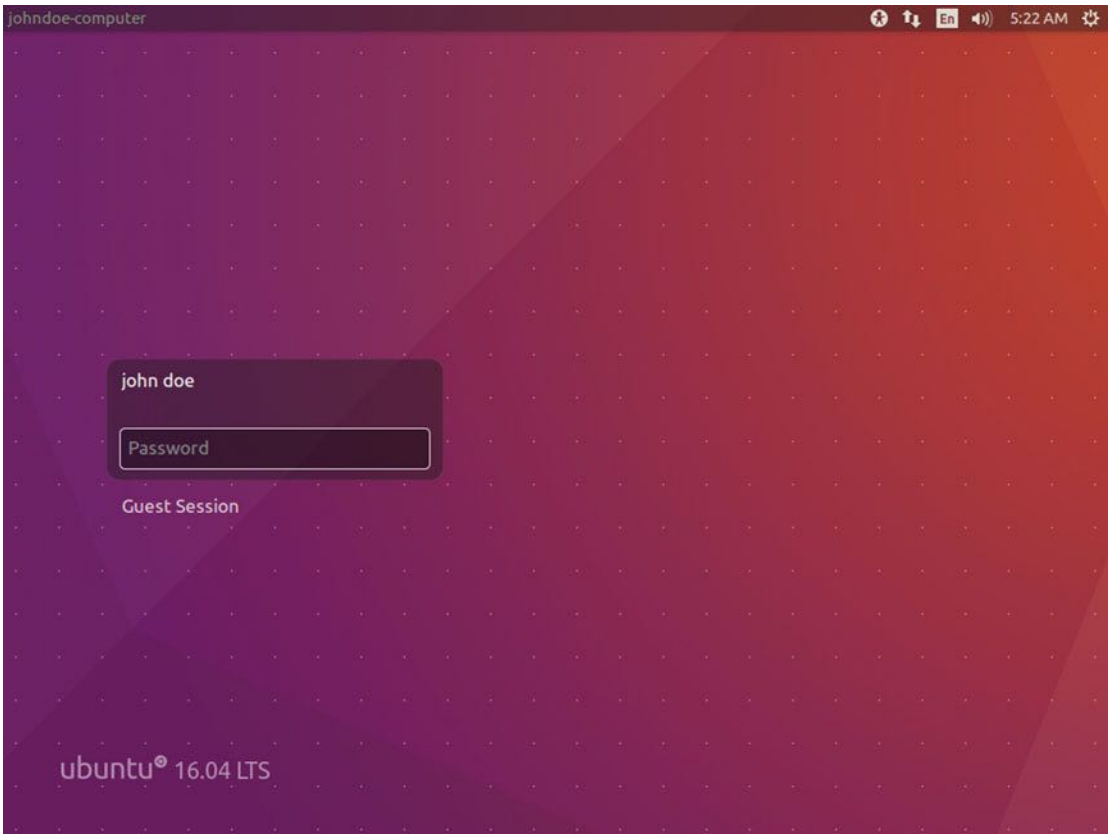


Figure 4-18. *The login screen to Ubuntu 16.04 LTS*

So enter your password. Now you can enjoy Ubuntu. The first screen you will see is a list of keyboard shortcuts (see Figure 4-19). This screen can always be shown by pressing and holding the “Super” key (usually the one with the Windows symbol). These shortcuts are very useful once you learn them. They will help you to increase your productivity, so do not ignore them.

Figure 4-19. *First screen after you successfully install Ubuntu*

As you can see, the install process is very easy. Don’t be swayed by those rumors that Linux can be very difficult to install and use. These days that is not true; you can install and use Linux in a very easy way, and Ubuntu is a great example. Of course, you can choose a more complicated or sophisticated one, and you can install the same Ubuntu from a text menu and in expert mode, but that is up to you and your knowledge and experience.

Maintenance

There are three essential tasks to maintain your Linux OS, and Ubuntu makes them very easy to do. As with the install process, I'm going to show you the easy way to do these things; I’ll cover the advanced options in other chapters. These three essential tasks are updating, installing/deleting apps, and upgrading.

Updating

Usually Ubuntu will notify you when new updates are available (see Figure 4-20) and you only have to follow the steps. But if you want to manually check if new updates are available and then install them, it is very easy; you only have to execute the Software Updater application (see Figure 4-21).

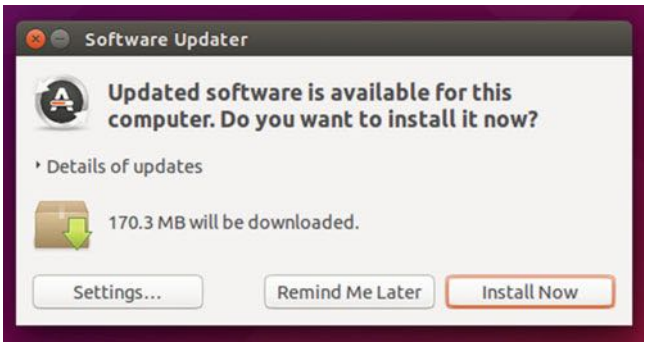


Figure 4-20. *Ubuntu updates notification dialog*

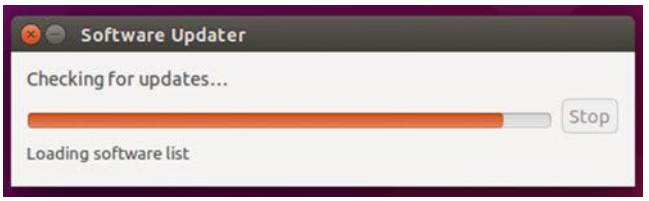


Figure 4-21. *The Software Updater application checking for updates*

Usually this process won't give you any problems, but sometimes bad things happen. If you frequently tweak your OS (by following tutorials from the Internet) or use PPAs to install programs, you will probably experience problems with this process sooner or later. It is very rare that any problem will appear if you only use the graphical apps of Ubuntu and follow the established way of doing things. Bad things happen when you play with fire. If you get burned, look for help in the available ways; this is usually enough to solve any problem. From time to time, some hardware driver updates, like the graphic cards, can give you some headaches, but this is rare and usually you can reinstall a previous version.

Some LTS versions have an .x suffix where x is a number, for example in the 14.04.3 LTS release. This suffix refers to the LTS enablement stack that is available at the moment. These enablement stacks are the way that Ubuntu chooses to bring the updates to the kernel and X windows systems in the LTS releases. This is included by default in the latest ISO images of a LTS release, but if you have one already installed and want to update it to that point, you have to do it manually through commands in the shell. You can get more information about this topic at `https://wiki.ubuntu.com/Kernel/LTSEnablementStack`.

Managing Apps

Sooner or later you are going to want to install new software or remove packages that you not use or like. To do so, Ubuntu gives you a very simple and complete app named Software, which is the official application management tool of Gnome (Figure 4-22). From it you can remove or add any application available in the official repositories, and there are many. You can also check for package updates and perform them from there.

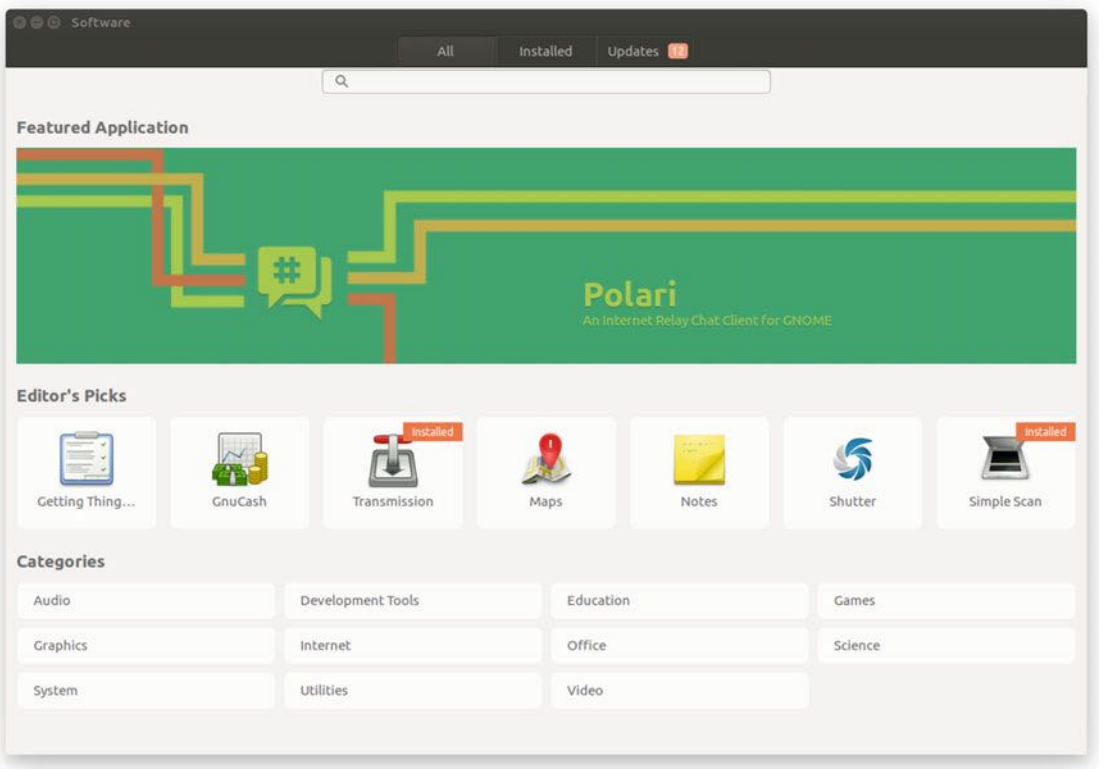


Figure 4-22. *The Software tool*

The applications are organized by categories (and subcategories). If you click over to one of them, you get a detailed resume of the app and sometimes even screenshots (see Figure 4-23). The installation/removal process is very simple and intuitive.

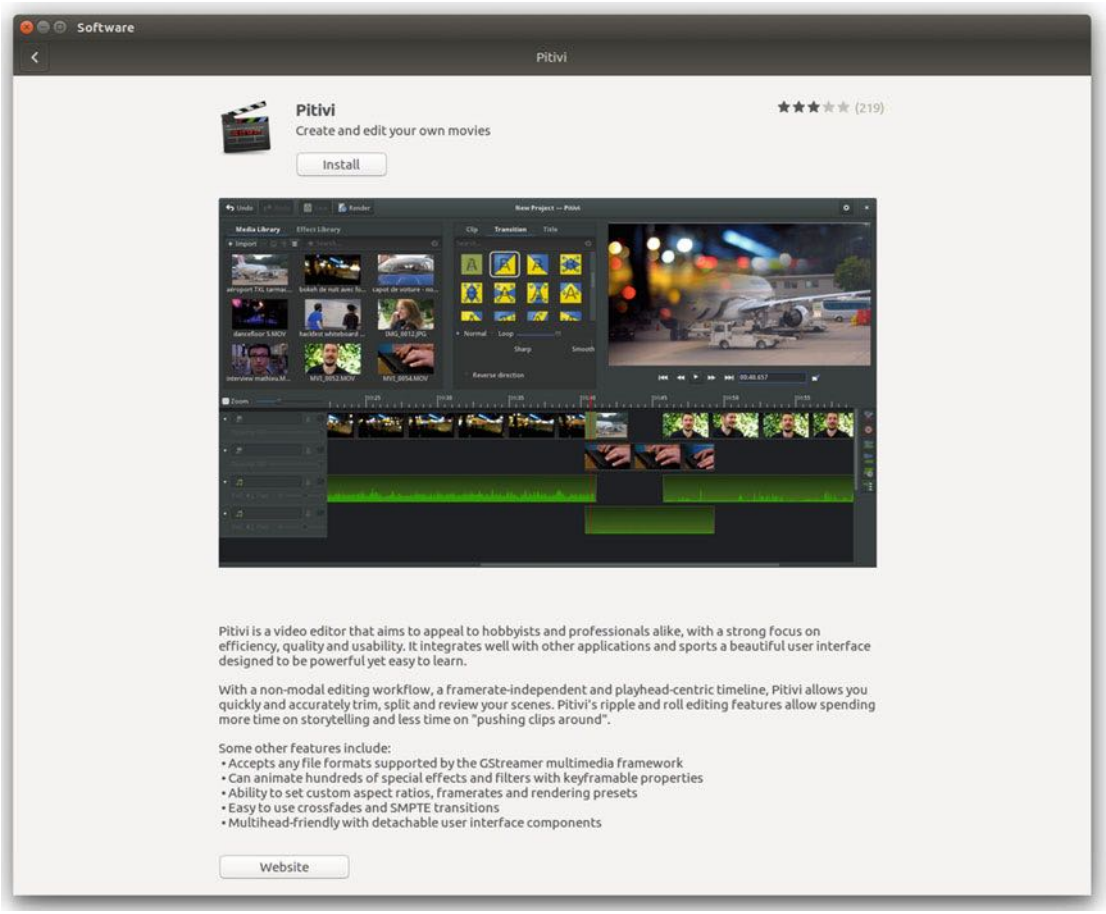


Figure 4-23. *The Software tool showing information about an application*

Upgrading

As in the updating case, you will usually be notified when a new release is available. However, you can select the "Software & Updates ➤ Updates ➤ Notify me of a new Ubuntu version" option of the System Settings tool if you want to be notified only of a new LTS release (the default) or when a normal release is available. The application used is the same as in the update case; you can also invoke it by hand. The difference here is the duration of the task and this is more critical; you can easily end up with an unusable system if something goes wrong during this process. If you tweak your system a lot and use external PPAs, very often the upgrading process will fail at some point and you'll end up with a half-upgraded and normally non-bootable OS. Enthusiasts who like to do those things with their Linux systems normally make a fresh install of the new releases by preserving only their data (or restoring it from a backup). I recommend you always make a backup of all of your data before you upgrade your distribution to a new release.

Pros and Cons

I'm going to list some of the things that I personally see as pros and cons of the Ubuntu distribution. Of course there is always room for discussion in this matter, but I'll do my best to be as objective as possible.

Pros

- Ubuntu is one of the easiest to use.
- It is without a doubt the most used and installed of all of the Linux distros.
- It has great official support from the distro (and others) and a very big community willing to help you.
- It has great hardware support. And if a company only supports a few distros, Ubuntu or Debian will probably among them.
- As for applications, if they only support a few package managers, probably dpkg is going to be included.
- It offers a regular release scheme. You always know when the next release is going to be launched, which is an advantage when scheduling maintenance. LTS releases offer extra quality and stability.
- You can use Ubuntu on a smartphone or a tablet with almost the same experience as on your laptop or desktop.

Cons

- Not everybody likes Unity. But you can always choose another flavor of Ubuntu.
- Canonical has a particular vision of Linux. The company has made things its way, and certain people do not like that.
- Sometimes Canonical introduces changes in a normal release that are immature or not well thought out. To avoid this, you can use a LTS release, but you lose the new innovations.
- If you prefer a totally free Linux distro with only free software, Ubuntu is not your distro (but there is an option in the text menu to only install free software).
- Sometimes the commercial aspects of the distribution overrule the community aspects, like the integration with Amazon.
- The big goal of creating a Linux distro available on all type of devices meant a loss of focus on the desktop and other areas. Instead of having the best distro for the desktop, they could have ended with a regular one on all of the scenarios.
- Ubuntu makes big use of the Debian distro work but it does not give back much; the same with the Linux kernel.

Summary

I analyzed in detail here the best-known Linux distribution of all, Ubuntu, from its origins and philosophy to the installation and maintenance. The section on pros and cons and the decision criteria give you a series of arguments to compare this distro to others in a more objective way.

In the next chapter, I do the same with Fedora.

CHAPTER 5

Fedora

Fedora is always in the top five of the most popular Linux distributions, and part of that merit comes from its "big brother," Red Hat, which not only uses Fedora as the community edition of its distro but also uses it as a prototype for development. But it's unfair to ignore the merits of Fedora itself. It has a large and expert community. It is not a favorite of newcomers to Linux, but many experienced Linux users, developers, and systems administrators use it as their day-bay-day default distro. And because of Red Hat, although it has lately been losing traction in favor of easier to use or trendy distros, it still has great future.

History

The Fedora Project was founded on September 22, 2003 when Red Hat decided to split Red Hat Linux into Red Hat Enterprise Linux (RHEL) and another OS based on a community, Fedora. At the same time, another distro, the Red Hat Professional Workstation, was created with the intention of filling the place RHL had once occupied but without a clear route sheet. Non-enterprise RHL users soon switched to Fedora instead. The first release of the Fedora distribution appeared on November 6 of the same year.

The Fedora name came from a former volunteer project, Fedora Linux, which made additional software; it's inspired by the hat worn by the character of the Red Hat logo (shadowman).

Philosophy

The philosophy of Fedora is a mix between two goals: to serve as the community edition of Red Hat Enterprise Linux and to serve as a sort of laboratory for new technologies that might later be integrated into Red Hat. In fact, the new releases of Red Hat come from a stabilized, secured, and improved version of a release of Fedora (they also offer technologies and software exclusive to Red Hat). As a result, Fedora focuses on innovation and on contributing and collaborating with the Linux kernel itself. Fedora aims to be the leader of innovation and the distro that creates the path that Linux takes. Thanks to Fedora, the new init system, systemd, was widely adopted by almost every distro. Linus Torvalds uses Fedora on his computers, probably because of the tight relationship between the Fedora community and developers with the Kernel communities and core developers.

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_5

Distro Selection Criteria

Now that you know a brief history of Fedora, let's see how it fares on the selection criteria from Chapter 2.

Purpose and Environment

Because it has the support of Red Hat, Fedora, even though it is a general purpose distro, has versions for different environments and for task-oriented purposes. Along with Ubuntu, Fedora offers the most versions/flavors available with official support.

The following are the main official versions of Fedora:

- **Workstation**: The general purpose version oriented to the desktop/workstation. This is the one that I will show you here.
- **Server**: For installing on servers, it's a task-oriented version.
- **Cloud**: To use in cloud environments, it's a specialized and minimal server version.

Like Ubuntu and its different flavors (which really only differ in their desktop environment), Fedora offers Fedora Spins, which are versions of Fedora with a DE different from the default one. You'll see them in the "Desktop Environment" section.

Fedora also offers another series of images that are made with a specific task in mind, the Fedora Labs (`https://labs.fedoraproject.org`). Thus, Fedora is a task-oriented distro in the following scenarios:

- **Design Suite**: Oriented to publishing, multimedia, and visual design.
- **Games**: A collection of games ready to run under Fedora.
- **Jam**: To create, edit, and produce music and audio.
- **Robotics Suite**: Packages aimed at beginners and experts in robotics.
- **Scientific**: Tools used in scientific research and numerical computing.
- **Security Lab**: For security auditing, system rescue, and forensics.

And finally, Fedora has special images for the ARM architecture, for servers and desktops. For desktops, there is the official version for that processor, and all of the Fedora Spins are available as well. For servers, there is a regular one and a minimal one (which you can think of as the Cloud equivalent, a core version). You can get all of them at `https://arm.fedoraproject.org`.

Support

Fedora is reasonably well-supported distro (not as well supported as Ubuntu, but well enough). This support comes only from the community and Fedora developers; there is no commercial support for this distro. The Red Hat company sponsors Fedora but only offers commercial support for its commercial distribution, Red Hat.

Also, the Fedora community is big, but it's not as big as the Ubuntu one. However, it does not lack ways of supporting its users. The following are the channels to get help from the community:

- **Documentation**: `http://docs.fedoraproject.org/`
- **FAQs**: `https://fedoraproject.org/wiki/FAQ`
- **Wiki**: `https://fedoraproject.org/wiki`

- **Ask Fedora**: https://ask.fedoraproject.org/en/questions/
- **Fedora Forum**: http://fedoraforum.org/
- **Community**: http://fedoracommunity.org/
- **Mailing List**: https://lists.fedoraproject.org/mailman/listinfo/users
- **IRC**: #fedora channel at freenode

User Friendliness

From the point of view of an Ubuntu user, you could say that Fedora is less user friendly, but from the point of view of an Arch Linux/Gentoo user, it is a very easy-to-use distro. Depending on what you want to do with it, Fedora occupies a middle point. For example, Fedora has a very good installation tool, Anaconda, which makes that process a comfortable experience. But when you need to do any administration/maintenance tasks, you need to know how to use command line tools. This was the norm for years in Linux, and is probably the same case as Debian, for example, but the truth is that there are other distros that are easier to use than Fedora.

Stability

Although Fedora is always on the bleeding edge of technology and innovation, it is a very stable distribution. From my point of view, it is more stable than Ubuntu, if you do things in the manner intended (in other words, don't use external repos, don't tweak it too much, etc.). Fedora offers a shorter period of support for its releases than Ubuntu (13 months versus 18 months for regular releases of Ubuntu). As with Ubuntu, Fedora uses a variation of the standard release scheme; both have the same cycle for new releases; 6 months. You can learn more about the Fedora release cycle at https://fedoraproject.org/wiki/Releases.

Hardware Support

As a result of its policy of not including private drivers, the hardware support in Fedora is not as good as in Ubuntu. However, it gets the benefits from being the community version of Red Hat, because a lot of hardware supports Red Hat. Also, a great number of companies develop drivers in the `.rpm` package format, which is supported in Fedora. And if you are willing to use alternative, non-official repositories for the distro, you can use various private drivers (especially graphics and network cards).

Note that using the most recent kernels helps detect new hardware supported by it. This is especially important with laptops.

Aesthetics

Apart from the logo, colors, and backgrounds, Fedora does not make a great effort to customize the design of its distro. It ships the standard and current versions of Gnome without customizations, which is the opposite of Ubuntu, Mint, and others. If you like the out-of-the-box designs of the current DEs, you will like Fedora; otherwise you should look for customizations.

Desktop Environment

The default and official desktop environment of Fedora is Gnome, currently version 3 of this DE. But as mentioned, the Fedora Spins provide alternatives to this in the form of available ISO images that use other DEs as defaults. Currently, the available Fedora Spins (`https://spins.fedoraproject.org`) are the following:

- **KDE Plasma**: Uses the KDE Plasma DE.
- **XFCE**: The XFCE DE is used here.
- **LXDE**: The lightweight LXDE desktop for this option.
- **MATE-COMPIZ**: Combines MATE (sort of the classical Gnome 2) with Compiz composition effects.
- **CINNAMON**: Uses the GTK3+ toolkit from Gnome 3, but with a more classical look.
- **SOAS**: Sugar on a Stick combines the Sugar Learning platform with the mobility of being able to fit on a USB drive. It is oriented as an education/learning platform.

Init System

If the majority of the Linux distros have recently adopted the systemd init system, obviously the distro that developed it (from Red Hat developers) and first adopted it continues to use it.

Package Management System

Fedora uses the same package management system as Red Hat, rpm, and the same tools for it. Although there is a GUI app to manage the software apps, named Software (gnome-software), the proper way to manage packages in Fedora is via the `dnf` command (formerly `yum`) in the terminal. Also there are tools to manage the `.rpm` packages directly.

The official repository (there is only one) of Fedora has an inferior number of packages compared to Debian or Ubuntu (about 5,000 packages less), and third party packages that are non-free are also absent from it. Thus, there is a tradition of external, non-official repositories that provide more packages and non-free ones. As I said in the Ubuntu chapter, this practice compromises not only the stability of the OS, but it's also a clear security risk because you must trust the repository maintainers.

Fedora benefits from the `.rpm` format from all of those companies and developers who create software that maybe not be in any repository, but is usually available in `.rpm` or `.deb` packages (or as source tarballs).

Architecture

As with many distros, the two main Intel architectures are supported, the 32-bit version and the 64-bit version. Fedora recently announced that they are going to focus only on the 64-bit version in future releases, leaving the i686 architecture on a second plane and perhaps not synced in the release cycle. ARM is also supported.

Security/Anonymity

Fedora has a well-deserved reputation of being one of the most secure distributions. For example, a firewall, PolicyKit, and SELinux are enabled by default, among other security features. Also, you can use encryption on all of your disk or by volume or file; Fedora is the leader in this field. See `https://fedoraproject.org/wiki/Security_Features` for more information.

On the anonymity side, you can configure those options when you first start the Gnome Initial Setup or later in the Settings app. You can decide what information is shared and what information is sent back to Fedora or Gnome.

Principles and Ethics

Fedora has a strong focus on only using free and open source software, but that does not mean that they don't distribute private blobs (non-free binary firmware) in the kernel. Other than that, Fedora is very strict with its policy of not using proprietary software of drivers. It doesn't include the proprietary drivers of graphics cards, mp3 software, the Flash player, etc. Users who want to use some of this software normally must rely on alternative, non-official repositories. The official Fedora policy about this matter can be seen on its wiki at https://fedoraproject.org/wiki/Forbidden_items.

Live CD

The ISO images of Fedora work well as Live editions of the distribution, so you can use them to start a Live session of Fedora without touching anything on your storage devices.

Professional Certification

Fedora itself has no professional certifications, but its supporting company, Red Hat, has several of the most prestigious ones, as outlined in Chapter 2. Thus, because Fedora is the base for Red Hat, you should be able to apply the knowledge gained by passing those certification exams to Fedora without a problem.

Installation

As stated in the previous chapter, I will deviate in this installation (and in the next ones) a bit from the "predefined" way that is given as the default workflow by the installation tool. As you will see later, this little change in Fedora doesn't make a great difference, but it does in other distros (e.g. Ubuntu).

The first step is always to download the ISO image of the distro to be able to install it. Obviously I chose the Workstation version of the distro to install here, because it is the one used by the majority of users. So go to the download page of the Fedora distribution, https://getfedora.org/en/workstation/download/, and you will something similar to Figure 5-1.

Figure 5-1. *The download page of Fedora 23 Workstation*

If you click the Download button, the download begins automatically. The release at the time of writing is 23 and the default architecture is 64 bits. The current size of the image, 1.4GB, only fits on a DVD or a USB drive. If you need a different architecture, image, or flavor, you can go to the bottom of the page where you will see something similar to Figure 5-2.

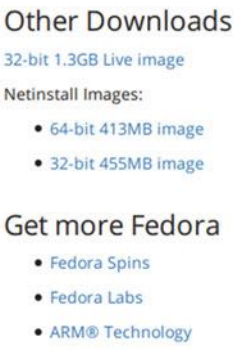


Figure 5-2. *Alternatives for downloading Fedora's ISO images*

You can get the 32-bit ISO image (i686) or the netinstall images for both architectures. You can also access the Spin and Labs versions and the ARM versions. This page also links to the Fedora Wiki to locate online or local vendors that sell DVD copies of the distribution or to their Free Media Request program for those who can't afford the purchase. Another alternative is to use a BitTorrent tracker to download the ISO image. These options are at `https://torrents.fedoraproject.org/`.

Fedora provides access to the release notes (I recommend always reading these notes when upgrading to a new version) and the documentation for the installation process for that release, which is very good (`http://docs.fedoraproject.org/en-US/Fedora/23/html/Installation_Guide/`). However, you only need this guide for the very first steps of the installation process; for the rest of the steps, good documentation is available directly from the installation program.

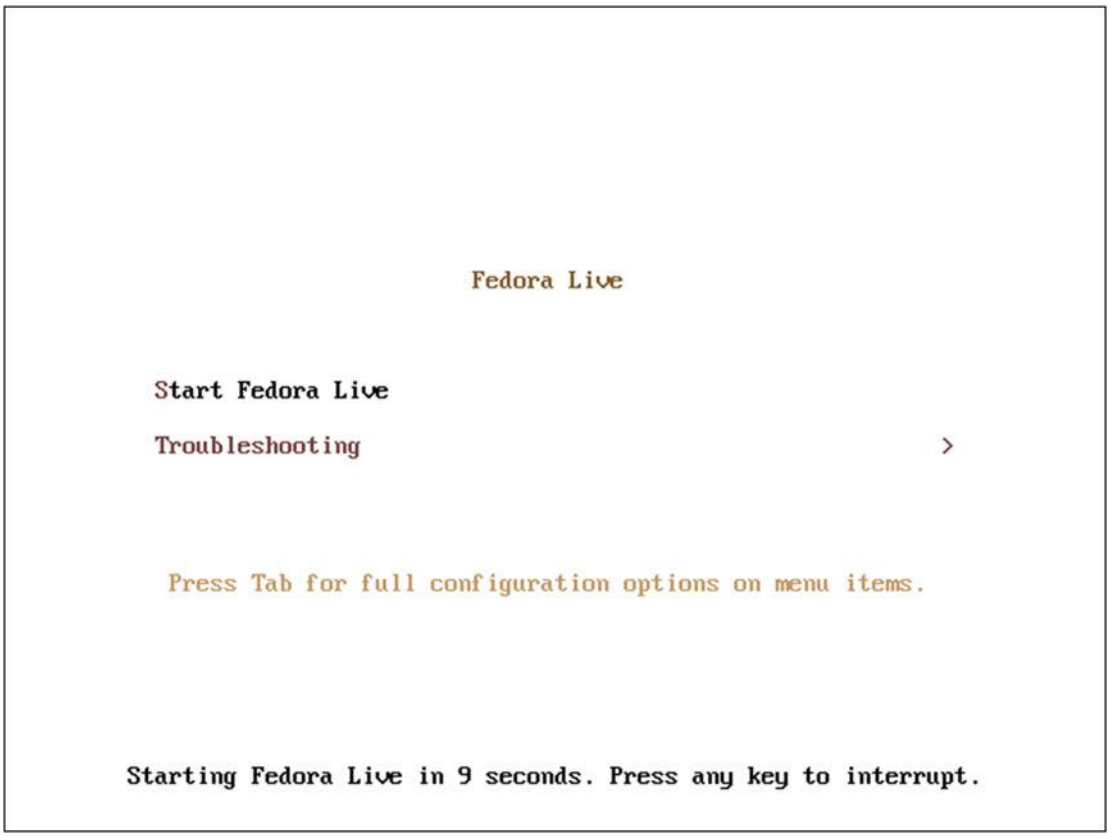


Figure 5-3. *First screen of the Fedora installation process*

Once you have the ISO image of the distro, you can boot up your system with it for the first time. The first screen that appears is the one in the Figure 5-3; it is a classical text menu (which is common in Linux). You may think that it's less polished that the Ubuntu one, but don't let its appearance trick you. This screen only lasts 10 seconds before it starts Fedora Live automatically unless you press any key. Thus, from here you can do four meaningful things:

- Nothing. The Live version of Fedora will start after 10 seconds.
- Start Fedora Live immediately by pressing Enter.
- Press Tab to access to the configuration options, as you can see in Figure 5-4. This is for advanced users who usually customize the installation because of hardware or deployment requirements.

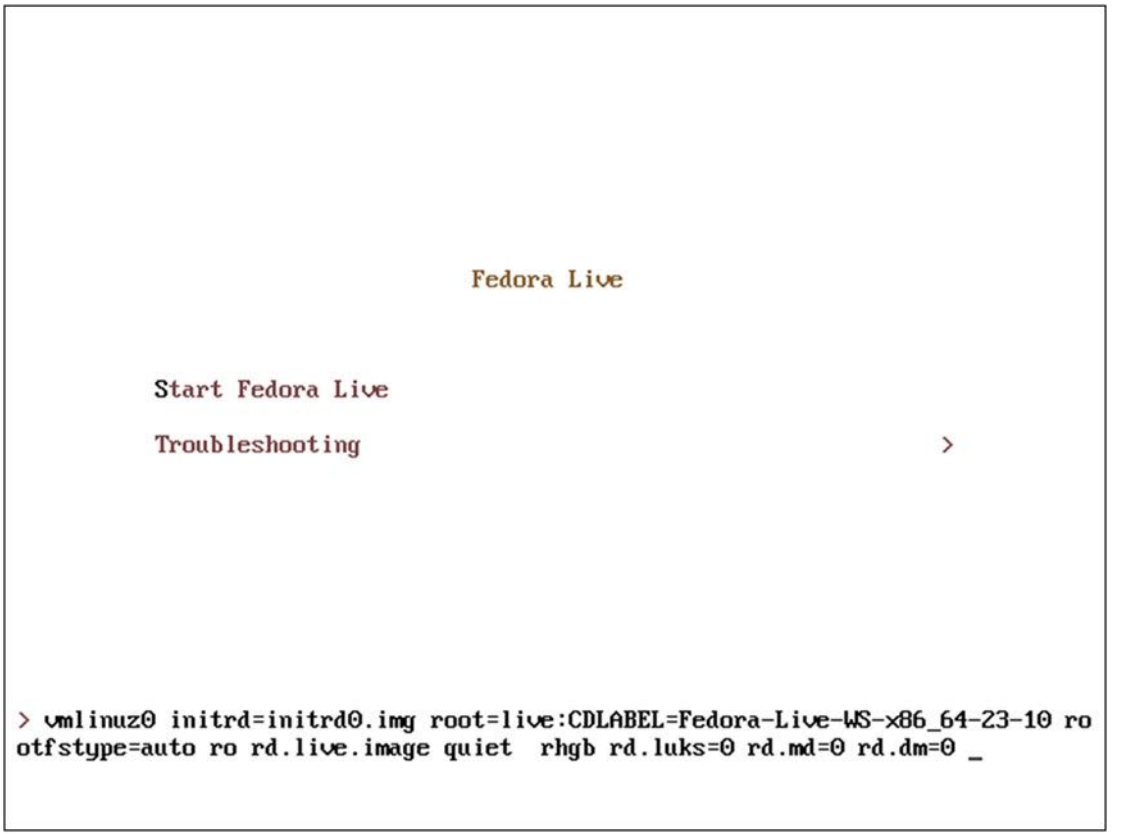


Figure 5-4. *The advanced configuration options of the installation boot*

- Go to the other menu option, Troubleshooting (Figure 5-5), which allows you to test your installation media or memory, or both, from your hard drive. It also allows you to start the Live version, avoiding problems with your graphics card (which is common in laptops). Do this when you experience rare things on your screen (like distortions or lines) with the default option.

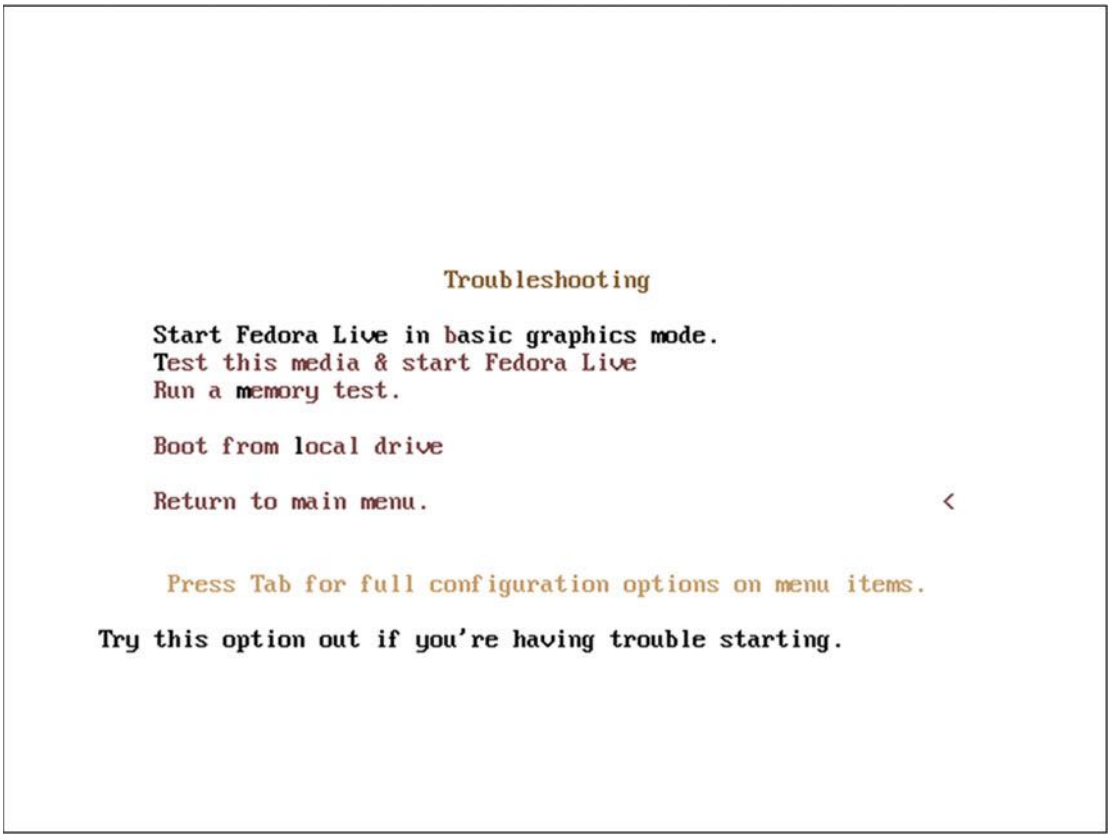


Figure 5-5. *The troubleshooting menu of the Fedora Live boot process*

The obvious choice is to continue with the default option, then wait 10 seconds or press Enter to go to the next step. The next screen that is going to appear in front of you is a black one with a sort of animated progress bar at the bottom and the Fedora release version at the end of this step, as you can see in Figure 5-6.

Figure 5-6. *The bottom of the screen when Fedora Live is booting up*

When Fedora Live is finally up, the welcome screen that appears (Figure 5-7) lets you choose between two options: installing the distribution on the hard drive of your system or keep testing Fedora in the Live session. If you are new to the Gnome environment or you want to test Fedora, you can always choose to install it later through the Activities menu. Let's choose the "Install to Hard Drive" option to keep going.

Figure 5-7. *Fedora Workstation Live welcome screen*

This is the first real step of the installation process, and the first thing to do is to choose the language to use in all of the next steps (and by default in the distro). If you have an Internet connection, it will try to auto-detect your region (time zone, to be exact) through your IP WAN address and set the language automatically for you. Otherwise, it will pick English (USA) by default and you must change it manually if necessary. There is also a button named Help!, which is a context help that can guide you in the process. After you pick your language, press the "Continue" button (Figure 5-8).

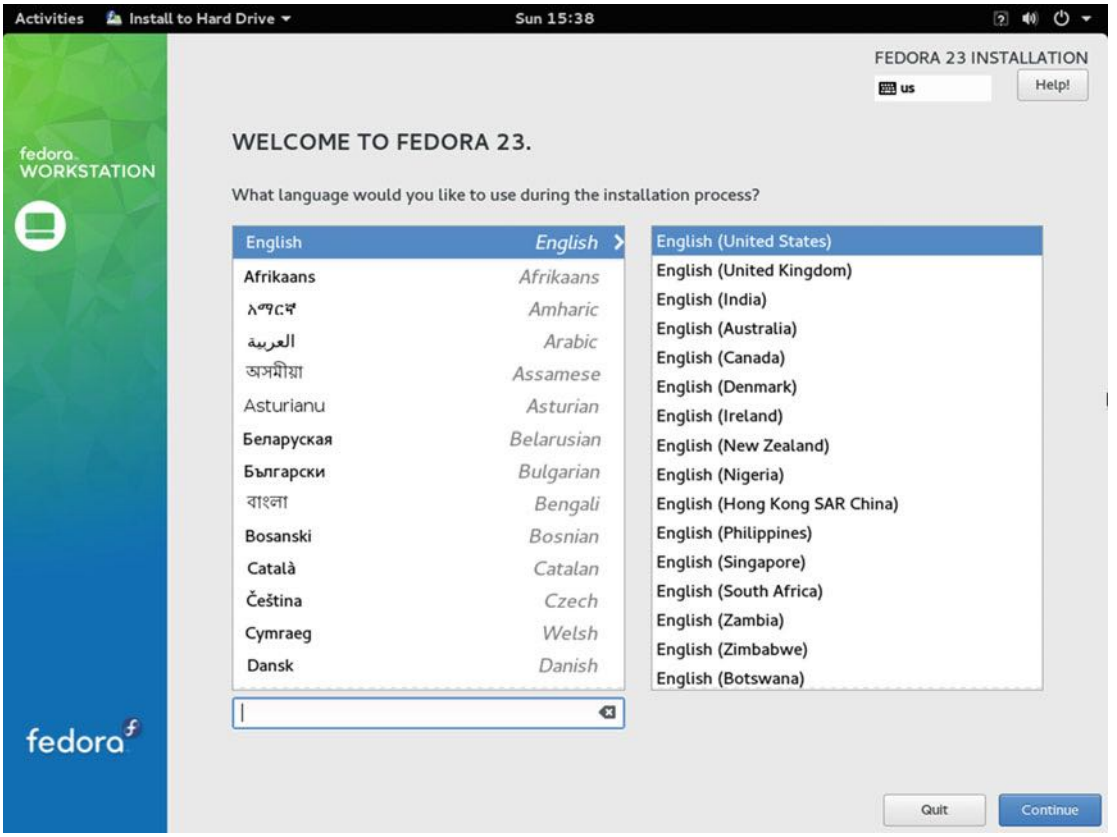


Figure 5-8. *The language selection screen*

The next screen includes almost all of the rest of the choices that you must make to complete the process. As in the previous screen, Anaconda (the Fedora installation program), is going to try to make all the decisions for you, and in a big number of cases they will be the right ones. However, even if all the choices are right, as in Figure 5-9, the program won't allow you to continue until you complete the Installation Destination section. The reason is obvious: before you make any changes on your hard disk, you want to check them out. Let's take a tour through all of the four sections included here. If you press the Help button, you'll get the help dialog shown in Figure 5-10.

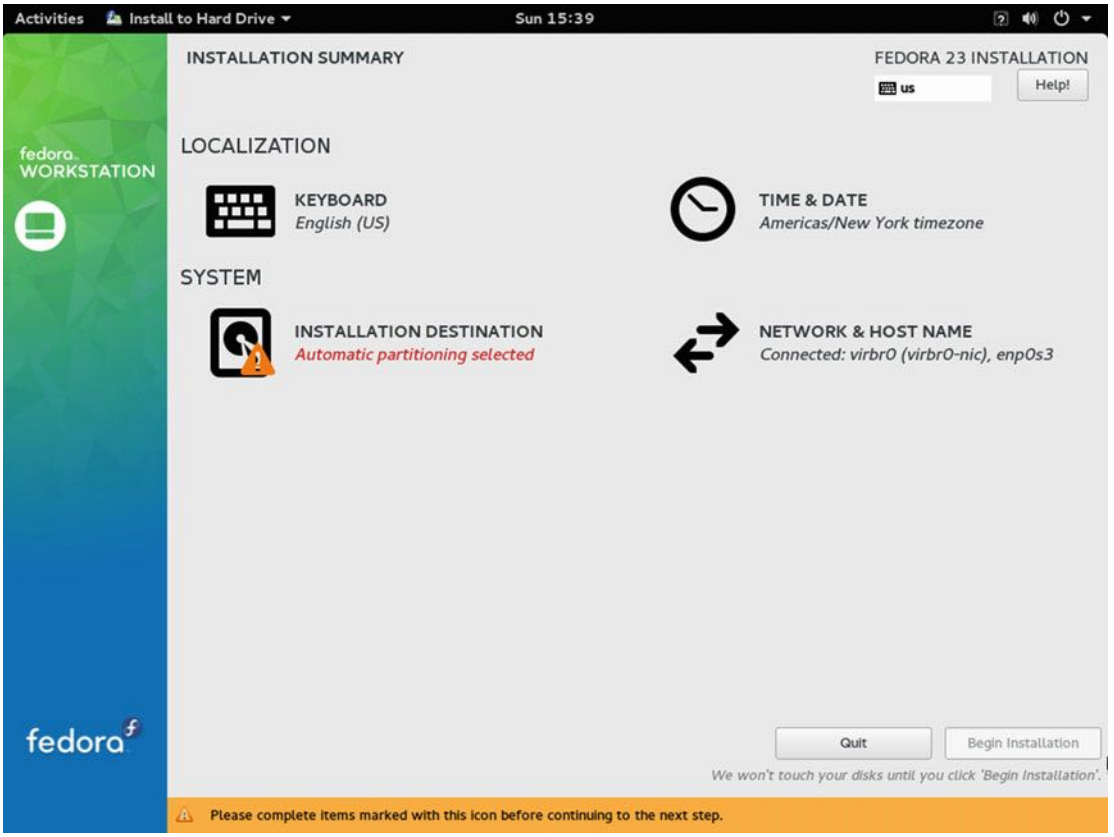


Figure 5-9. *The installation summary screen*

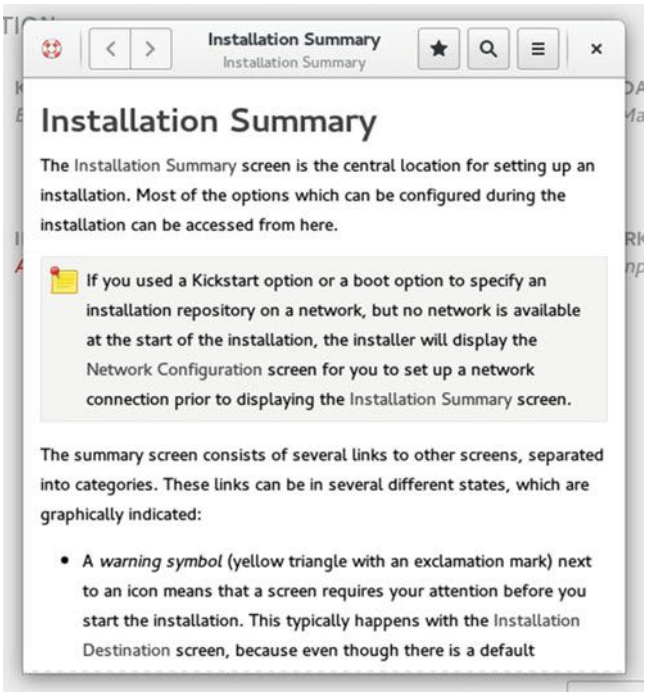


Figure 5-10. *The context help that can guide you through all of the installation*

In the Keyboard section (Figure 5-11) you can choose the keyboard layout(s) that you want to use in your distro. As usual, there is a big probability that the correct one was chosen automatically; if not, you must choose it manually. There are two interesting options here. First, if you select with your mouse any layout from the list, you will see an interactive representation of it (to help you identify it visually). Second, if you added more than one layout, you can pick from a list the keyboard combination that you want to use to switch between them. To do the latter option, you must use the Options button.

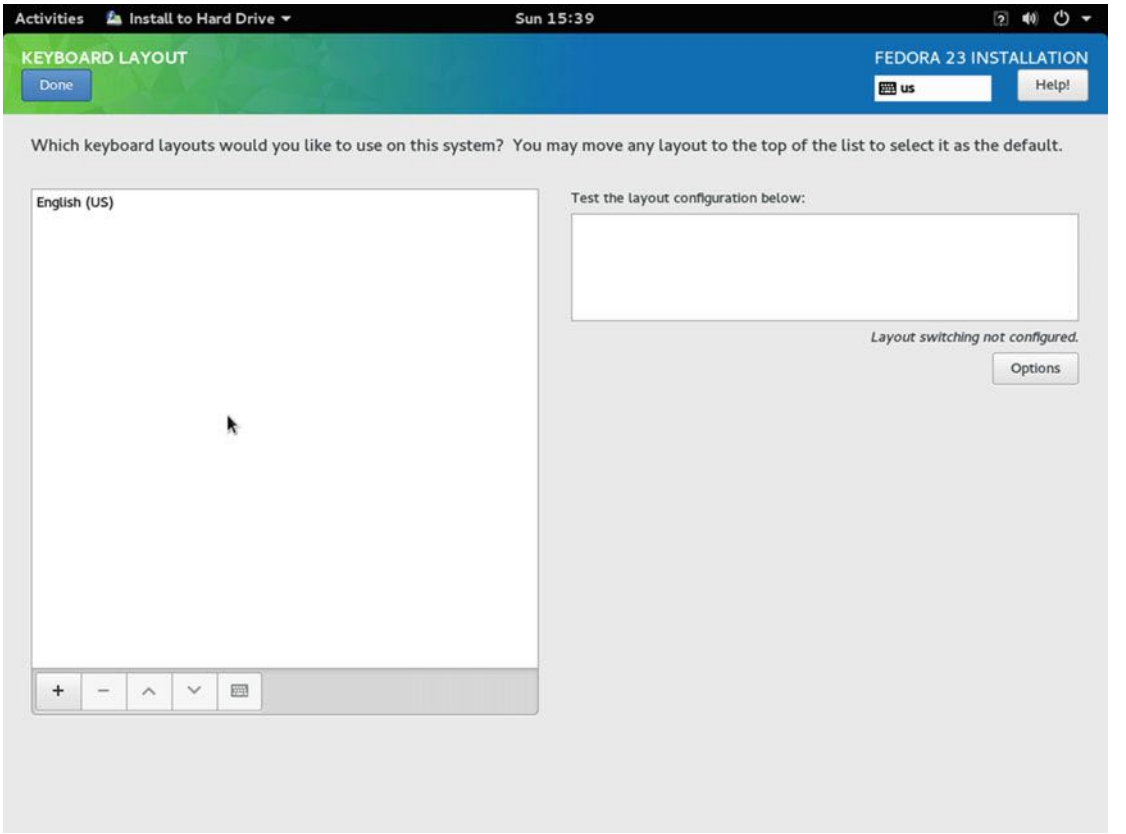


Figure 5-11. *The keyboard layout selection screen*

The Time & Date section allows you to select your time zone interactively on a map and set both time and date manually (Figure 5-12). If you are online, this is probably automatically selected correctly. One of the differences from Ubuntu is that, apart from being able to manually adjust the clock and calendar, you can also toggle the network time and even configure it. The NTP (Network Time Protocol) means that your system clock will always be correct because it's adjusted regularly by checking a time server on the Internet. You can also set your NTP servers via the gear icon if you want a different one from the Fedora default. This is turned on by default in Ubuntu, but you can see that Fedora has a different approach to the matter. As part of the Red Hat legacy and also as part of its different philosophy, you will continue to see how Fedora, although it is easy to use and install, gives the user more responsibility. I recommend you stick with the default NTP option. Make the necessary adjustments or simply check that all is correct, and then press the Done button.

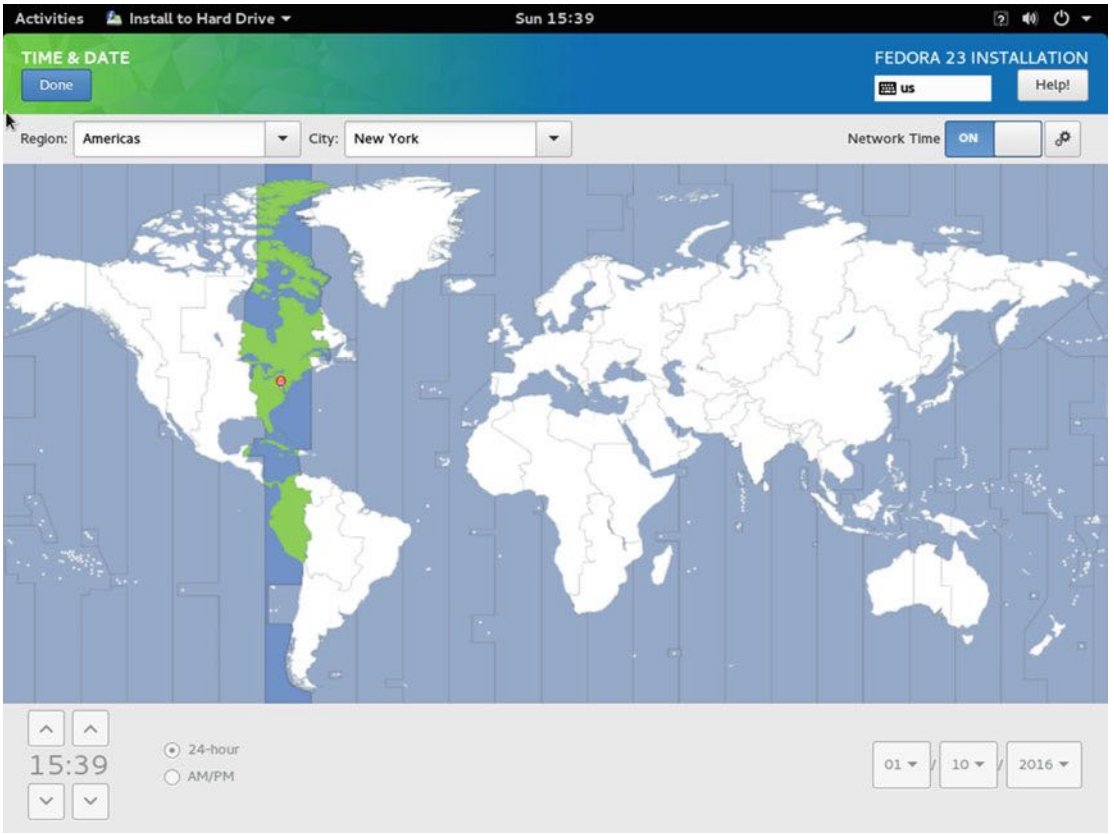


Figure 5-12. *Selection of time and date by region*

The Network & Host Name section (Figure 5-13) is a little tricky since you can only change the host name here. The rest of the network configuration is under the configuration options in Gnome. If you are not on a local network or you don't understand the option, keep the host name value as the default one. The rest of the configuration is accessible via the icons to the right of the top Gnome menu bar, where you can see the volume, net, and power icons. If you click this icon, you will see a dialog like the second one in Figure 5-14. In this dialog, you will see the current network configuration detected by Anaconda; it can be a wired or wireless connection. You'll also see the user and volume settings and the icons to access the configuration, screen lock, and power settings. If you click the network connection icon, it will show you two options: to toggle the connection on/off and to access the settings. Click the settings and you access a pop-up dialog with all the network settings, as shown at the bottom of Figure 5-14.

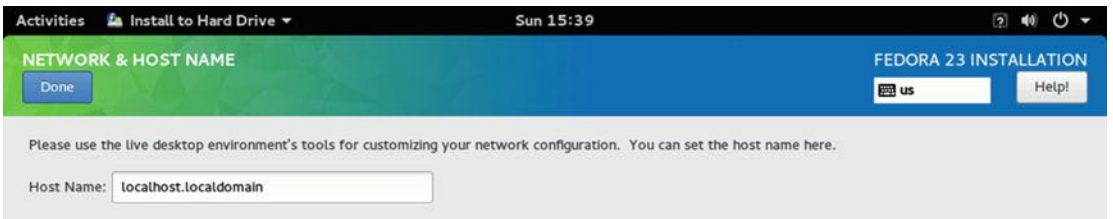


Figure 5-13. *The first part of the network configuration*

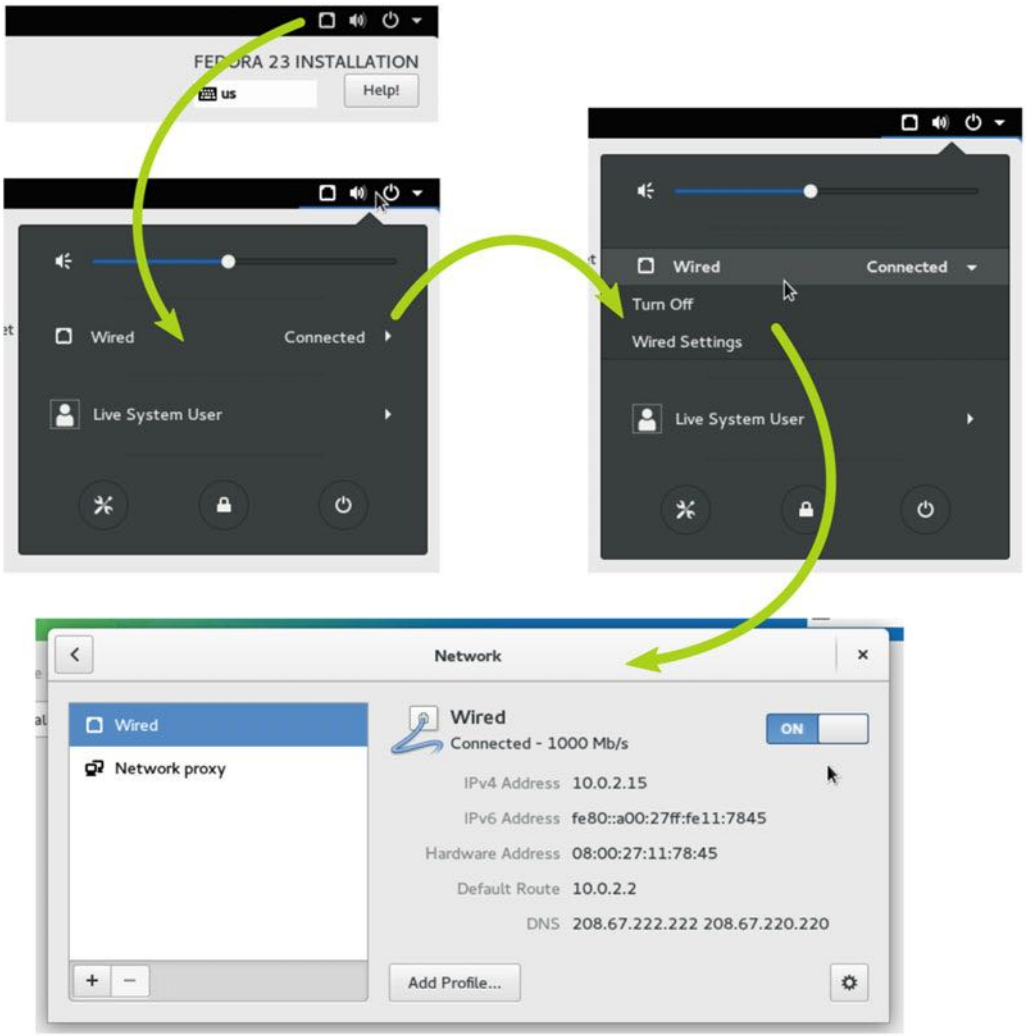


Figure 5-14. *How to access the network settings in the installation process*

On the left is a list of all the available connections detected (you can add more) and the proxy configuration. On the right is the information about the current connection selected and a button to toggle it on/off. If you are connected to the Internet with a router (wired or Wi-Fi) with a DNS server or via a local network, all of this will be correctly configured automatically. If not or if you want to tweak something, you can always change the settings via the gear icon. Also, you can manage different profiles via the Add Profile button, which is very useful for laptops to connect to different networks (automatically or by asking for password every time; it's your choice).

The last section of this Installation Summary screen is the Installation Destination part. This information is unique to you and it must be reviewed (it's mandatory). As you can see in Figure 5-15, this screen has four important parts:

- **Local standard disks**. Here you will find the normal hard disks (HDD, SDD, USB) currently available on your system. For now, there's only one regular 1TB HDD disk; in other examples, I'll show you other situations.
- **Specialized and network disks**. This is where you will find LAN disks like iSCSI and FCoE ones. These are usually for advanced users working on a local network. I'll skip this part here.
- **Other storage options**. This is where you decide to accept the automatic settings already made for you or customize them. This is the part that I will explore next.
- **Full disk summary**. It appears unimportant because it's just a link at the bottom, but it's a summary of your disks and you can select from which one you want to boot (usually you don't need to change it).

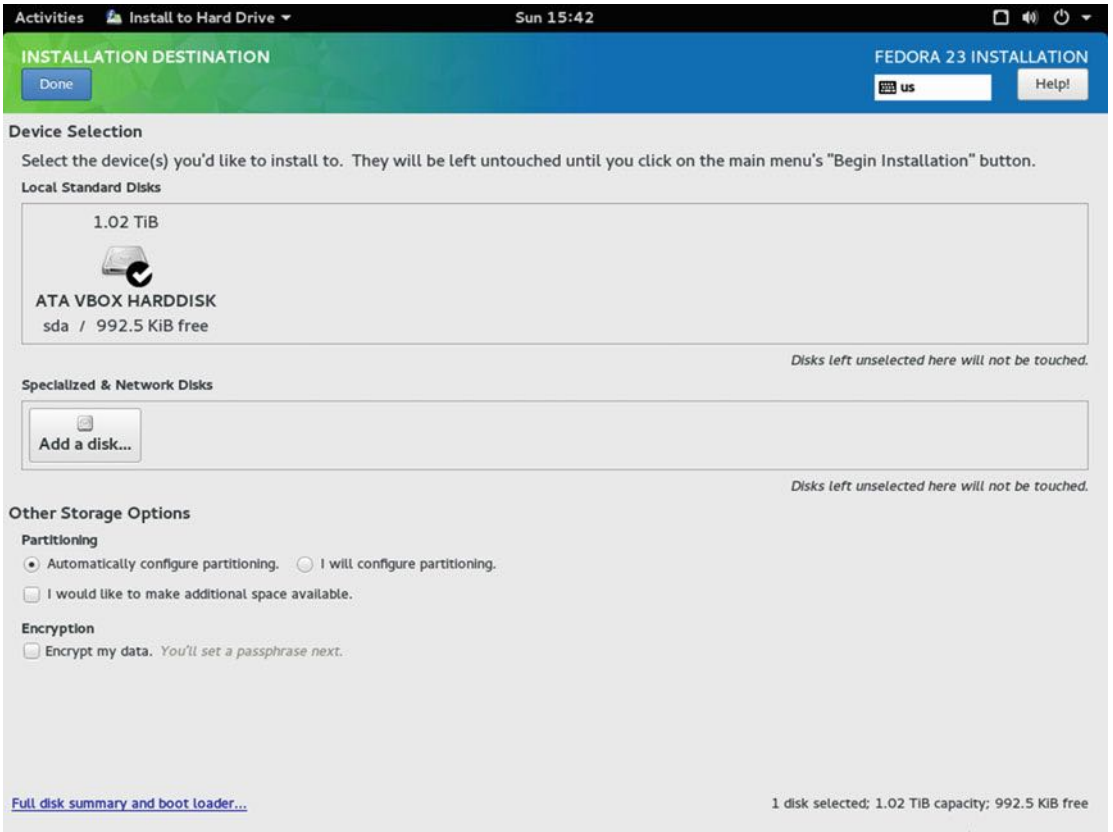


Figure 5-15. *The Installation Destination section*

The Other Storage Options section is where you must take special care about what you are doing. You can decide here if you want to encrypt your data (the entire disk) or not, as I explained in the Ubuntu chapter. I'll skip this step in this installation example. It's the partitioning topic where I want to take a moment to explain the options. As mentioned, this configuration is usually automatically taken care of, so you could simply press the Done button and begin the installation. I would recommend doing this only if you have one disk and it's empty (or if you don't mind losing the data currently there). But if you want to leave some disk space empty (e.g. to install another OS), you must choose the "I would like to make additional space available" option.

Let's ignore all of that and manually configure the partitions by selecting the "I will configure partitioning" option. Once you select that option, press the Done button. You'll get the screen shown in Figure 5-16.

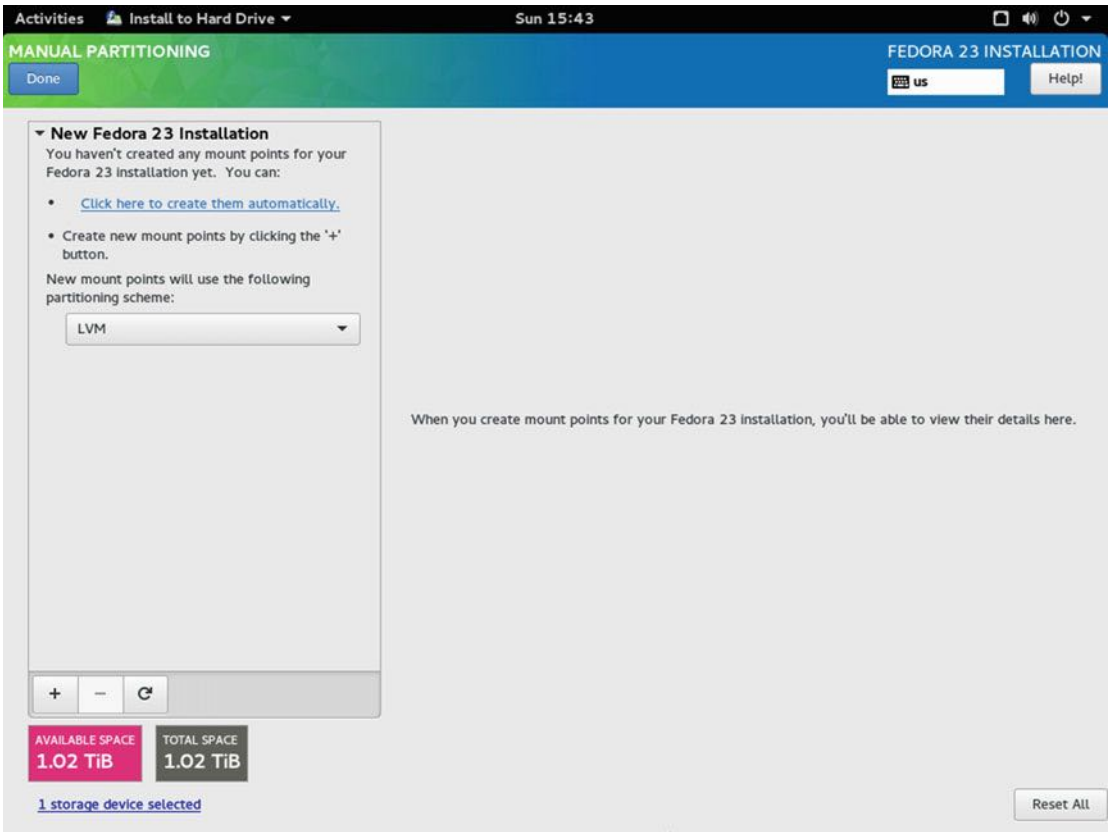


Figure 5-16. *Manual disk partitioning*

It is a blank canvas where you can make a totally customized partition or get assistance from Anaconda. The most important thing here is the partitioning scheme. Fedora currently offers four alternatives:

- **Standard Partition**: The traditional scheme with regular partitions and the ext4 file system by default.
- **Btrfs**: Because it's on the bleeding edge of technology, Fedora offers Btrfs. Btrfs is a file system with volume managing and error tolerance capabilities based on a binary tree similar to Solaris ZFS.

- **LVM**: The Logical Volume Manager offers you more flexibility and ease of use than the traditional partitions (among other advanced functionalities).
- **LVM Thin Provisioning**: This is an advanced feature of LVM that dynamically allocates disk space for the logical volumes when needed. If you are not an advanced user, ignore this.

Fedora by default chooses the LVM scheme because it is more balanced in features/performance/easiness. If you prefer the traditional way, you can choose the Standard one. Btrfs offers you a good number of additional capabilities versus LVM but it is still experimental and its performance is worse than ext4 fs. So if you don't have any particular requirements, keep the LVM scheme.

■ **Note** Consider one particular thing when choosing LVM or BTRFS as your partition scheme: although both offer more flexibility to manage your partitions (like allocation or resizing), there are only a few GUI tools to manage this properly (and they're not very good) so you should manage this volume through the command line.

After you chose your partition scheme, you have two options: to manually add your volumes or let Fedora manage this automatically. Fedora differs here from the traditional way of doing this task; usually you should first define your partitions or logical volumes, then your file system for each one, and then choose the mount points. Anaconda makes you first choose your partition scheme, then your mount points, and the rest is decided automatically (you can change it later). From my point of view and experience, this is a much better approach, especially for non-advanced users.

If you decide to manually add the mount points, you must use the + icon at the bottom of the screen. A dialog similar to Figure 5-17 should appear. You choose a mounting point like /, /boot, /swap, or /home and then the disk space you want to assign to it. You can create as much as you need while there is free space available (you can always change the initial settings by selecting that mount point). And here is where I want to introduce you to the partition scheme that I recommend (it differs from Ubuntu's, which I told you I didn't like).

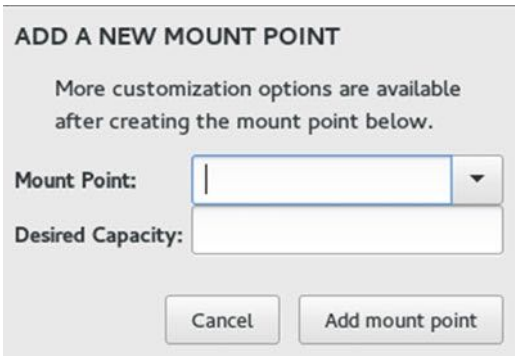


Figure 5-17. *Dialog to add a new mount point*

To refresh your memory, Ubuntu only creates two partitions with two mounting points, one for the swap partition and other for the root (/) partition. All of the files from the system plus your personal data are in the same partition, the root one. This way is easy to manage and wastes less disk space, but only in the short term. I don't like it because in the long term it usually gives you more headaches and problems, so it's not worth it. For example, if you want to upgrade your Linux distribution to a new release and the upgrading process does not work well, you are bound to make a backup/delete/restoration cycle of all your data (at least the `/home` directory) in a mandatory way (although I recommended doing it regularly) if you want to do a fresh installation and not end up with an unmanageable mess. So I recommend that you separate the / mount point into two different ones: the / itself for all of the system files and the `/home` one to store your personal files and program configurations. And I recommend adding an additional `/boot` partition to manage the boot files and avoid future problems with system upgrades, migrations, or certain BIOSes. In order to do this, I recommend a minimum of 500MB for the boot partition (and use a standard partition) and 15GB for the / mount point.

If you let Anaconda do this for you, the result will look like Figure 5-18. It also shows the default automatic partition that Fedora would apply if you choose the default automatic way instead of this pseudo-manual one. As you can see, Fedora chooses the same partition scheme that I recommended you, and it is way better than the one that Ubuntu chooses by default. Fedora always chooses this scheme, regardless of whether you choose Standard, LVM, or BTRFS. Thus, Fedora makes one standard partition for the `/boot` mount point with an ext4 fs and a LVM volume group named `fedora` for the rest as a logical volume with an ext4 fs (`swap` for the swap mount volume). The final free available space is tiny, less than 1MB, so it works very well in that matter. The amount of space dedicated to the `swap` volume is assigned as a function of how much physical RAM you have in your system, because if you want to hibernate your system, the data in your memory is going to be stored in that partition. In each of the volumes/partitions, you have a series of data/options that you can change, from the mounting point and capacity to the file system and name. I like to change the amount of disk space dedicated to the / volume; 50GB is a bit conservative and safe, so maybe 35GB is enough. You can decrement that space in the / volume and press the Update Settings button, and then increment the `/home` volume. As you can see, you can encrypt individual volumes here, instead of the previous screen option that does it globally. Additionally, you can set other advanced features like RAID via the Modify button of the Volume group (this settings affects all of the logical volumes). You can also do things like mix LVM volumes with BTRFS ones, or use a file system other than ext4. One last thing: if you make any changes and don't think they are good, you can always start again from scratch by pressing the ↻ button at the bottom of the volumes/partition list and then the Rescan Disks button in the new dialog that appears. Well, let's leave these settings as is and press the Done button in the left top corner of the screen.

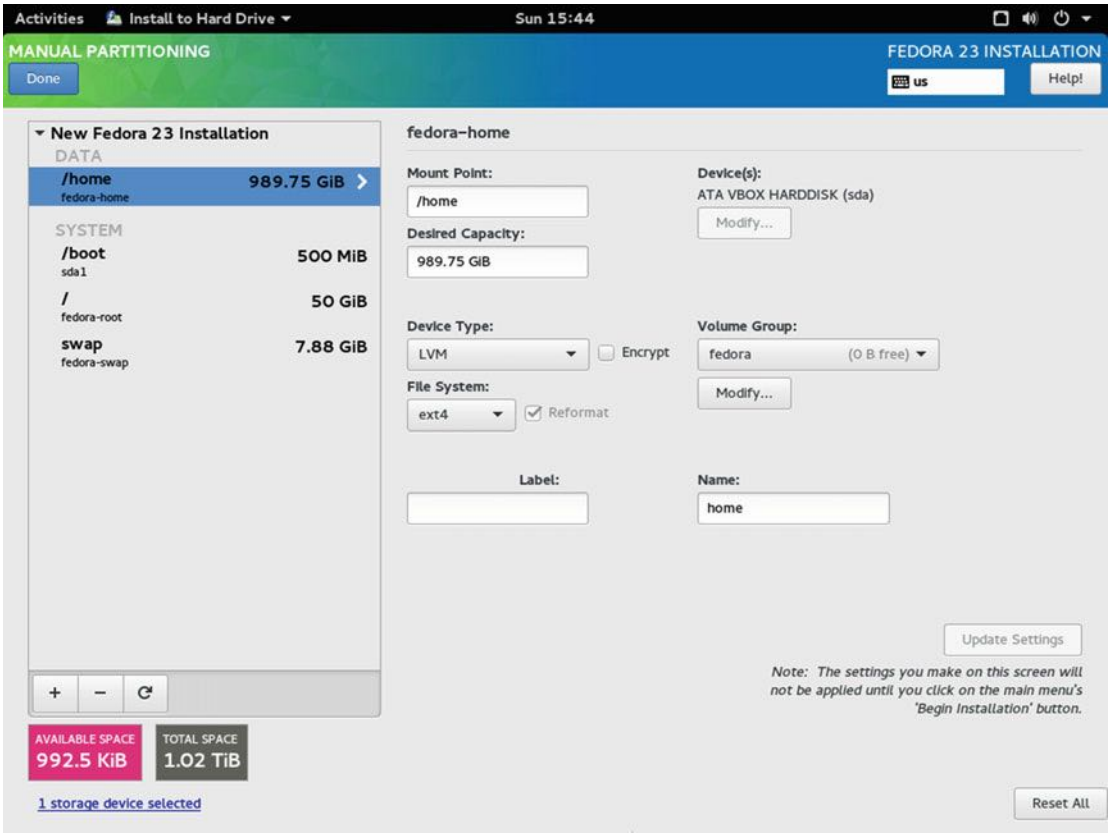


Figure 5-18. *The automatically defined partition scheme for LVM*

A dialog appears, showing you the changes that are going to be done to the disk; if they are OK, press the Accept Changes button (Figure 5-19). You will return to the Installation Summary screen. The Begin Installation button is enabled; press that button and the installation process will write the changes to your disk and begin installing the OS on your system. However, even when the process of partitioning and installing Fedora is going on, you still need to set two more things (Figure 5-20).

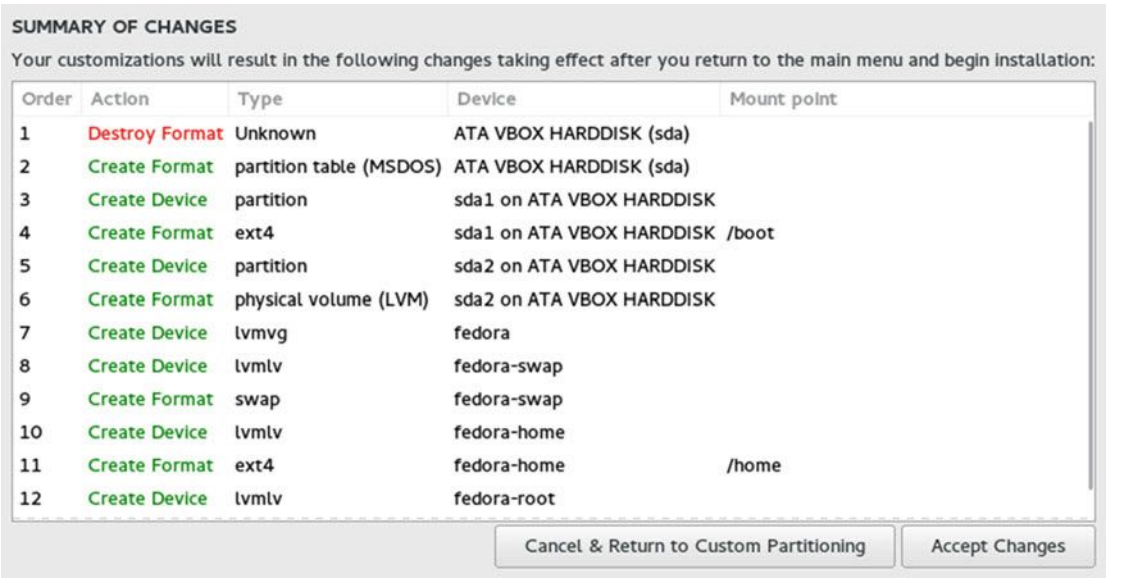


Figure 5-19. *The summary of changes to write to the disk*

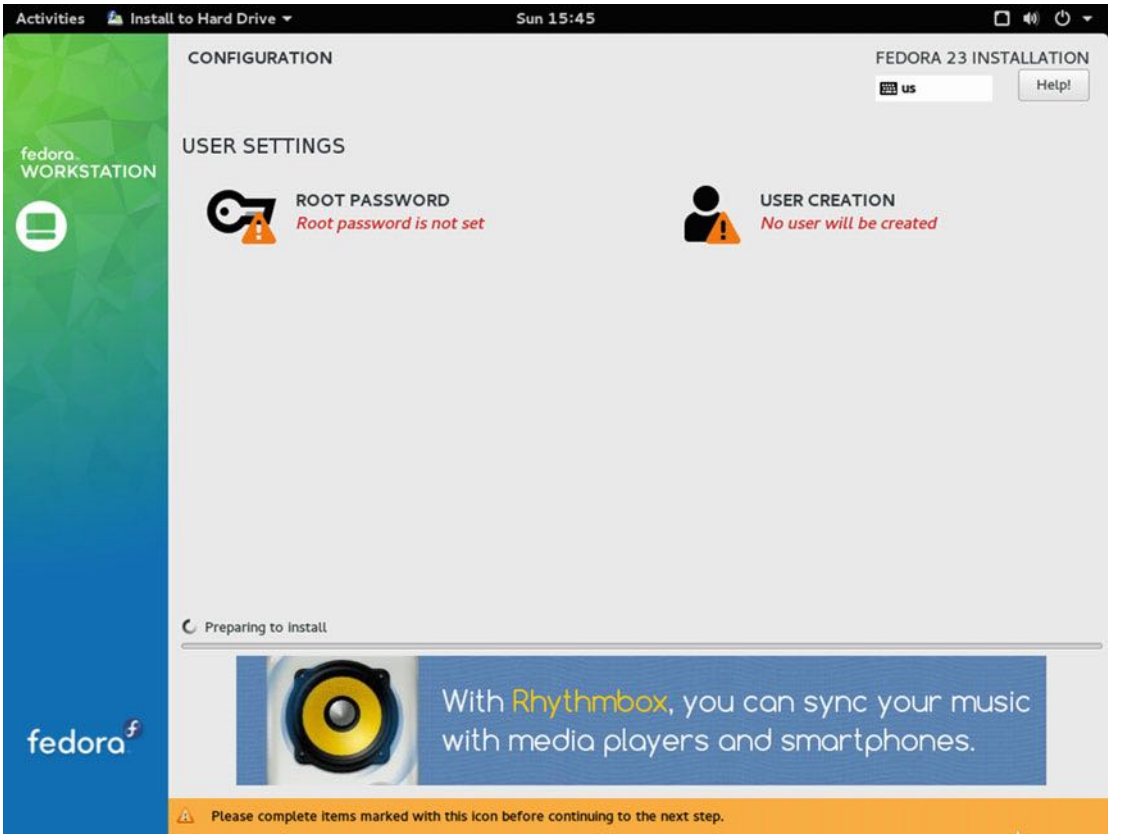


Figure 5-20. *The User Settings screen of the Fedora installation process*

Another big difference here from Ubuntu is that you need to set a password for the root user (the main account and the default admin account); this is disabled by default in Ubuntu (you use your regular user, requiring your password for admin things). This is the right thing to do, although it is less easy and takes responsibility from the user. If you want to emulate the Ubuntu behavior, you can always add your user to the `wheel` group later. Let's establish the root password in the Root Password section (Figure 5-21).

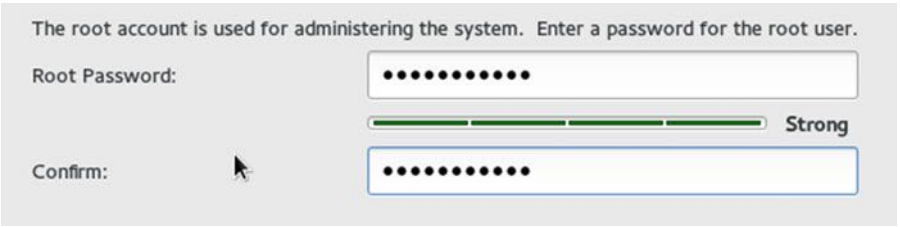


Figure 5-21. *Setting the root password*

Take care here to set a strong password because this is one of the most crucial points of attack when we talking about security. If someone knows your root password, he can own your entire system, Caution! After setting this password and returning to the previous screen, you may notice something weird. The User Creation section is now OK and you are good to go without needing to create another user. You could proceed, but I strongly recommend that you never, but never, use your system with the root account. Always use another regular user to work with your system. It's a serious security flaw (and the main one for many years for Window users). I suppose that Fedora does this in order to allow OEM installations. (You can always create your own later; in fact, you will do it in the Gnome setup if you don't do it now). Anyway, let's create a new user that will be the default one for your computer Figure 5-22).

Figure 5-22. *Create a new user account*

Here you have two main options: to make that user an administrator, and toggle on/off the need of a password. If you want to emulate the Ubuntu behavior, set the first option; the second option should be never set (only to create Guest accounts). Never, absolutely never, set the first option without setting the second one; a user with admin capabilities without a password is a straight route to disaster (especially if your system is connected to the Internet). In the Advanced section, you can adjust these settings, but don't touch them if you don't understand them. After setting this new user and returning to the previous screen, you only have to wait for the installation to finish and then restart the system to be able to enjoy your new Fedora Linux OS (Figure 5-23).

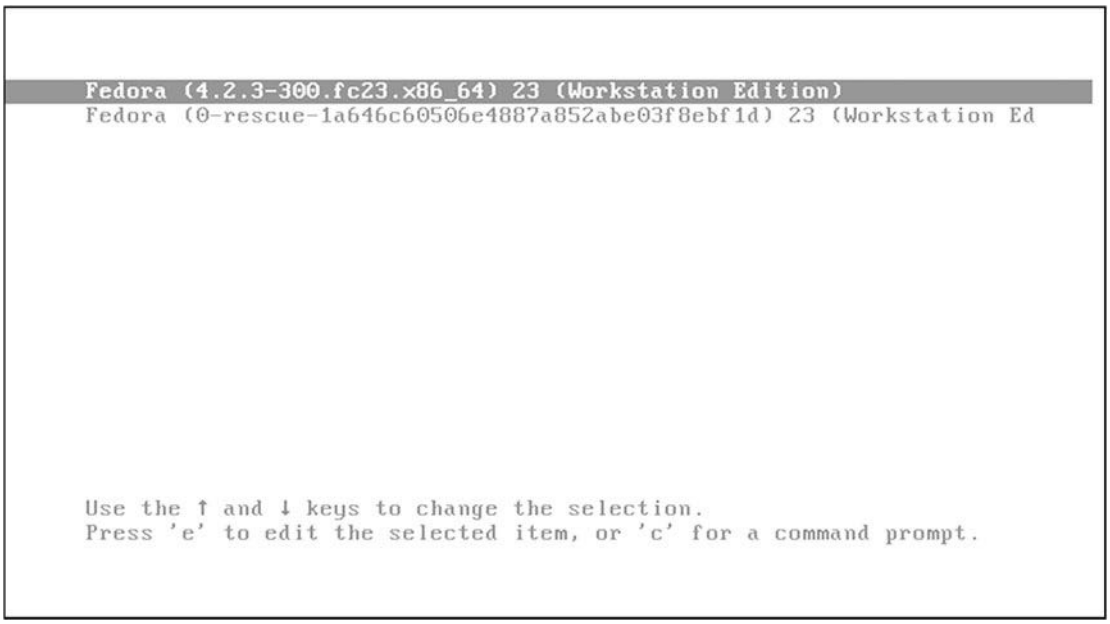


Figure 5-23. *The first screen of a fresh new Fedora Linux OS*

After you restart the system, the first screen that appears is an ugly one (Figure 5-24), but it's very common in Linux; it's a text menu that allows you to choose between your regular Fedora session and a rescue one (which you use if your regular one does not boot). There are also two options oriented to advanced users, like the "e" and "c" keys. If you press Enter or wait a little, the system will continue with the boot process.

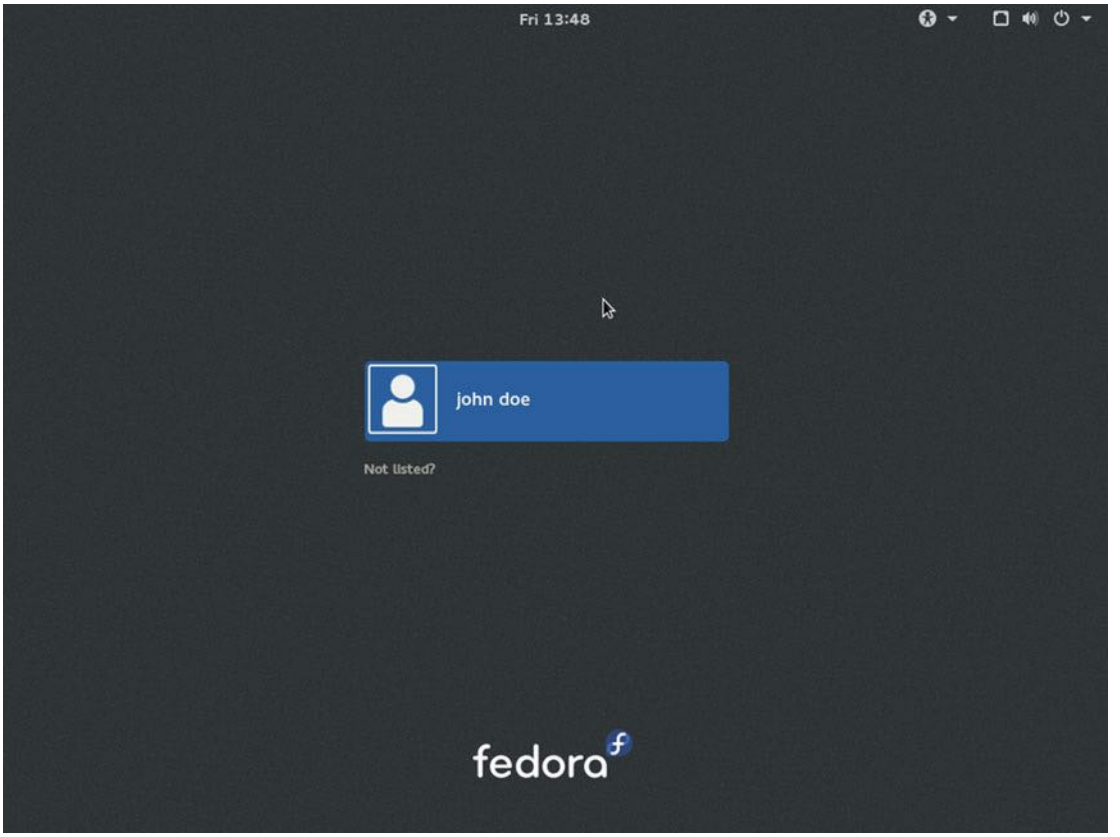


Figure 5-24. *The Fedora login screen*

When Fedora finishes the boot up process, you'll get the Gnome login screen (Figure 5-24) with all of the users listed (at this moment, you only have one). If you select your user, it will ask for your password. When you enter your password, you'll see a gear icon to the right of the Sign In button. This icon (Figure 5-25) lets you to choose between the available desktop environments installed on your system. By default you only have three: Gnome (Gnome 3), Gnome Classic (Gnome 2), and Gnome on Wayland (Gnome 3). I recommend you keep the default one, Gnome, because Classic can cause problems with some applications. Wayland is another example of Fedora's use of new technologies; Wayland aims to be the substitute of the venerable and old X window system, but it's still too new and it may crash with some apps and especially with some graphics cards.

Figure 5-25. *The desktop environments available to you*

So now you're logged in to your new operating system but a new surprise awaits you: the Gnome initial setup. This is where you set up the Gnome configuration for language, keyboard layout, privacy settings, and online accounts (and a local account if you didn't create a user in the installation process). These settings are usually set by default, and if you don't want to change anything, you can skip all of them. At the end, a Getting Started manual (Figure 5-26), part of the Gnome Help system, will be shown to help new users become familiarized with how to do things with Gnome.

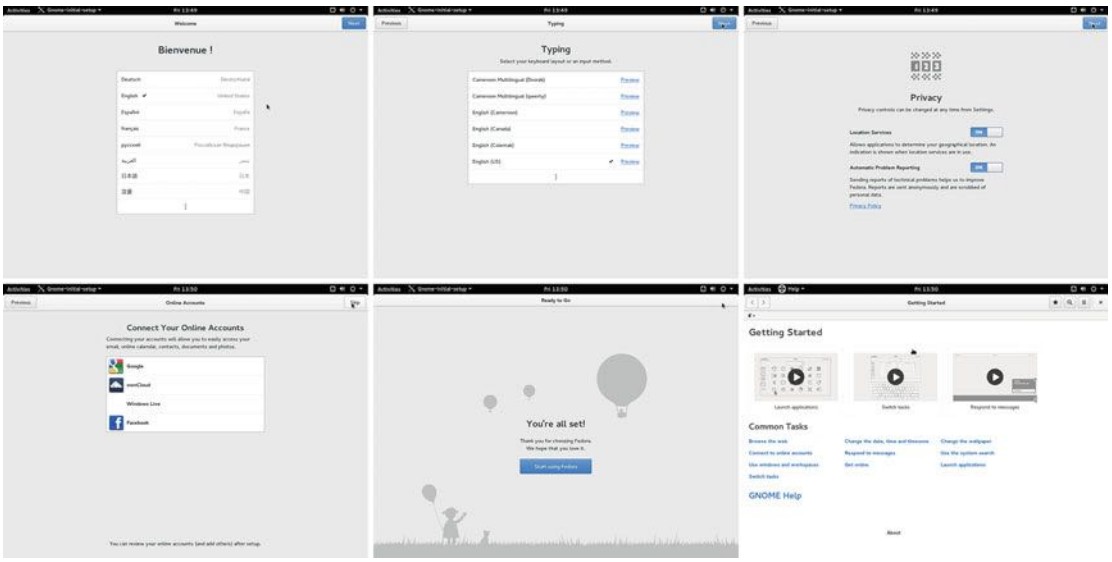


Figure 5-26. *The Gnome initial setup and the Getting Started Help*

And that's it! You can now freely enjoy your Fedora Linux OS. In the next Linux distro installations, you will continue to explore different things.

Maintenance

As with Ubuntu (and in general), there are three essential tasks to maintaining your Linux distro: update and upgrade your distro and manage your applications. But unlike Ubuntu, with Fedora these tasks require the user to use the terminal, something most users dread.

Managing Apps

In Fedora, managing apps is at first an easy task; you can use the Software app (gnome-software) to install and remove applications. It's a very easy-to-use application and it is very similar to the one in Ubuntu. You can see it in Figure 5-27. But you can't find all of the Fedora apps there; for the majority of Fedora's own packages, you must use the command line, like it or not.

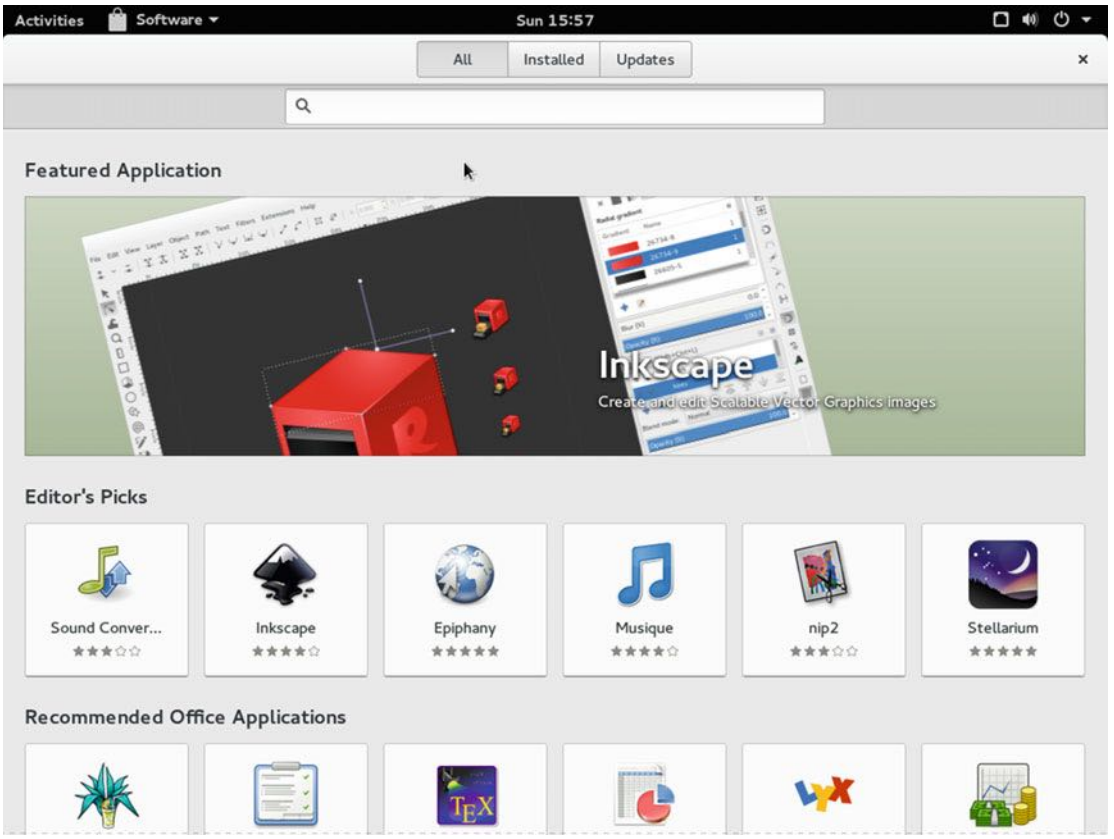


Figure 5-27. *The Software manager application of Gnome*

Updating

Figure 5-27 shows an Updates tab, so you might think that you can update your Fedora from here. Wrong. This is yet another time where you must use the terminal and command line apps. To manage packages in Fedora from the command line, you must use the command `dnf` (formerly `yum`), and you must open the terminal for that. If you want to update the packages of your Fedora OS, you should use this command:

```
$ sudo dnf update
```

Basically this command refreshes the information about the packages (metadata) from the repositories online. If there are updates available, it will show you a list and a summary, and request your confirmation to update all of them. Obviously you should do this regularly to keep your OS up to date (at least to avoid security problems) .

■ **Note** I'm using the `sudo` command here to perform an administration task (update the packages) because I'm assuming that you created your default user at installation time with administrative permissions (as I suggested). If this is not the case, you need to use the root user to do this task. To do so, use this code:

```
$ su root:      # you are going to switch here your user with root
Password:       # you have to introduce here the root password
# dnf update:   # then you can update, don't forget to log out (exit) after end
```

■ **Tip** If you don't want to use the user `root` anymore, you can add your user to the `wheel` group to be able to use the `sudo` command and your password. Use this code (from the `root` user):

```
# usermod -a -G wheel youruser
```

Figure 5-28 is graphical proof of what I'm telling you. The Software app says that there are no updates and that everything is up to date, and the `$ sudo dnf update` command tells you a completely different story. So, you have to update 552 packages (which also requires installing 11 new ones), adding up a total of 799MB. You cannot trust the `gnome-software` app here. If you choose Fedora as your Linux distro, learn how to use the dnf package. It's not that hard and you'll have more control over your system.

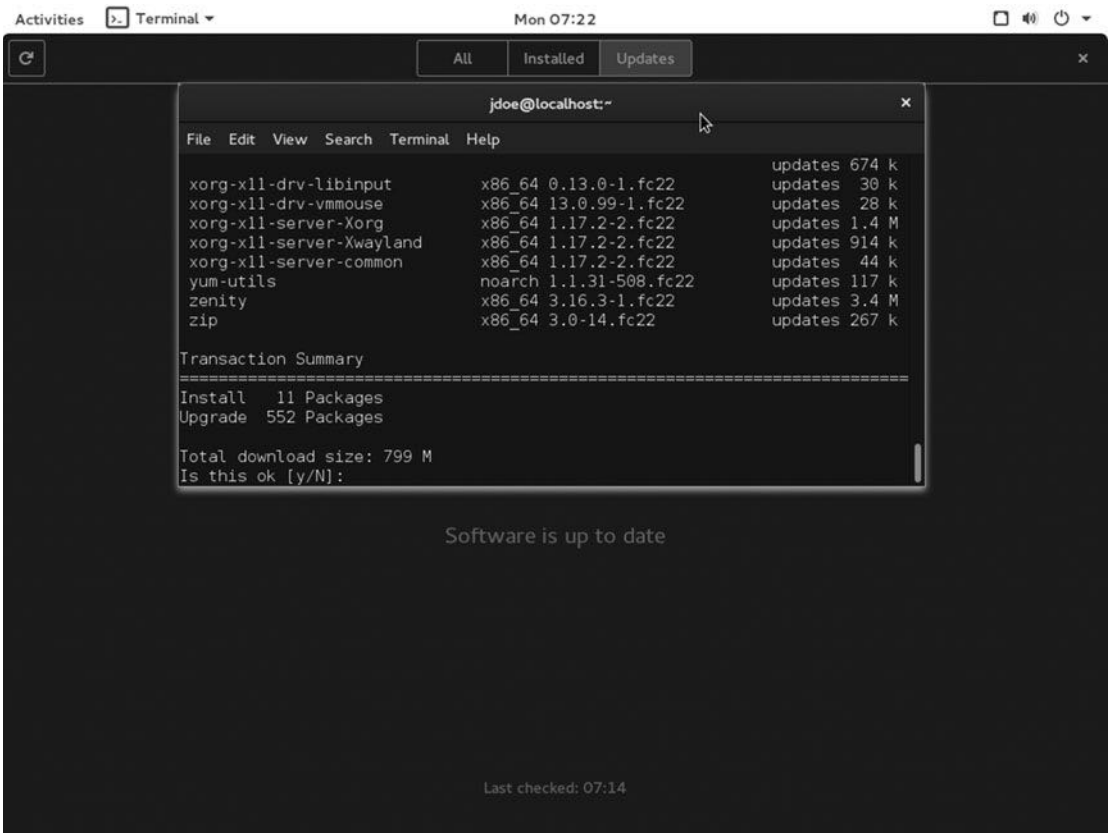


Figure 5-28. *The gnome-software vs. the dnf package updates search results*

You can also set up automatic updates from dnf, but it's not recommended because you lose control of what and when is updating.

Upgrading

If you want to upgrade your current release of Fedora when a new one arrives, the command line and the terminal are your friends. You can use the dnf package to do this, too, but I'll show you a better way. There is a package exclusively dedicated to this task, FedUp (for Fedora Upgrader). It's not installed by default, so the first thing to do is install it.

```
$ sudo dnf install fedup
```

After that, you must update your system (which I just showed you). It's important to do this to avoid any problems with the upgrading process (a very delicate one, as I told you in the Ubuntu chapter; the same applies here). I strongly recommend that you make a backup of your data, because even if you have a dedicated `/home` volume for all of your personal files, bad things happens and you would not be the first one who, after a failed upgrade, tried a fresh install and forgot to not format his current `/home` volume. If I only had a dollar for every time that I saw that…

Now it's time to do the upgrading. In order to do this, you need to have an Internet connection. The command is this:

```
$ sudo fedup --network 23    # 23 would be the number of the relese to update to
```

This will take a while and it is only the first step. When finished, you should have to reboot. Then a different boot screen will appear (Figure 5-29).

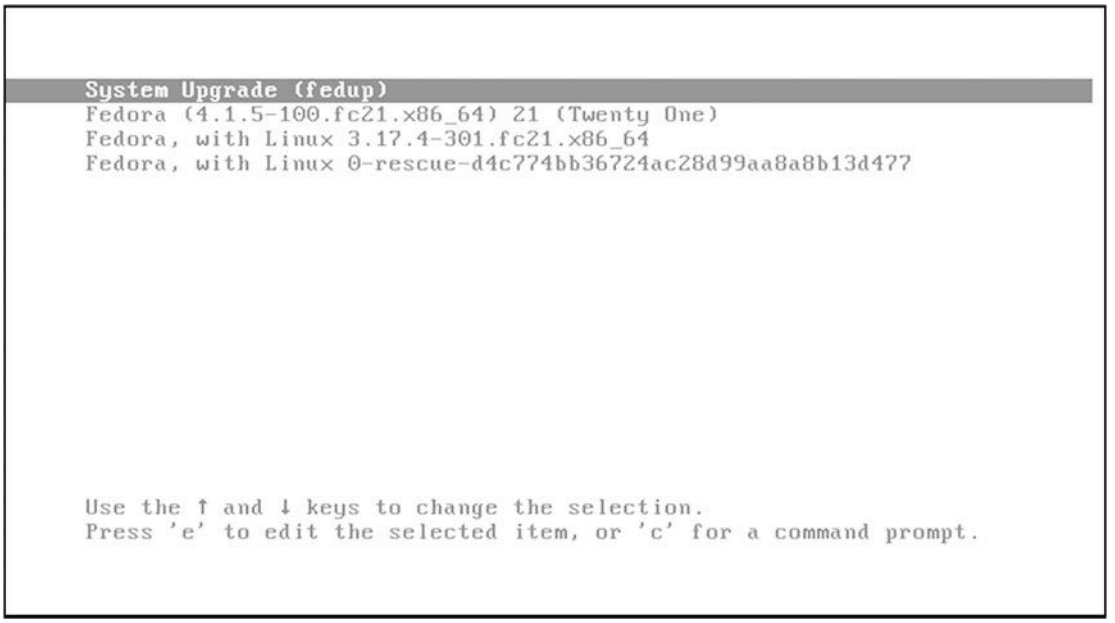


Figure 5-29. *The boot screen that appears after upgrading (21 to 22)*

You should select the first option (or do nothing and wait) to continue to the second step, which will take a while too. If all goes well, you can now enjoy your new release of Fedora. You should also execute a series of commands to make a cleanup and avoid problems when you update your packages:

```
$ sudo rpm --rebuilddb
$ sudo dnf distro-sync --setopt=deltarpm=0
$ sudo dnf install rpmconf
$ sudo dnf install rpmconf
```

More information about this process is at the Fedora wiki page, https://fedoraproject.org/wiki/FedUp.

Pros and Cons

The following is a list of things that I personally see as pros and cons of the Fedora distro. There is always room for discussion in this matter, but I've done my best to be as objective as possible.

Pros

- Fedora has a good balance between ease of use and advanced features.
- It's a very secure distribution by default.
- It uses only free and open source software.
- It has a regular release scheme, so you know when a new release is coming.
- It uses the latest technologies and innovations, and gives a lot back to the Linux community and the rest of the distros.
- It has a large and talented community behind it, which drives it.
- Red Hat is behind Fedora, and Fedora is the base for Red Hat.
- It uses one of the most extended package managers, rpm.
- It is an original distribution, not a derivative.
- There are many task-oriented predefined images (music, games, etc.).
- It has a very good installer application.
- Its DevAssistant tool, with its Docker and virtualization tools integration, provides one of the best environments for software developers.

Cons

- It does not offer much in the way of aesthetics.
- It has no commercial support.
- It has no proprietary drivers by default, so the hardware support in some cases is not good.
- There's a lack of proprietary third party software.
- Because of the two previous points, many people use alternative repositories. This introduces security and instability risks.
- The Gnome Software manager is a buggy and slow application.
- You must use the terminal for a lot of administrative tasks.

Summary

Fedora is the second Linux distro that I analyzed for you. In this chapter, you got to see how Fedora handles certain issues. Now you can see how two distros are not necessarily the same thing. This contrast will grow in further chapters and you will gain an understanding of why there are so many distros and why it is so important to know what you need.

In the next chapter, I put the Debian distro under the microscope.

CHAPTER 6

Debian

Debian is not only one of the most famous, charismatic, and oldest Linux distribution, it is also the perfect example of an independent and community-driven distro. There is a saying among Linux users, "If all other distros disappear, we'll always have Debian." It is the favorite distro of many system administrators, developers, and users, and this is not just due to its long life; it's also because Debian's community, philosophy, and management make it a unique distro. Its "genealogy tree" is the biggest one of all the distros. Debian is the parent of many distros. In fact, almost half of the currently active, maintained distros are based on Debian, either directly, like Ubuntu or Knoppix, or two or three generations removed, like elementary OS or Mint.

History

Since Debian is one the two oldest Linux distros, we could say that the history of Debian is a fundamental part of the history of Linux itself. Let's focus on the beginning. As I said in Chapter 3, Debian was born as an alternative to SLS, the distro that inspired Ian Murdock (then an undergraduate at Purdue University) to make something better. He released the first release of Debian in August 1993. He coined the name by pairing the name of his then-girlfriend (later wife and currently ex-wife) and himself: **Deb**orah + **Ian** = Debian.

With the birth of Debian also came *The Debian Manifesto*[1] (later revised in 1994), which was a declaration of what Debian should and wanted to be. Although it names the Free Software as a fundamental pillar (in fact, Debian was sponsored by the GNU Project for the first year) and the importance of the community, these guidelines focus mainly on the development and maintenance of the distribution.

> *The Debian design process is open to ensure that the system is of the highest quality and that it reflects the needs of the user community. By involving others with a wide range of abilities and backgrounds, Debian is able to be developed in a modular fashion. Its components are of high quality because those with expertise in a certain area are given the opportunity to construct or maintain the individual components of Debian involving that area. Involving others also ensures that valuable suggestions for improvement can be incorporated into the distribution during its development; thus, a distribution is created based on the needs and wants of the users rather than the needs and wants of the constructor. It is very difficult for one individual or small group to anticipate these needs and wants in advance without direct input from others.*
>
> *The Debian Manifesto*, Ian Murdock, 1993-4.

[1] www.debian.org/doc/manuals/project-history/ap-manifesto.en.html

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_6

In June of 1997, two Debian developers, Ean Schuessler (who had the initial idea) and Bruce Perens (who coordinated its creation), created the *Debian Social Contract*[2] to complement the initial manifesto. This contract addresses the moral agenda of the Debian project to preserve the project as independent, free, driven by the community, and under the umbrella of the Free Software. These written guarantees would shape the future of the distro but at the same time would preserve its unique identity. This contract also contains the *Debian Free Software Guidelines* which would serve as the basis of the *Open Source Definition*[3].

Also in 1997, the Software in the Public Interest was created as a non-profit organization by Bruce Perens to allow the Debian Project to accept donations and support it. Over the years it would become the supporter of other Free Software projects.

Philosophy

There is no other distro with its purpose and philosophy as clearly and well defined as Debian (well, maybe Fedora). The initial *Debian Manifesto*, the *Debian Social Contract*, the *Debian Free Software Guidelines*, together with the *Debian Constitution*[4] and the *Debian Code of Conduct*[5], clearly define the guidelines of the distro. This strong commitment to the morals and ethics of Software Libre is the best definition of its philosophy, but let's look at the main points of the *Debian Social Contract* to understand this better:

> *"Social Contract" with the Free Software Community*
>
> *1. Debian will remain 100% free.*
>
> *2. We will give back to the free software community.*
>
> *3. We will not hide problems.*
>
> *4. Our priorities are our users and free software.*
>
> *5. Works that do not meet our free software standards* [are supported; *see the* Contract].
>
> *The Debian Social Contract*, version 1.1, April 2004.

Distro Selection Criteria

Now that you know a little history of Debian, let's see how this particular distro fares on the selection criteria from Chapter 2.

Purpose and Environment

Debian is essentially a general purpose distribution with a unique version for all purposes, thus the same image is used both in the desktop and server environments (among others). The user is responsible for using it in one way or another, and there is almost no graphical and easy help to do so. However, there are several ways to install it, from Live CDs or DVDs to netinstall ISOs. Also, you can opt for different kernels: the standard and official Linux one, a FreeBSD one, and an experimental and unofficial microkernel, the "Hurd" from the GNU Project.

---

[2] www.debian.org/social_contract.en.html
[3] http://opensource.org/docs/osd
[4] www.debian.org/devel/constitution.en.html
[5] www.debian.org/code_of_conduct.en.html

Support

Obviously, all of the support come from the Debian community, and like Fedora or Ubuntu, it has multiple channels:

- **Debian Documentation**: www.debian.org/doc/index.en.html
  - **The Debian Administrator's Handbook**: www.debian.org/doc/manuals/debian-handbook/
  - **Debian Reference**: www.debian.org/doc/manuals/debian-reference/
- **Wiki**: https://wiki.debian.org/FrontPage
- **Mailing Lists**: www.debian.org/MailingLists/index.en.html
- **Forums**: http://forums.debian.net/ and www.debianhelp.org/
- **Q&A site**: http://ask.debian.net/
- **IRC**: #debian at irc.debian.org

Although the Debian support comes from the community, the web site provides links to third party consultants (people/companies) that can provide unofficial commercial support (see www.debian.org/consultants/index.en.html).

User Friendliness

Debian is not a particularly friendly distro. You must know the basics of what Linux is and how to manage the usual administration tasks. It's harder to use than Fedora or Ubuntu but not as hard as Arch or Gentoo. Still, it was the starter distro for many people for a long time, before Ubuntu arrived, and it still is because it continues to be one of the most recommended distros. You will see in the installation process how, although it's not difficult to install, it gives more space to the user. The installer is less friendly than the Fedora and Ubuntu ones. Obviously, if you use the same desktop environment in all of them, it balances things a little, but in the end, the perception of how things work is fundamental. You can see the same phenomenon in Android smartphones: there are so many different phones and the differences between them are minimal, yet people still have strong feelings and preferences for one instead of another.

Stability

Debian has a very particular approach to stability. It can be a very stable distro and it can be a very unstable and up-to-date one. To do this, it has three main branches: Stable, Testing, and Unstable.

- **Stable**: The main and current version. It is the most stable, with only well-tested software, and it is very conservative (the packages may be very old by regular distro standards, except Red Hat/CentOS). It only receives security and major fixes. It follows a standard release model and has a cycle of about two years, but this is not written in stone; a new major version is only released when is finished. The Stable version is the one normally used in servers and workstations.
  - **OldStable**: It's the former Stable version. It is supported until one year after a new Stable is released and then it is archived.

- **Testing**: It is the future new Stable version, where all of the packages, changes, and innovations are tested before deciding that a new version is needed. It receives continuous updates, and when it is time to prepare a new version, it "freezes" and then only bug fixes and improvements are made until it is ready to become the new Stable.
- **Unstable**: Here is where the development of Debian happens; the packages are almost up-to-date and it can be very unstable. It is intended for Debian developers and people that like to always have the latest technology, but this comes at a cost: broken systems. You cannot find any ISO image for this branch (the opposite of the others); you have to change the repositories to a testing installation and then upgrade the distro. It is a sort of rolling release distro, but technically it is the development branch of the distro, unlike true rolling release distros like Arch or Gentoo where it is the main branch.
    - **Experimental**: A sort of staging area for packages that are experimental and with a high probability of breaking your system. It is the only repository oriented to the Unstable branch.

Since 2014, Debian has also adopted a derivation of the Ubuntu LTS releases system for the Stable version of the distro with a LTS repository mainly for security updates. In this way, each Stable release is supported for five years. (The regular support is only for three years.)

A curiosity about Debian it is that uses code names based on characters from the Toy Story films. The current Stable version is called Jessie, the Testing and future Stable is Stretch, and the Unstable version is called Sid (and obviously that never changes). You can learn more about Debian releases at `www.debian.org/releases/`.

Hardware Support

Debian is in the middle ground between Fedora and Ubuntu in terms of hardware support because although Debian supports only Free Software, non-free software is also available in their repositories, like binary drivers for network cards and graphics cards. You can also find private blobs in the kernel. Thus, Debian has reasonable good hardware support.

Aesthetics

The aesthetics of Debian are the default of the desktop environment that you choose; the only aspects that are taken care of are the logo and the desktop background. Despite this and the installer, you may have to customize the DE in order to have something that you like if that is not the default aesthetic.

Desktop Environment

The de facto standard desktop environment of Debian has always been Gnome, but others are available in the DVD ISO image or in dedicated CD images for each DE alternative (but not all of them). Currently these are the DEs that can be installed from the DVD image directly (you can choose another from the repositories):

- Gnome (version 3)
- XFCE
- KDE

- Cinnamon
- MATE
- LXDE

Init System

Until 2015, the init system was the traditional SysV, but now it is the same as almost all of the others distributions: systemd. This decision was the cause of great controversy, and it even caused the creation of a new fork, for the sole purpose of continuing the Debian distro with the old SysV init system: Devuan[6]. Debian still offers SysV as an alternative in their repositories, but installing it and making it work is not an easy or pleasurable task. It requires considerable effort, especially with some DEs.

Package Management System

The package management system of Debian is the same as Ubuntu, since Debian is its original creator: dpkg (as the basis of `apt`). Its repositories hold the most number of packages of any distribution, currently about 50,000 packages.

Architecture

Debian is the distro that supports the most hardware architectures, but some things (like the LTS support for security upgrades) are only available for the amd64 and i386 architectures. The current officially supported architectures are amd64, i386, ARM, i64 (Intel Itanium), MIPS, PowerPC, SPARC, S390, and some variations of those. Debian also unofficially supports other architectures; some of them are not actively developed, but you can find images for those machines. Because Debian has a big community and a focus on freedom, it can afford to maintain some architectures that are not "profitable" for other projects.

Security/Anonymity

Debian offers the regular level of security out of the box. There is a special security team that takes care of finding and fixing security vulnerabilities in its packages, but the distro itself does not come with any extra security features enabled by default. Of course, you can have a very secure distro with Debian but you have to harden it by yourself. There are guides to do this, but you should know what you're doing. In this matter, Fedora and Ubuntu are more secure by default (in that order). You can learn more about Debian security at `www.debian.org/security/`.

Principles and Ethics

Although Debian has a clear focus on and commitment to the Software Libre, it also has a pragmatic view of non-free software. As stated in the *Debian Social Contract*, all of the non-free software available for Debian should work in the distro, and it encourages this. The intention behind this is to let users who may not share the same vision of freedom still use Debian as their distro. On this point, its position is more relaxed than Fedora's. Although the non-free software is not officially supported, it can be installed from their repositories; in Fedora you must use third party repositories to achieve this.

The morals and ethics of the Debian Project and its community can be further researched via the links provided in the "Philosophy" section.

[6] `https://devuan.org/`

Live CD

There are dedicated Live ISO images available for the i386 and amd64 architectures. These are only available for the Stable branch.

Professional Certification

Debian does not offer any professional certification, but it is one of the distros that the Linux Professional Institute endorses in its certifications, the well-known LPIC.

Installation

Installing Debian is not as hard as people say (there is a joke that says that Ubuntu means "I do not know how to install Debian") but, as you are going to see, it is clearly more difficult than the previous installations I've showed you. Don't let this intimidate you. It's perfectly doable by anyone; in the past, almost every Linux was installed in this manner.

Let's begin, as always, by obtaining the ISO image to do the installation. You have to go to the Debian downloads page at `www.debian.org/distrib/`. You'll see something like Figure 6-1.

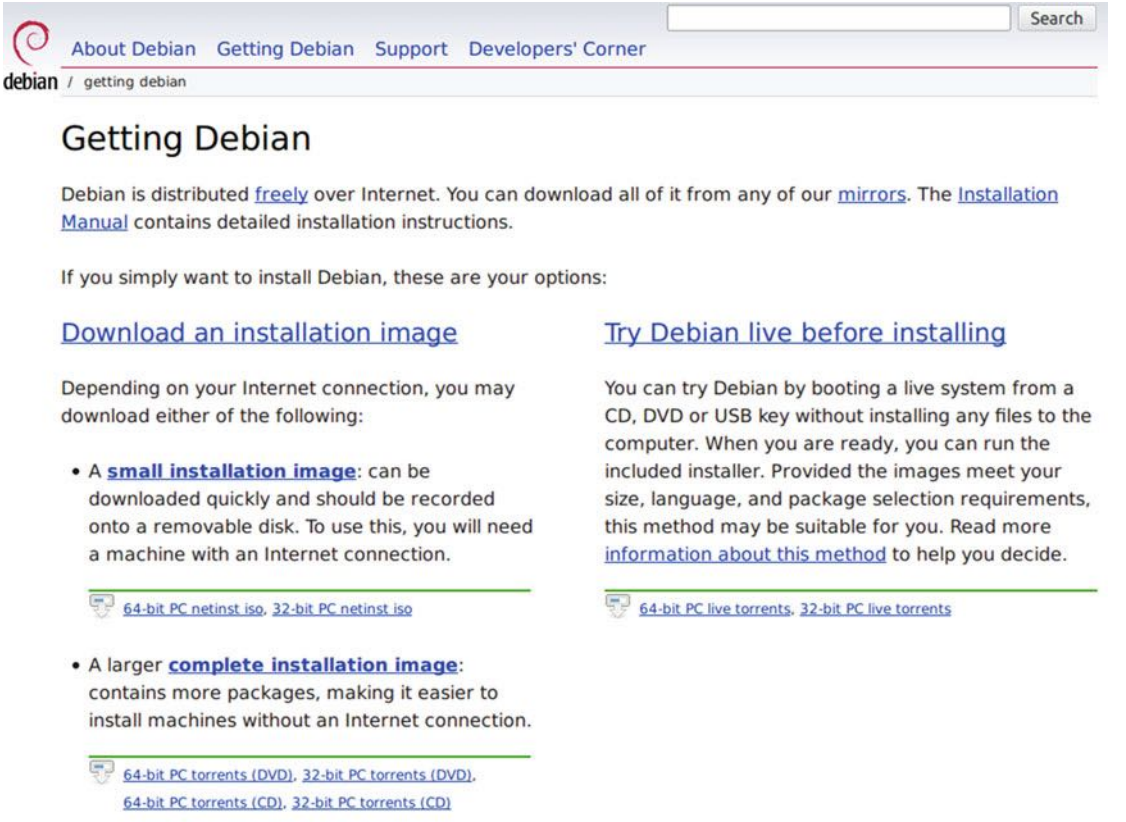


Figure 6-1. *The downloads page from the Debian project*

You can choose from several options: a netinstall image if you have a good Internet connection to install almost all the packages from the Internet, a Live image, or a complete image. With the complete images, you can download all of them (there is a collection of them) to be able to install any package offline, or you can download only the first image to do a basic installation (you can always install anything from the repositories later). This is an interesting alternative if you want to install and have a "hard" offline CD/DVD repository for those places where an Internet connection is not available. Below, on the same page, Debian also offers you options to buy the DVD/CDs or a Debian pre-installed computer from third parties.

Let's choose the most common option, the complete installation image, so click the link and go to a new web page, like the one in Figure 6-2.

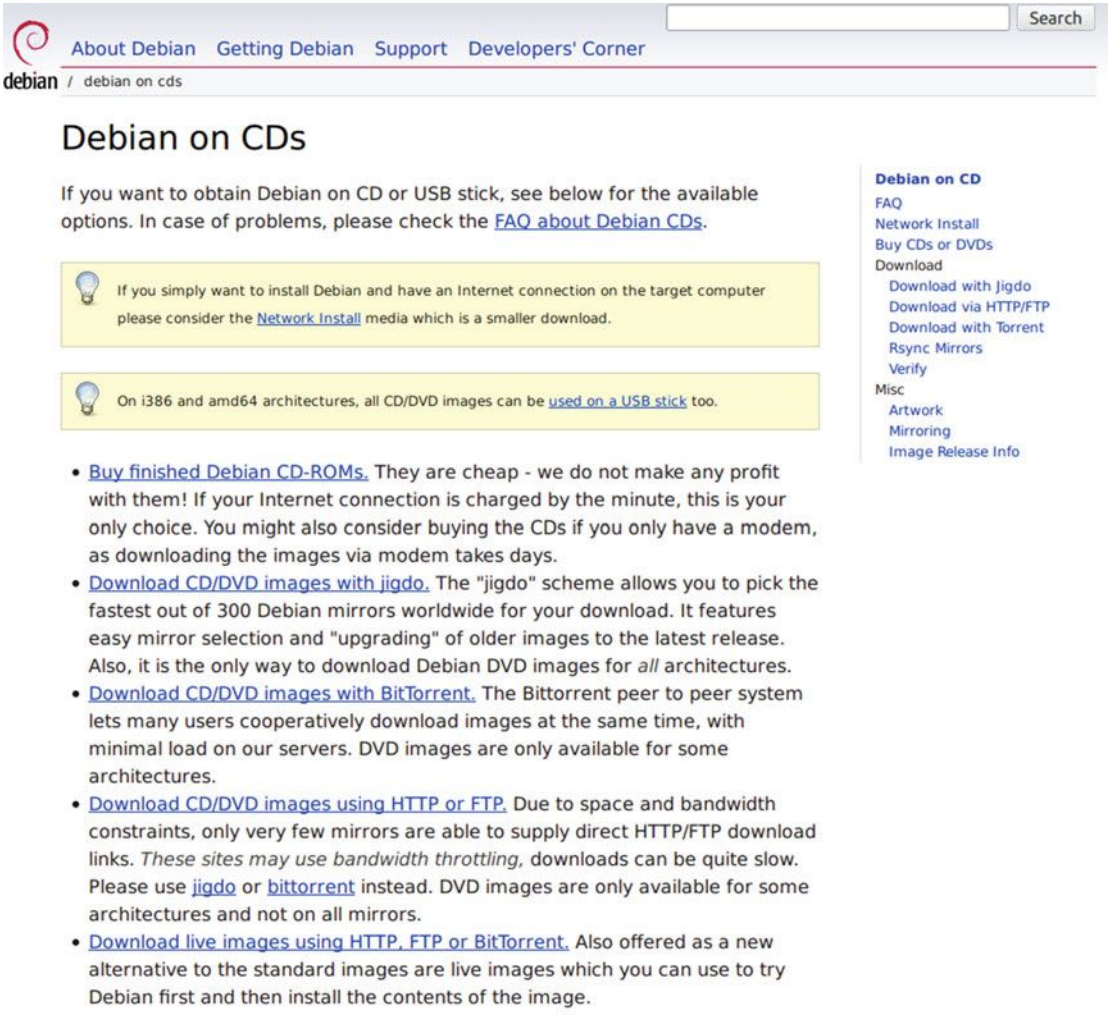


Figure 6-2. *The Debian options to download a complete ISO Image*

More options to choose from! Basically you can download the images using the traditional HTTP/FTP way, using Bit Torrent or jidgo. The best option to not saturate the Debian servers is the Bit Torrent one (especially when a new version is released) or jidgo (a tool to download a file from several simultaneous locations/files, a sort of Bit Torrent based on a client/server model). Let's choose the traditional way. Click the "Download CD/DVD images using HTTP or FTP" option and you will be send to the page seen in Figure 6-3.

Figure 6-3. *Debian Stable ISO images*

Now you can select the Stable version or the Testing one. Choose the Stable one; it's available in two formats: CD and DVD. The CD is currently is a collection of eight pieces; there are only three in the DVD format. Get the image corresponding to your computer architecture; in my case, I got the amd64 version. Click it, and you go to the page shown in Figure 6-4, where you actually can download the image that you need.

Name	Last modified	Size
Parent Directory		-
MD5SUMS	2015-06-08 00:17	893
MD5SUMS.sign	2015-06-08 00:32	819
SHA1SUMS	2015-06-08 00:17	1.0K
SHA1SUMS.sign	2015-06-08 00:32	819
SHA256SUMS	2015-06-08 00:17	1.3K
SHA256SUMS.sign	2015-06-08 00:32	819
SHA512SUMS	2015-06-08 00:17	2.2K
SHA512SUMS.sign	2015-06-08 00:32	819
debian-8.1.0-amd64-DVD-1.iso	2015-06-06 17:33	3.7G
debian-8.1.0-amd64-DVD-2.iso	2015-06-06 17:33	4.4G
debian-8.1.0-amd64-DVD-3.iso	2015-06-06 17:33	4.4G
debian-update-8.1.0-amd64-DVD-1.iso	2015-06-07 02:41	4.1G

Figure 6-4. *The Debian DVD images currently available for amd64*

After you select the first DVD (you only need the first one), it starts the download. When it's done, you can boot up your computer for the first time with that image (Figure 6-5).

Figure 6-5. *The first screen of the Debian installation*

After your computer has completely started, the first screen will be very familiar to you because it's the same as you saw in the Ubuntu/Fedora installations. However, it offers some unusual options. There are three installation options (Install, Graphical install, and Install with speech synthesis). Additionally, the advanced options include Expert install, Rescue mode, Automated install, and the graphical versions of them. You can see both menus in Figure 6-5. The options without the word "graphical" at the beginning are text interfaces, which are very useful for systems where the auto-detection of the graphic card does not work well or for a faster option for those used to it (it's the one that I prefer). The graphical one is the friendliest one, and you can use the mouse. You can see the difference between them in Figure 6-6. So choose the Graphical install option and begin the installation process.

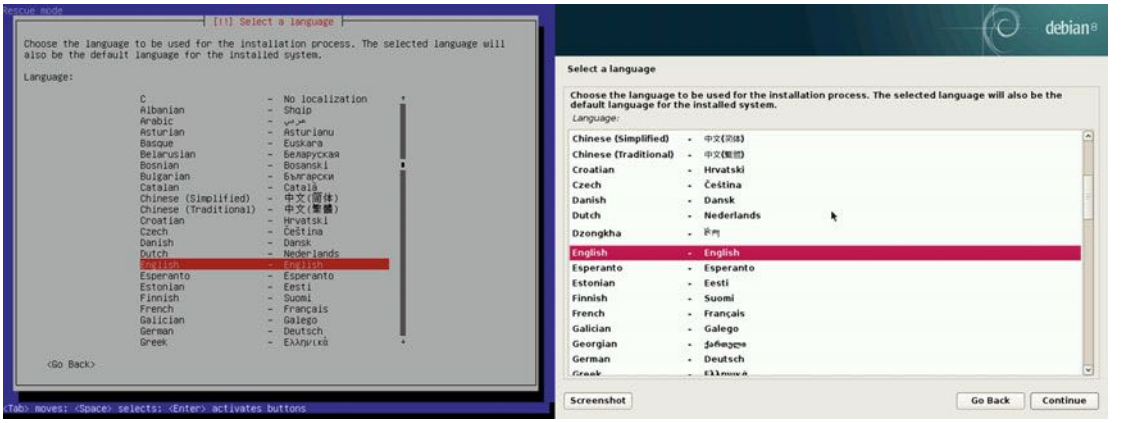


Figure 6-6. *Both versions of the installation process: text-based and graphical*

The first screen (Figure 6-6) ask you for the language that you want to use in the rest of the process, and then your location and your keyboard template, as you can see in Figure 6-7.

Figure 6-7. *Location and keyboard selection*

Next, the installer will show you progress bars, the first to detect and mount your CD-ROM (or the ISO image on a USB drive), the second one to load the rest of the installer components from the image. A third one will appear to detect and auto-configure the network; if not, it will prompt you for your network values.

The next screen (Figure 6-8) is about the hostname of your system; it suggests "debian" by default (and I left it) but you can choose whatever you want. Next, it asks for a domain name, but you can leave it blank, as I did in Figure 6-9.

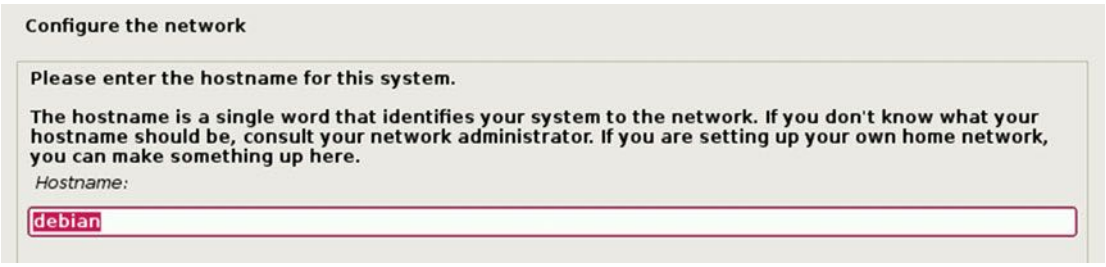


Figure 6-8. *The hostname value*

Figure 6-9. *The domain value*

Now you have to set the password for the root user (Figure 6-10). Remember the advice I gave you in the preceding chapter and choose a strong one here, or leave it blank to disable this user and use your regular user and the command "sudo" like in Ubuntu; it's up to you. I set one here because old habits die hard, but the latter option is completely fine (some prefer it because the bad guys must guess two things, the user name and the password, to attack the system remotely).

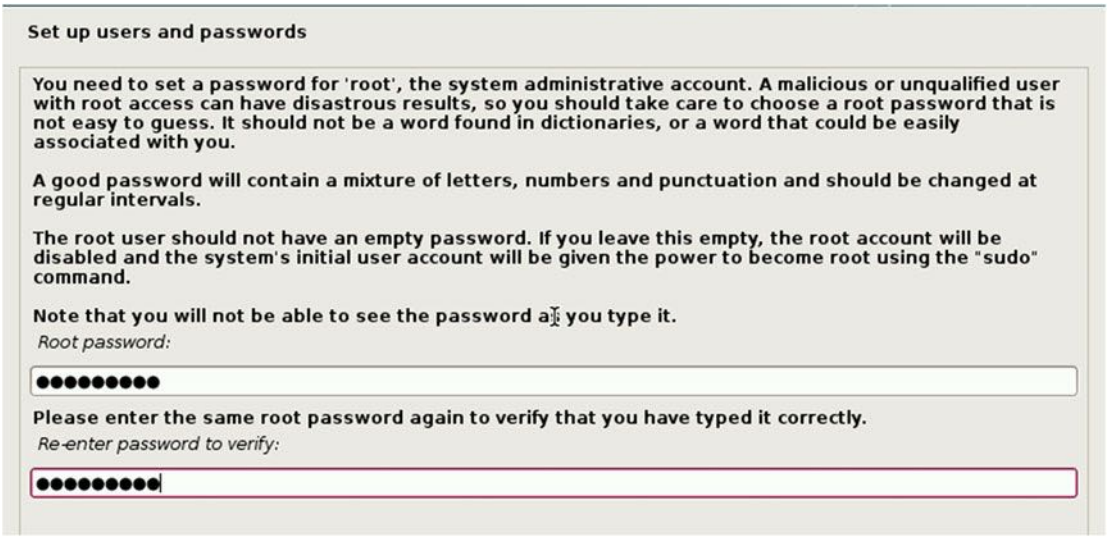


Figure 6-10. *The root password*

In the three next screens (Figures 6-11 to 6-13) you will fill in your full name, the user name, and the password for your user. Choose a good, strong password, especially if you left the root one blank and you are going to use sudo.

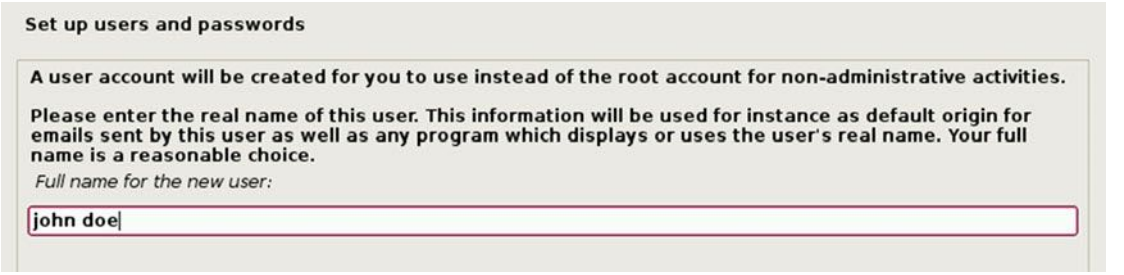


Figure 6-11. *The full name of the user*

Figure 6-12. *The user name for the account*

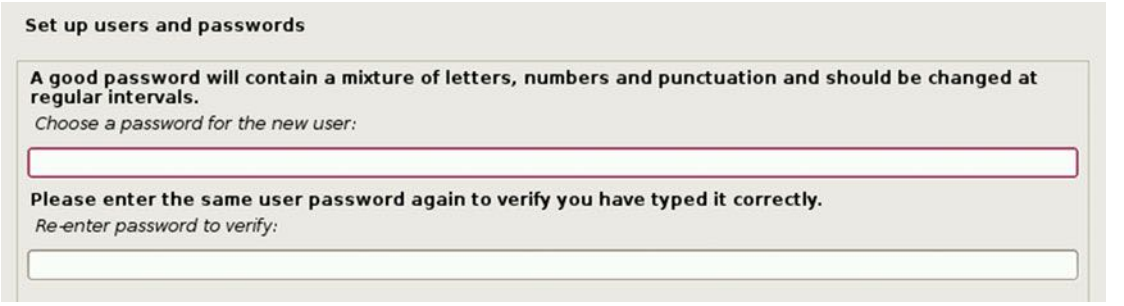


Figure 6-13. *The user password*

The following step is to configure the time zone (Figure 6-14); if your country only has one zone, this step will be skipped.

Figure 6-14. *Configuration of the time zone*

Now you begin the hard part of the installation: using the partition manager. After it detects your disks, you will see a screen like Figure 6-15.

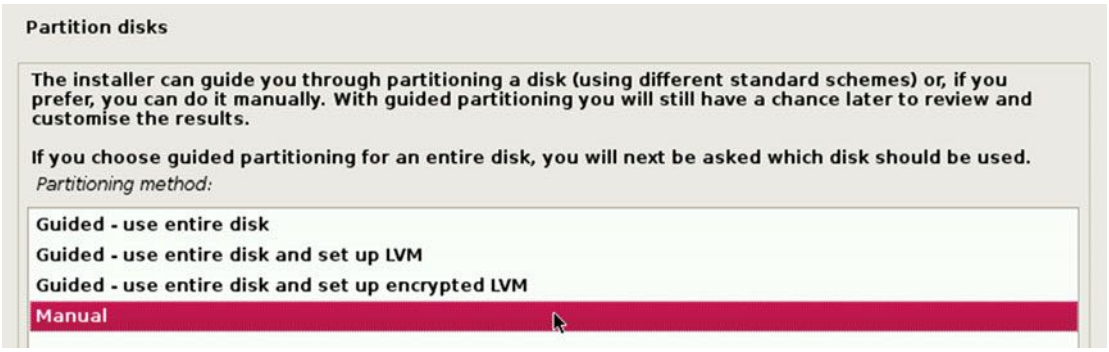


Figure 6-15. *Selecting the partition method*

I'm going to show a different scenario from the previous ones. Let's imagine a very common situation: a desktop system with two hard drives, one SDD (250GB) and one HDD (2TB). To get the best of both drives, you probably want to have the system and configuration files on the SSD for good general performance and your personal files and data on the HDD for good storage capacity. To do this, you must choose the Manual option on the menu.

The screen in Figure 6-16 is what you should find in a scenario like the one I propose. You can see the two disks; you can differentiate them by the capacity and how both don't have any partitions yet. Take a good look at them. In fact, I suggest that you write this info down on paper because you will need it later. Now you must make the partitions. I will show you how to make one of them and you can figure it out how to make the rest.

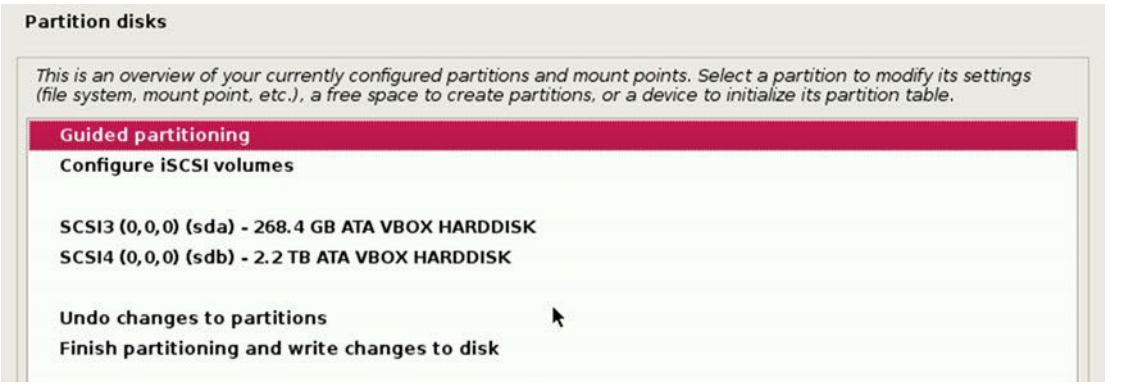


Figure 6-16. *The initial partition screen*

From my point of view, this is the weakest part of the install program, because it's not that intuitive and it's hard for newcomers to understand how the interface in this part works. I will show you the easiest way possible.

Let's begin with the first partition of the SDD. To do this, click the description of the disk. You'll get a screen similar to Figure 6-17. It asks for confirmation to create a new empty partition table on the disk. In this case, let's assume that both disks are new, so say yes.

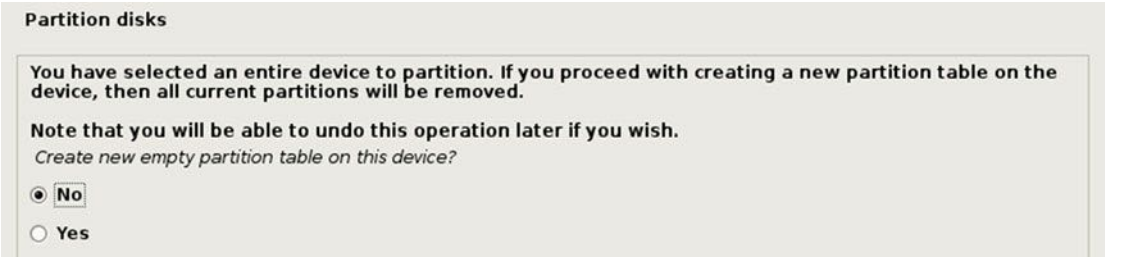


Figure 6-17. *New empty partition confirmation*

In Figure 6-18, you can see how the SSD disk has a new empty partition table. You can also see that you have new options in the superior menu. To simplify things, I will avoid any LVM, RAID, or encryption settings. Thus, to create your first new partition, click the partition table.

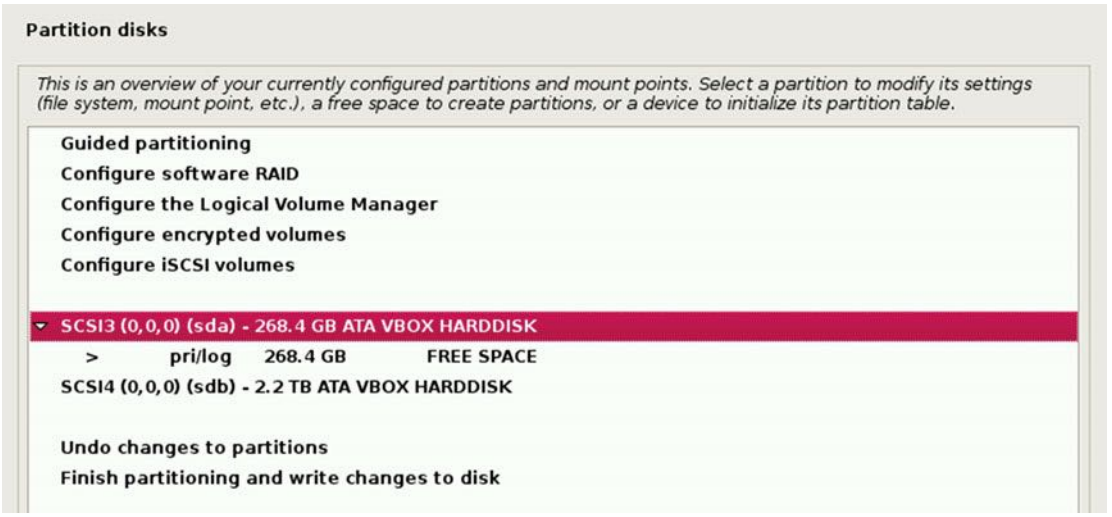


Figure 6-18. *A new partition table is created*

Another screen appears (Figure 6-19). Select the "Create a new partition" menu entry.

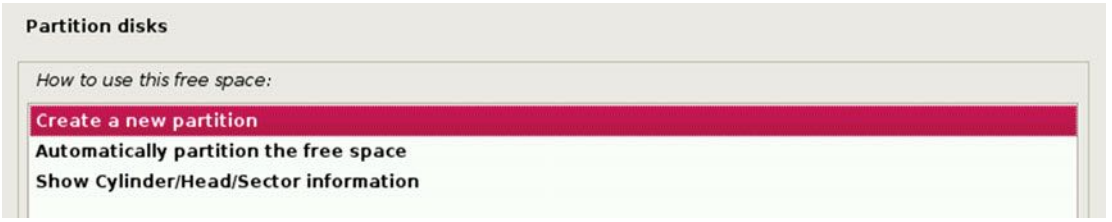


Figure 6-19. *Choosing the action on the free disk space*

The first partition that you are going to create is the boot partition with a size of 500MB (Figure 6-20).

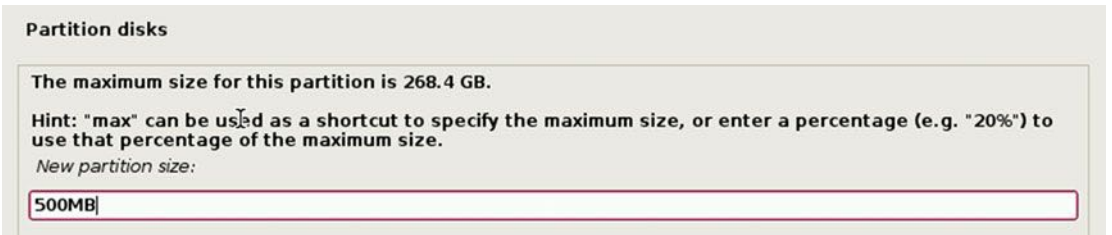


Figure 6-20. *Size of the partition*

Next, you must set the type of the partition (Figure 6-21), Primary in this case, and then you must say where you want to allocate the partition, at the beginning or at the end of the free available space (Figure 6-22). Obviously you want to allocate this partition at the beginning (actually, being an SSD, this is not as important as in the hard disk case, but it is the norm).

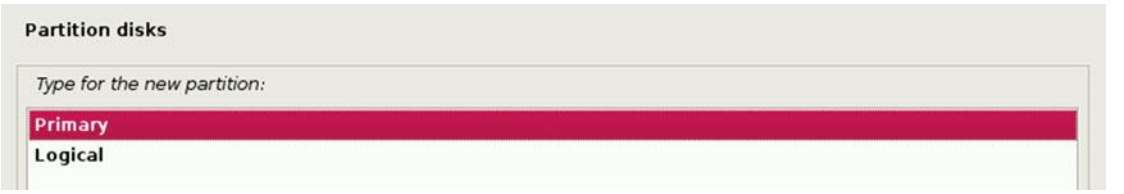


Figure 6-21. *Selecting the partition type*

Figure 6-22. *Where the partition is allocated*

The screen that appears will show the default settings of this partition (Figure 6-23). Each one of them can be customized; in this case, let's keep the defaults except for the mount point, which you set to `/boot`, and the bootable flag that you need to activate. In the rest of the partitions, you only need to set the mount point and the file system for the swap partition (as "swap" instead of "Ext4"). You can safely ignore the rest of the parameters in the majority of the cases. You may want to change the file system and the typical usage, for example to store large files (like videos), or change the mount options to establish disk quotas.

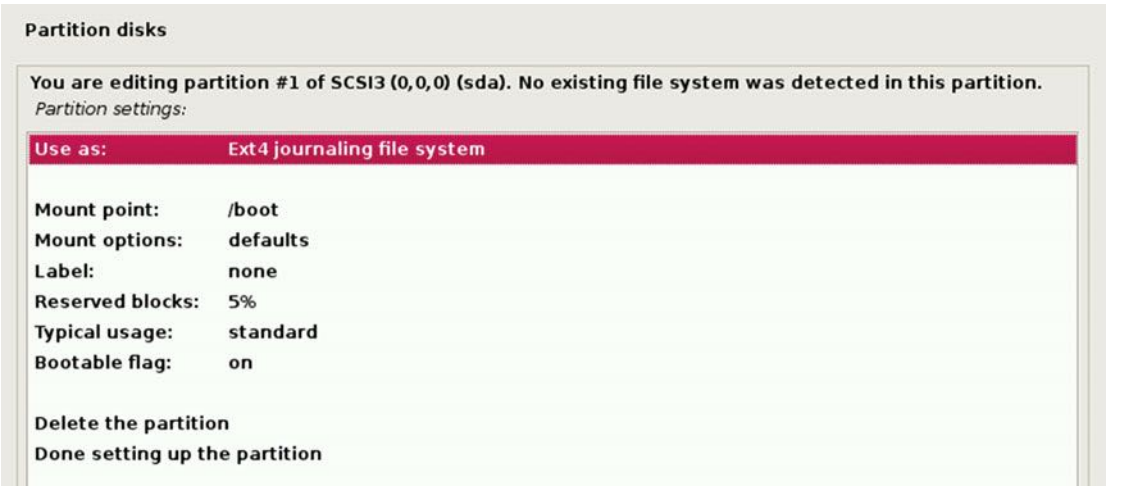


Figure 6-23. *Partition settings*

As you can see, this method is a little less friendly than the graphical installers of Fedora and Ubuntu, but it's not as hard as the rumors said. Actually, comparing it to both previous distros, it's not that hard to install it. The maintenance is where things change a bit versus Fedora and a lot versus Ubuntu.

Once you set the partition settings, you can see how the partition table has changed to reflect that (Figure 6-24). All of the important parameters are shown here: number, type, size, boot flag, format flag, file system, and mount point. And you can see the amount of free space that remains after creating the new partition.

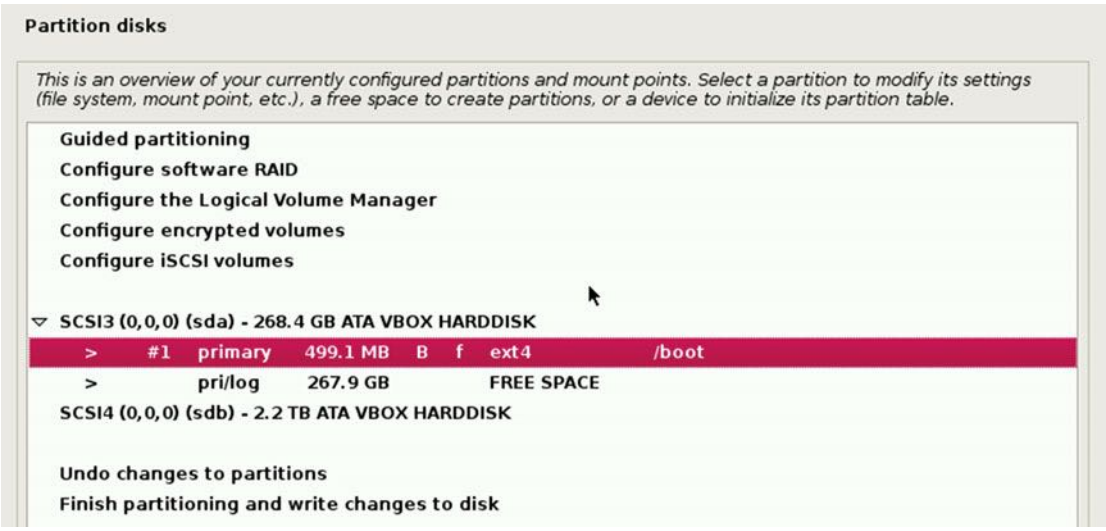


Figure 6-24. *A new partition is created*

You can proceed in the same manner to create the rest of the partitions to achieve the goal shown in Figure 6-25. At the end, you should have the following partitions/mount points:

- `/boot`: The partition where you store the files needed to boot the system. It is the only one with the boot flag "B" activated.
- `/`: The root partition, where the system files are stored. This is located in the SSD to improve the performance of the read/write operations.
- `swap`: The size of this partition depends of the amount of memory that you have; these days, a good option is to make it almost as big as your RAM memory (to allow the system to suspend to disk or hibernate). Some people think it's a bad idea to put this partition on a SSD drive; well, the days when they easily wear out are behind, and when you have enough memory, this partition is usually only used to store the contents of your RAM when hibernating. So it's not a problem to locate this partition here.
- `/home`: Where your configurations files are going to be stored. By default, this partition should be used to store your data files too, so why not create this partition in the HHD? For same reason you created the data partition in the HDD, to achieve a balance between speed and big storage capacity. You want to have a big space available to store your files (especially multimedia, like your pictures or your videos), but at the same time you want your programs to start quickly and have good performance. The binary files of your programs are going to be in the `/` partition, but the configuration files for a lot of those programs are stored by default in the root folder of the `/home` partition. So, no matter how fast your program starts, if it has to wait to read the configuration file in the HDD, it's a problem. You should store only these files and the files that you want to speed up because you use them frequently, such as work files like spreadsheets, etc.

- `/home/john/data`: This is where you should store all of your big files as well as files that you do not use regularly. It has the disadvantage of worse performance but it has great capacity. Notice that I used the following scheme for the mounting point, `/home/youruser/data`, where john is the user I created before. Usually this would be mounted as something like `/media/data` or `/run/media/youruser/data`, but I used this little trick to avoid to using a symbolic link later into the `/home` directory and to make it appear more natural to the users not accustomed to working with mounted/removable devices, so it will show up like another additional folder in your `home` directory. All of these methods have little inconveniences/annoyances that I will show you later; I'll also show you how to fix them easily.

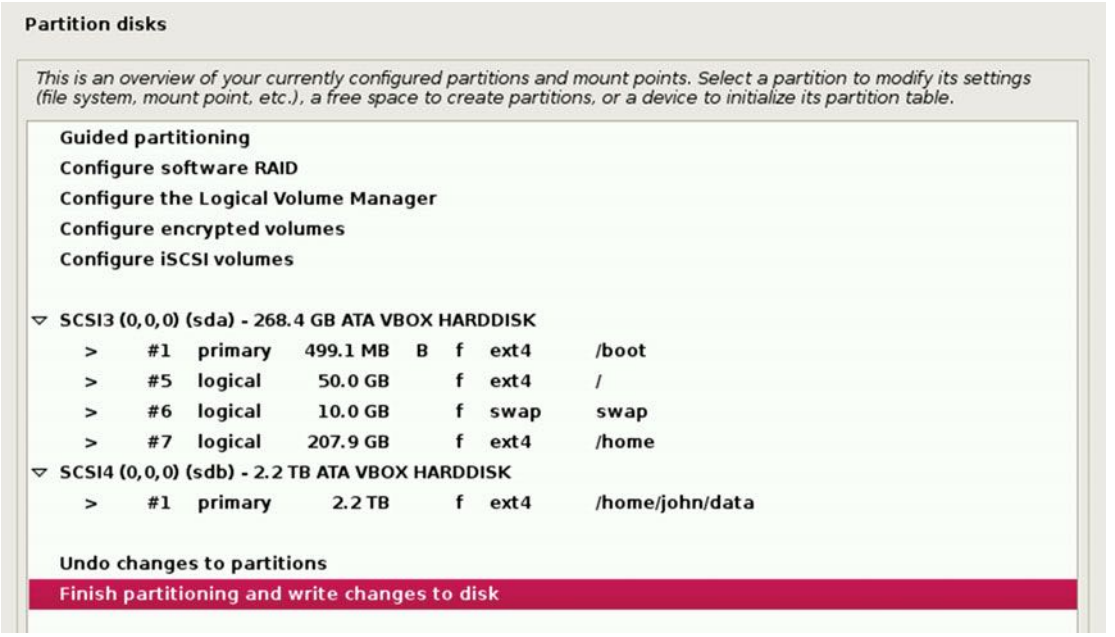


Figure 6-25. *The final partition settings for both disks*

Finally, after you create all of those partitions, the installer will ask you to confirm the writing of these changes to your disks by showing you a resume of them (Figure 6-26). If everything is OK, you can continue with the installation.

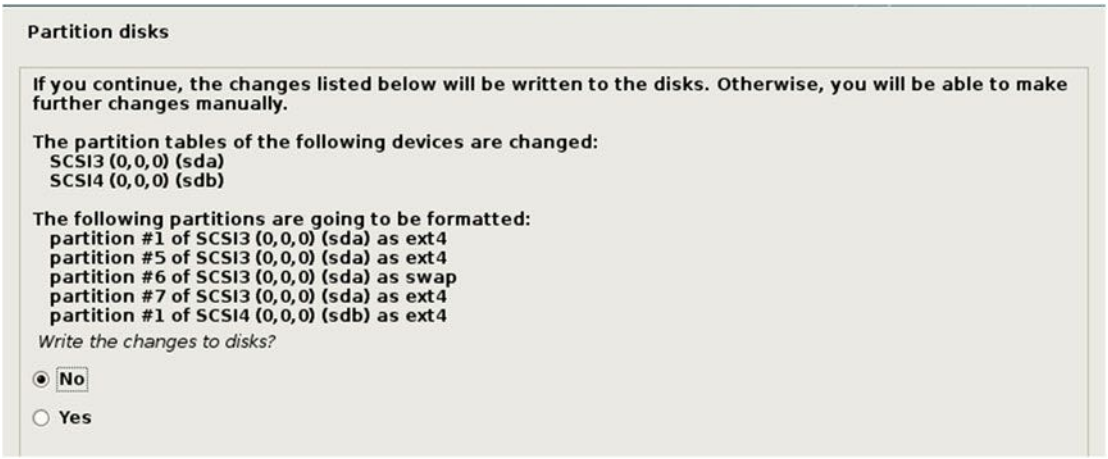


Figure 6-26. *Disk changes confirmation screen*

The installer is going to create the partitions, format them, and then begin to install the base system. This will take some time (Figure 6-27).

Figure 6-27. *Installing the base system*

After a while you have the option of scanning more CD/DVDs (images) if you already downloaded them. This can be very useful when there is no Internet connection available. In this case, select the No option and continue (Figure 6-28).

Figure 6-28. *Optional scanning of another CD/DVD*

The image that you downloaded contains all the packages that you need in order to install the whole system, but if you want to use a network mirror (Figure 6-29), you can benefit from the newest versions available and save time on your next update (also, you will have the latest security updates from the first moment that you start your new OS for the first time). Select the Yes option. Then you must configure some settings to allow this feature.

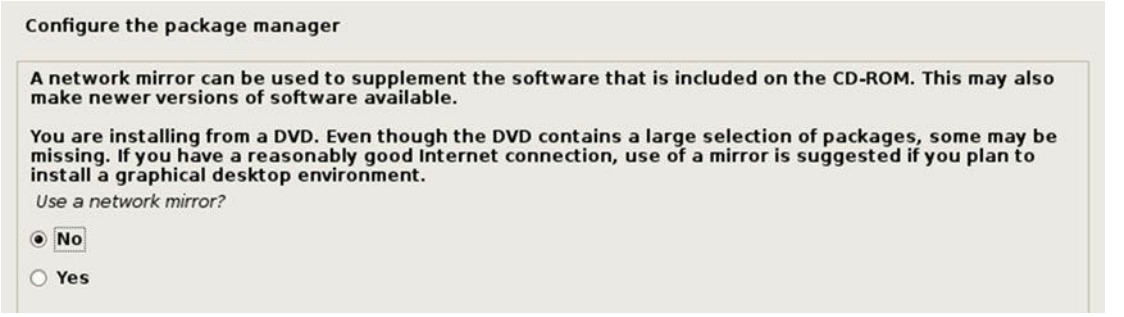


Figure 6-29. *Deciding to use a network mirror*

You need to select the country where the mirror server is located (Figure 6-30), the actual mirror server (Figure 6-31), and if you are going to use a network proxy (Figure 6-32).

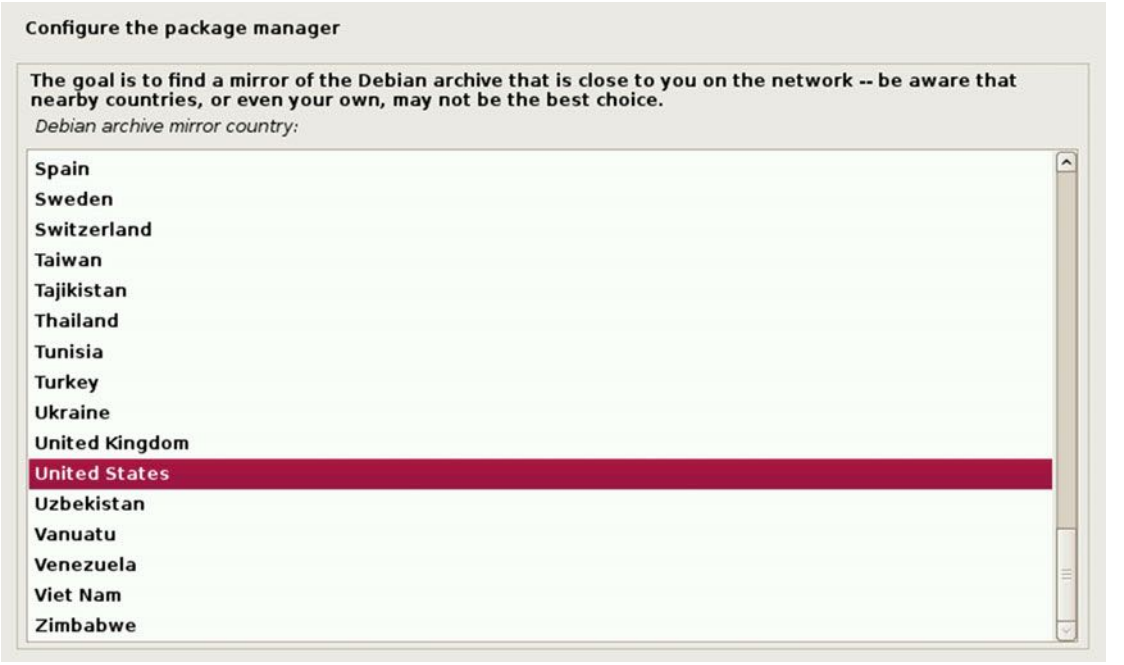


Figure 6-30. *Choosing a mirror country*

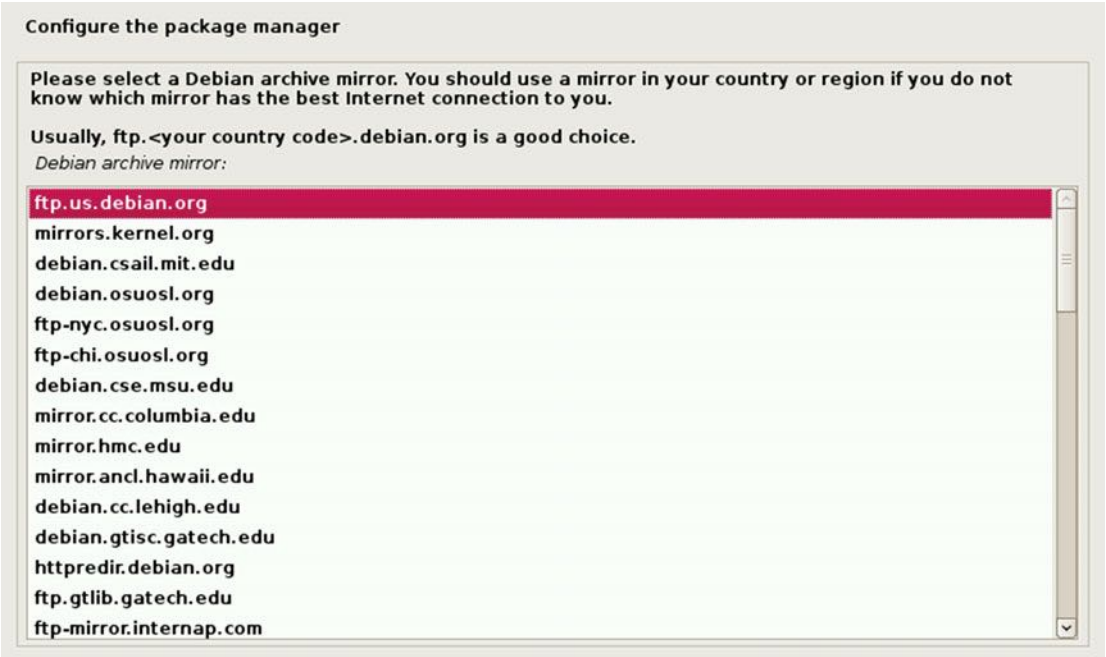


Figure 6-31. *Choosing an archive mirror*

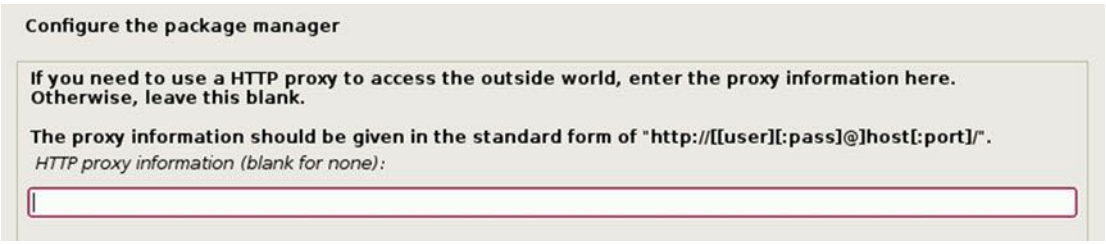


Figure 6-32. *Configuring an HTTP Proxy*

The next step (Figure 6-33) questions if you want to send information (anonymously) from the packages that you are going to install on your system. This is up to you, but it's very helpful to Debian to be able to collect data to make usage statistics.

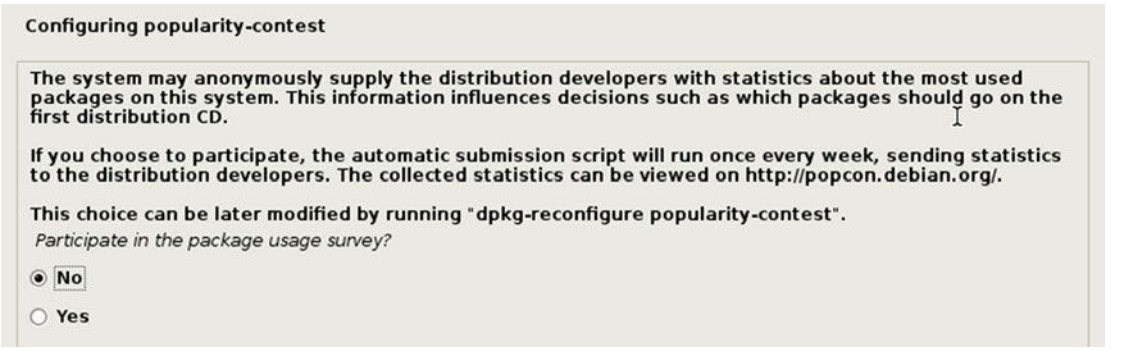


Figure 6-33. *Participating in the package usage survey*

Now you must select what predefined collections of software you want to install (Figure 6-34). Here you see mainly the desktop environments that Debian supports and a few more things, but if you had scanned another CD/DVD you would see more collections. Although the de facto DE of Debian is Gnome, I want to choose one that you won't see in any other distro installation: LXDE. So choose it and leave the other options selected by default. If you choose the SSH server, you better know how to secure it because it is one of the most frequent vectors of attack in a Linux system.

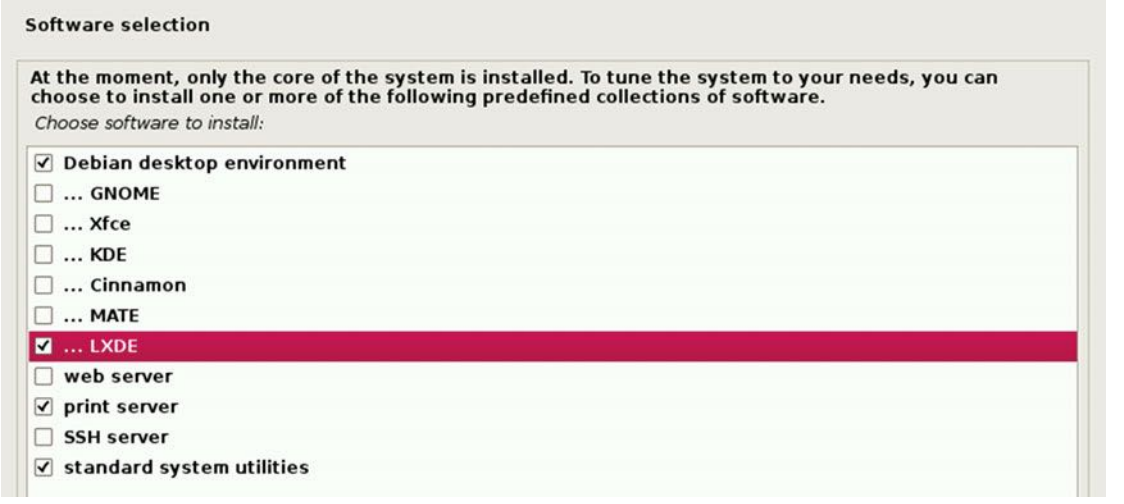


Figure 6-34. *Selecting the software to install*

Once you select the software, the longest step (Figure 6-35) of the installation process begins, the installation (and download) of the packages to your SSD drive.

Figure 6-35. *Installing the software*

Finally, you have to decide (Figure 6-36) whether to install the boot loader (Grub, in this case) on your hard drive to manage the boot up of your system. In this case, pick Yes and continue.

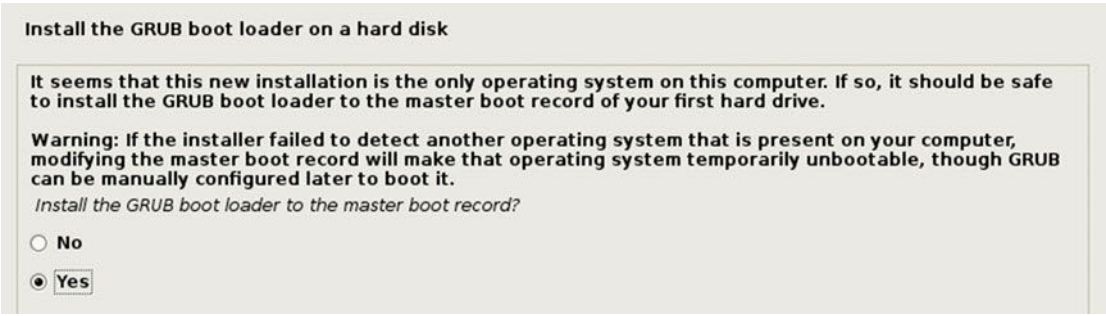


Figure 6-36. *Deciding to install Grub*

The final decision to make is upon which disk you want to install Grub (Figure 6-37). You must install it on the drive that has the bootable partition (`/boot`, in this case), which is the SSD one. I told you before to write down the references of the disks (sda and sdb) because it was going to be helpful later; well, that's now. You can see that the only references to identify both drives are the UUID (Universal Unique Identifier) and the device path (e.g. `/dev/sda`). You could figure it out by the order, but it's always better to know for sure, isn't it? Well, your SSD is the `/dev/sda` device, so choose it and continue.

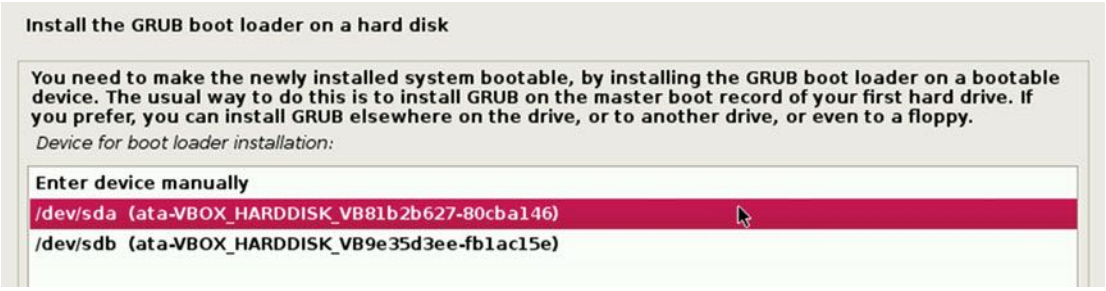


Figure 6-37. *Choosing the disk upon which to install Grub*

That's it! Your Debian OS is installed (Figure 6-38). It wasn't that hard, right? You must reboot your system now.

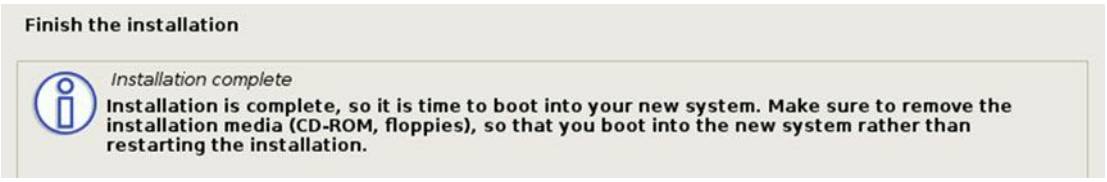


Figure 6-38. *The installation is finished.*

The first screen that you see when you boot up your system for the first time is the already familiar Grub screen, which you saw in the Fedora install (Figure 6-39). Press Enter or wait to boot your new Debian OS.

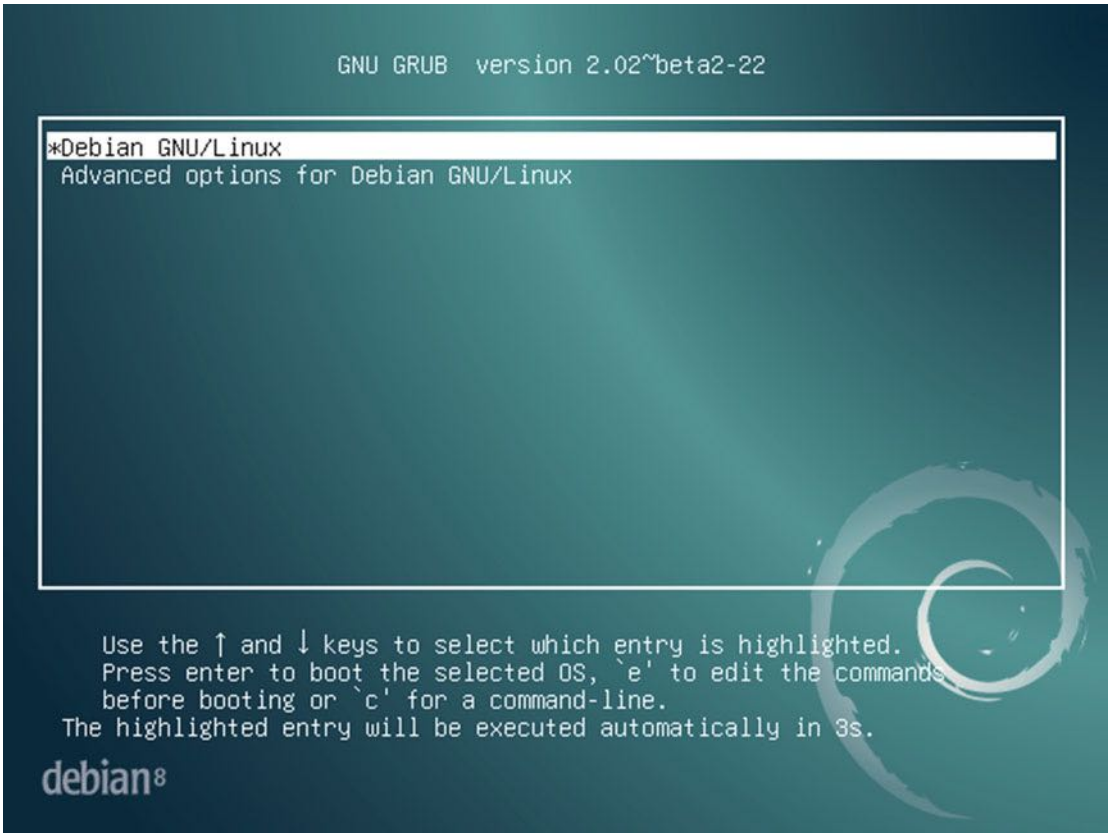


Figure 6-39. *First boot screen*

Figure 6-40 shows the login manager to log into your session. Introduce your user name and password, and press the Log in button.

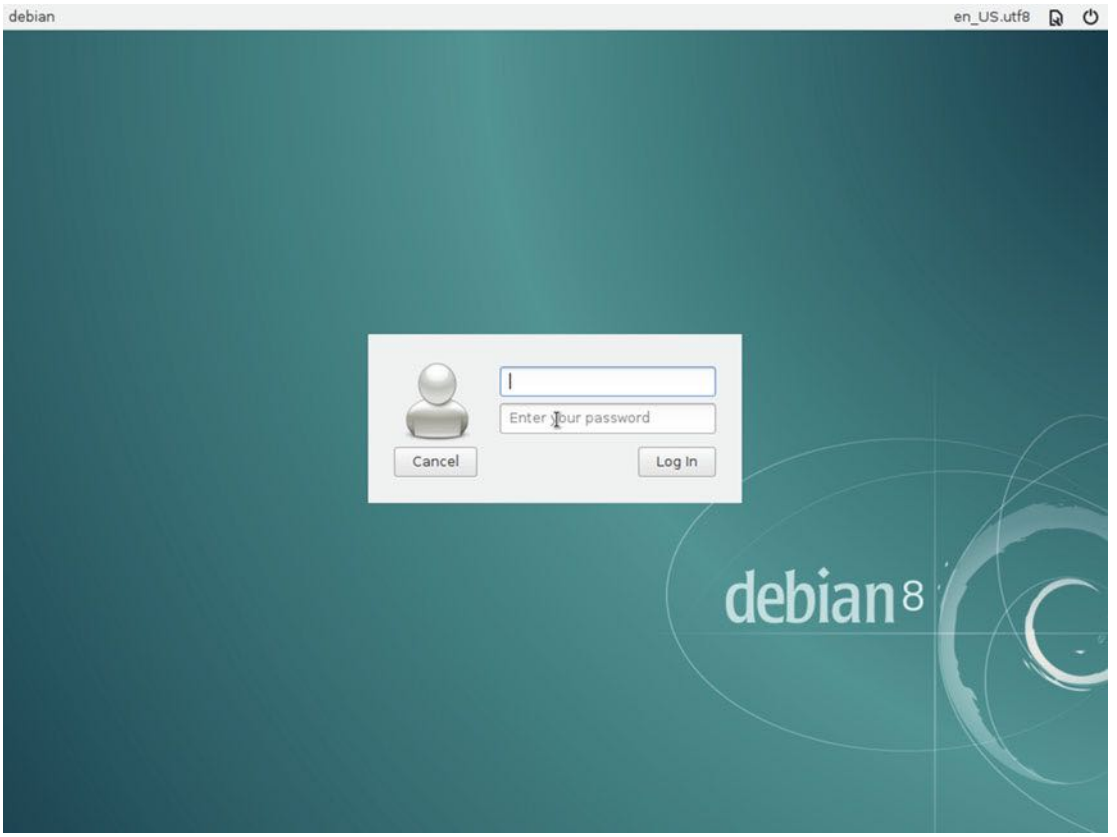


Figure 6-40. *First login in Debian*

You are logged now in the LXDE desktop environment. The first thing that you can see in the middle of the screen is a pop-up dialog (Figure 6-41) that gives you advice about ClipIt and the security risks of saving that information in plain text. It's up to you; I don't even use ClipIt, so I don't recommend activating it. (I use a clipboard without history, with only the current data in it.)

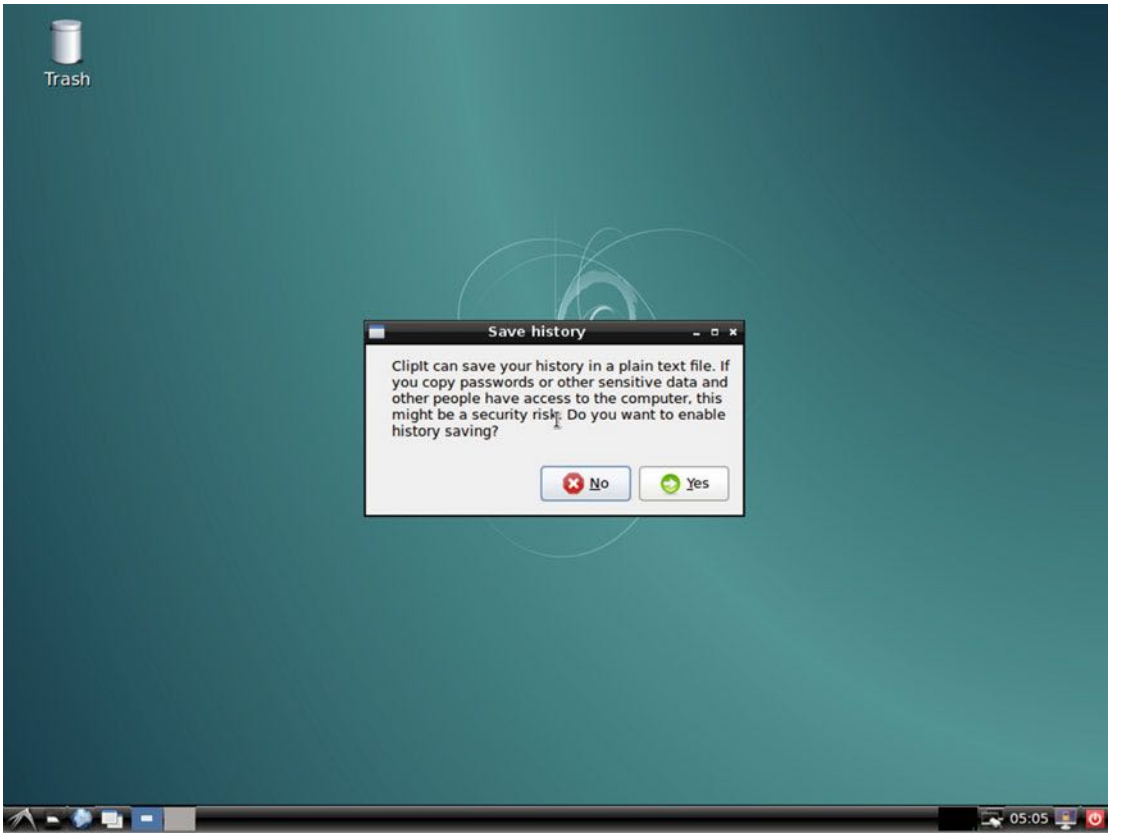


Figure 6-41. *First screen of LXDE*

I told you before that mounting the data partition would have a little inconvenience/annoyance, and this is it. The partition is created with the root user permissions, and you are not allowed to write anything in there (and therefore you must create a new file, as you can see in Figure 6-42). But this is easily fixable.

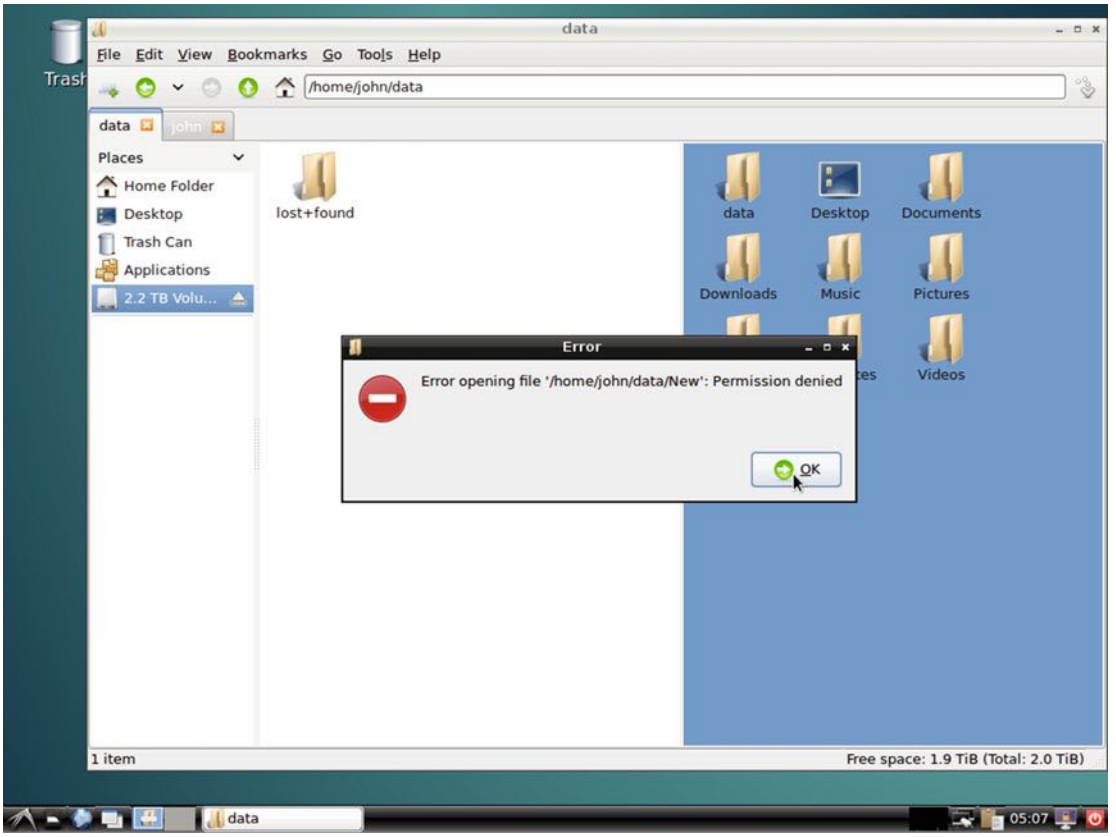


Figure 6-42. *Wrong permissions in data partition*

You have to change the permissions for the mounted partition to the user that you created. You can see how to do this in Figure 6-43, but I repeat it here where it's easier to read.

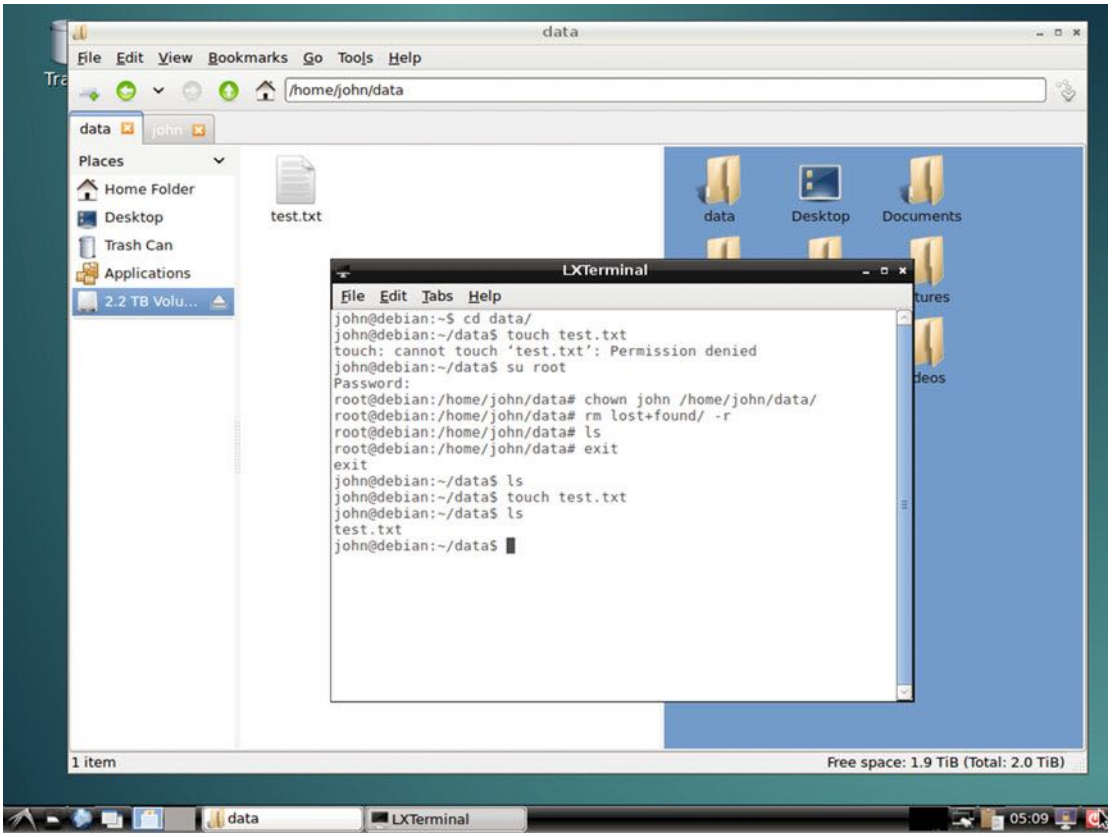


Figure 6-43. *Permissions in data partition fixed*

If you gave a password to the root user, you have to do it like this:

```
$ su root
Password:                       # enter the root password
# chown john /home/john/data/   # user your username instead of john
# rm lost+found/ -r             # delete the lost+found folder
# exit                          # never let the root user log in
$
```

If you left the root password blank, you have to use the sudo command:

```
$ sudo chown john /home/john/data/  # user your username instead of john
Password:                           # enter your user password
$ sudo rm lost+found/ -r            # delete the lost+found folder
$
```

And that's it! Your new Linux Debian OS is ready to use! Enjoy it!
You can read an installation guide at www.debian.org/releases/stable/amd64/index.html.en.

Maintenance

Again, the three essential tasks are updating, installing/deleting apps, and upgrading. Like Fedora, you must deal with the command line, like it or not, and be familiar with the essential apps to do the job.

Managing Apps

To install/delete applications you can use a graphical application, as is the case with Ubuntu/Fedora, like Synaptic. But almost every Debian user users the command line to do this. You can use the typical `apt` tools or you can use the friendlier `aptitude`. For example, if you want to install an app (such as gnome-chess), you should use this command:

```
# aptitude install gnome-chess
```

To remove it:

```
# aptitude remove gnome-chess
```

It's as simple as that, and you have a level of control over what you're doing that is superior to what you get with the graphical application (it's hard to see from a simple example like this, but it's true).

Updating

The same happens with the package updating process: there's no easy way. You can use Synaptic, but the best way to do it is to use `aptitude`. To do so, you only need to type this:

```
# aptitude update
```

That's it!

■ **Note** Note that the DVD that you used to install the distro is still considered as a source of packages, so whenever you use any package command, Debian will search the DVD for the packages first. If you have an Internet connection, you don't want this to happen. You always want the freshest packages, so you must fix this behavior. Again, Debian does not provide an easy way to do this (well, that's not entirely true; you can do this with the Synaptic application in a graphical way), so you must edit a plain text configuration file. Here I use the graphical text editor that comes with LXDE (I would use Vi(m) or Emacs, but that's another story) to edit the file:

```
# leafpad /etc/apt/sources.list   # open the file with Leafpad
```

And now you must locate two lines at the beginning of the file that begin with something similar to this:

```
# deb cd rom:
deb cd rom:
```

The only thing you need to do is comment the second line by preceding the line with the "#" character, like the first line. Save the file, close Leafpad, and that's it.

Upgrading

To upgrade the distro, you must update the currently installed packages. You should also make a backup of your data first, because this is a critical process. Let's see how it's done:

```
# apt-get update         # update the current packates using apt tools
# apt-get upgrade        # a first minimal upgrade to avoid conflicts later
# apt-get dist-upgrade   # the full upgrade of the distribution
```

This is enough in desktop installations and in a majority of cases. If this is a server or another critical machine, there are additional steps that you should take, but that's a story for another book.

Pros and Cons

The following is a list of some things that I personally see as pros and cons of the Debian distro. Of course there's always room for discussion in this matter, but I've done my best to be as objective as possible.

Pros

- The *Debian Manifesto*, the *Debian Social Contract*, the *Debian Constitution*, plus its long longevity and big community are guarantees that Debian is going to be among us for a long, long time.
- The stability of the Stable branch is one of the best of all the Linux distros; it's always a bit outdated but stable as a rock.
- Although there's no company behind Debian, its community is one of the friendliest and most willing when it comes to helping its users, so you will have very good support.
- The size of its community.
- The big number of available packages.
- It uses one of the most extended and best package managers: dpkg.
- It is an original distro, not a derivative.
- It is widely adopted by a lot of professionals and organizations.
- It supports many architectures and even various kernels.

Cons

- There is no regular/fixed cycle of releases for the Stable branch, so you could wait for a long time for the next one.
- Aesthetics and design are greatly ignored in this distro.
- It has no official commercial support.
- You must know how to use the terminal to perform admin tasks.
- The installer program is unfriendly and outdated.
- It is not the easiest distro to manage and configure. You have to use the terminal too much. It's for advanced users or those willing to learn.
- The Stable branch has outdated packages even when it is released.

Summary

With the analysis of Debian, you have the third example of a Linux distro. As you could see, Debian has many points in common with Fedora and Ubuntu, but there are clear differences that make Debian unique.

The next chapter on openSUSE will introduce you to another way of doing things in Linux.

CHAPTER 7

openSUSE

openSUSE, as its name implies, is an open distro and the base of the SUSE Linux Enterprise commercial one. SUSE was the top European distro and the alternative to Red Hat, at least until the arrival of Ubuntu. SUSE adopted Red Hat's idea of having a community version of its commercial distro, and openSUSE received a good reception from the community, so good that it eventually displaced SUSE as the chosen distro for regular users. The SUSE family of distros have always been very well maintained, designed, and finished products. openSUSE was for years one of the few distros that was sold in a box, accompanied by a large printed manual; you can still purchase openSUSE in this manner in Germany (only in the German language, however).

History

Although SUSE started as a German company in 1992, the openSUSE project didn't start until 2004. SUSE was purchased the American company Novell, and openSUSE was released in October of 2005. In 2011, Novell was acquired by The Attachmate Group (which merged with the British firm Micro Focus in 2014); it decided to make SUSE an independent company with its headquarters back in Germany, specifically Nuremburg. As a result, Germany and United States are the main zones where openSUSE is very popular; these locations account for almost half of the installation base.

Philosophy

openSUSE is the community version of SUSE's commercial distro, similar to Fedora/Red Hat. openSUSE, like Fedora, is also a kind of test laboratory for new ideas and technologies. Also, both have a strict policy of only using open and free software in its official repositories/installation images. As for differences, openSUSE uses KDE as its default desktop environment, while Fedora uses Gnome. Fedora offers better security and innovation, while openSUSE provides advanced tools and services like OBS, openQA, and SUSE Studio to its community. In other words, both distros have the same initial purpose, but their results differ mainly due to the influence of their mother distros and community (in size, the Fedora community is larger than the openSUSE one). You can read about openSUSE's guiding principles at `https://en.opensuse.org/openSUSE:Guiding_principles`.

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_7

Distro Selection Criteria

Now that you know a little history of openSUSE, let's see how this particular distro fares on the selection criteria from Chapter 2.

Purpose and Environment

openSUSE is a general purpose distribution, but there are a few unofficial versions developed by the community that are task-oriented ones[1]. Like Debian, there is no special version for server purposes (SUSE has one). It offers only two official versions of the distro:

- **Leap**: The regular release version of the distro and the main one. Leap is the stable version.
- **Tumbleweed**: A rolling release version of the distro, with frequent updates of the newest packages, this is a bleeding-edge version.

Also openSUSE (more specifically, SUSE) offers a unique service, SUSE studio, which allows you to create your own derivative or customized image of openSUSE. It is a very powerful tool with advanced features that lets you create images for a DVD, virtual machines, the cloud, etc. You even can deploy your images to the cloud directly or test the result online. You can find SUSE Studio at `https://susestudio.com/`.

Support

Similar to Fedora, openSUSE only offers community support, even though SUSE is behind this distro as a sponsor. The community is far from the size of Ubuntu's or Debian's, but, as with Fedora, it is big enough to offer good support. As with the majority of distros, you can access this help through numerous channels:

- **Documentation**: `https://en.opensuse.org/Portal:Documentation`
- **Wiki**: `https://en.opensuse.org/Main_Page`
- **Forum**: `https://forums.opensuse.org/forum.php`
- **Mailing Lists**: `http://lists.opensuse.org/`
- **IRC**: `https://en.opensuse.org/openSUSE:IRC_list`

User Friendliness

You can see in openSUSE the influence of its mother distro's orientation (to the corporate world) by how many advanced settings you can configure in your OS via graphical interfaces (and in the installation process). Still, it's reasonably easy to use.

It claims that its unique configuration and installation tool, YaST (Yet another Setup Tool), is the best and easiest to use in the Linux ecosystem. The rest of the usability of the OS is due to the default desktop manager utilities. I personally think that YaST is a very good tool, but nothing replaces the command line when you want to manage your OS. In comparison, it is comparable to Fedora (maybe a little easier); it's less easy than Ubuntu, but more than Debian. Moreover, YaST is a very powerful tool but because of this, it's perhaps is not the easiest one for beginners. Thus, I don't recommend this distro for beginners; they may get overwhelmed by so many advanced options.

[1] `https://en.opensuse.org/Derivatives`

Stability

openSUSE is a very stable distribution. It also has a unique (at least as a public service) characteristic: an automated testing tool to test the build of packages, the installation process, and several features of the OS like GUI actions or if a program works as expected. With this tool, openQA (https://openqa.opensuse.org/), they can test several combinations of hardware and installation options automatically each time a new build is ready (the builds are made through another automated open tool, openSUSE Build Service). Of course it is impossible to cover all possible errors and fails with tests, but it helps greatly to build a very stable release. This tool is used for both versions of the distribution, the regular one (Leap) and the rolling release one (Tumbleweed).

The release cycle of the Leap version, the one that follows the standard release model, is intended to be a year for each minor release and three years for a major release. The support life of each version is around 18 months after the release date for each minor release and three years for a major release. Also, as of the current release, openSUSE is based on SUSE Linux Enterprise (SLES).

Also, the community selects several openSUSE versions to support (by the community) for a long time (not a fixed period) through the Evergreen program. If you want to know more about openSUSE releases, go to https://en.opensuse.org/Lifetime.

Hardware Support

In terms of hardware support, openSUSE is in a similar place as Fedora. Because of the strict policy of not including private drivers, hardware support is not as good as that of Ubuntu by default. This can be solved partially thanks to several community repositories and the use of the rpm package system (many drivers are in this format).

Aesthetics

Like Fedora, the aesthetics of openSUSE are the default ones of the desktop environment chosen, except for the desktop background, colors, and the installer (YaST).

Desktop Environment

Traditionally SUSE (and openSUSE) have always offered the two main desktop environments (Gnome and KDE) as an option in the installation process, but KDE is the default/official one. openSUSE also offers the option to choose XFCE or LXDE in the installation process.

Init System

The init system of openSUSE is the most common these days among all the distributions: systemd. openSUSE was the third big distro to adopt systemd, after Fedora (its creator) and Mageia.

Package Management System

The package management system is the same as Fedora: rpm. Thus, it benefits from the wide distribution of this package system from third party developers. To manage the packages, it includes graphical and command line tools: YaST and zipper.

Architecture

Until very recently, openSUSE supported two hardware architectures, 32 and 64 bits, but since the current release, Leap 42, only the Intel/AMD 64-bit architecture is supported. This is a decision that many distros are looking to make soon; Fedora and Ubuntu make this change in the near future.

Security/Anonymity

openSUSE is a secure distro. Like Ubuntu, it enables AppArmor by default and a firewall. It is less secure than Fedora, but through YaST you can customize AppArmor, the firewall, and configure other options to harden the OS. Also, you can encrypt your partitions or your home folder.

Principles and Ethics

Like Fedora, openSUSE provides only free and open source software by default, even drivers. But there are non-official community repositories where you can find software, which can easily be added through YaST. You can read more about openSUSE's policies at `https://en.opensuse.org/Restricted_formats`.

Live CD

openSUSE offers two Live DVD ISO images, one for each desktop manager (KDE and Gnome).

Professional Certification

Like Fedora, openSUSE does not offer any professional certification program, but its mother distribution, SUSE, does (detailed in Chapter 2).

Installation

The first thing you must do before installing openSUSE is choose between the regular release version and the rolling release version. Here I will show you how to install the distro using the Leap version (the standard release) because it is the main one and the most probable selection by the majority of users.

As in the previous distros, you must go to the project page to download the ISO image to use it to install the OS. If you go to `www.opensuse.org/`, you will see a screen like Figure 7-1.

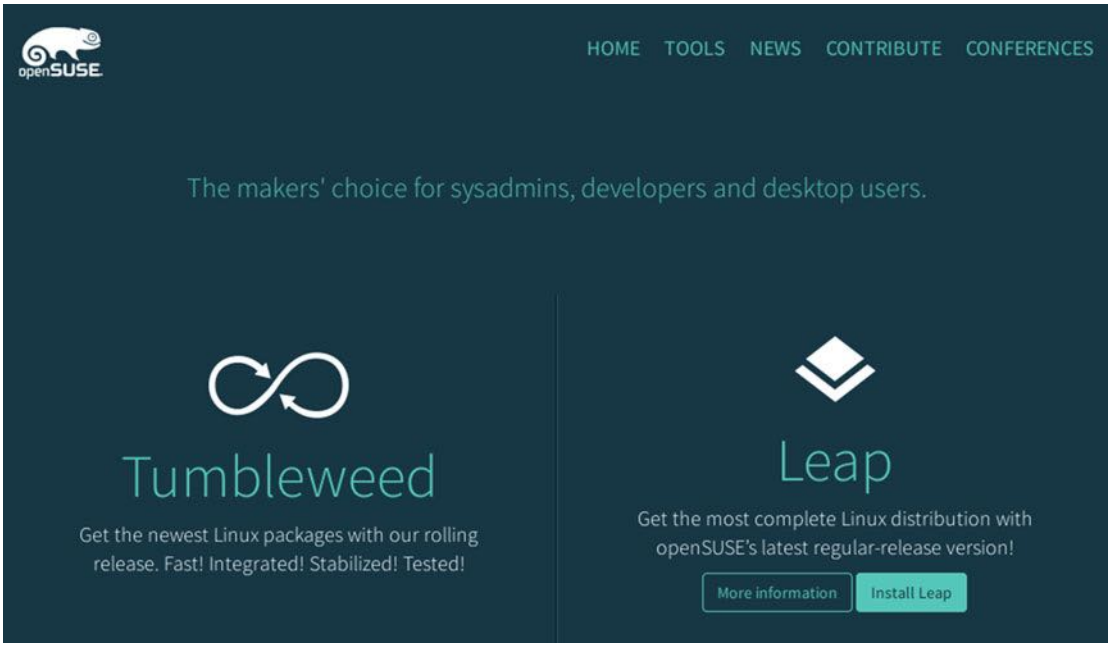


Figure 7-1. *The OpenSUSE project web page*

Click the Install Leap button for the Leap version. You will go to a screen like Figure 7-2, where you can directly access the ISO image that you need by pressing the Direct Link button. You can also choose another download method such as Bit Torrent.

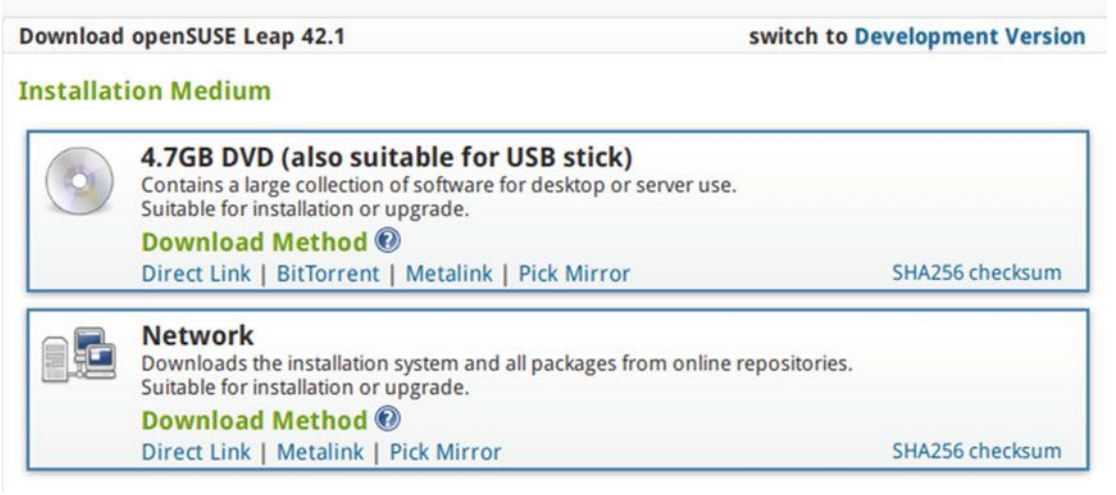


Figure 7-2. *The downloaded ISO image dialog*

This screen shows two interesting options:

- The *Switch to Development Version* allows you to download the current Beta or Development version of the distro.
- The *Network* version is the minimal installation base and it relies on an Internet connection to download all of the packages.

After you download the current ISO image of the Leap version and boot up your system, you will see a screen where you can see how openSUSE takes care of certain things like design (Figure 7-3).

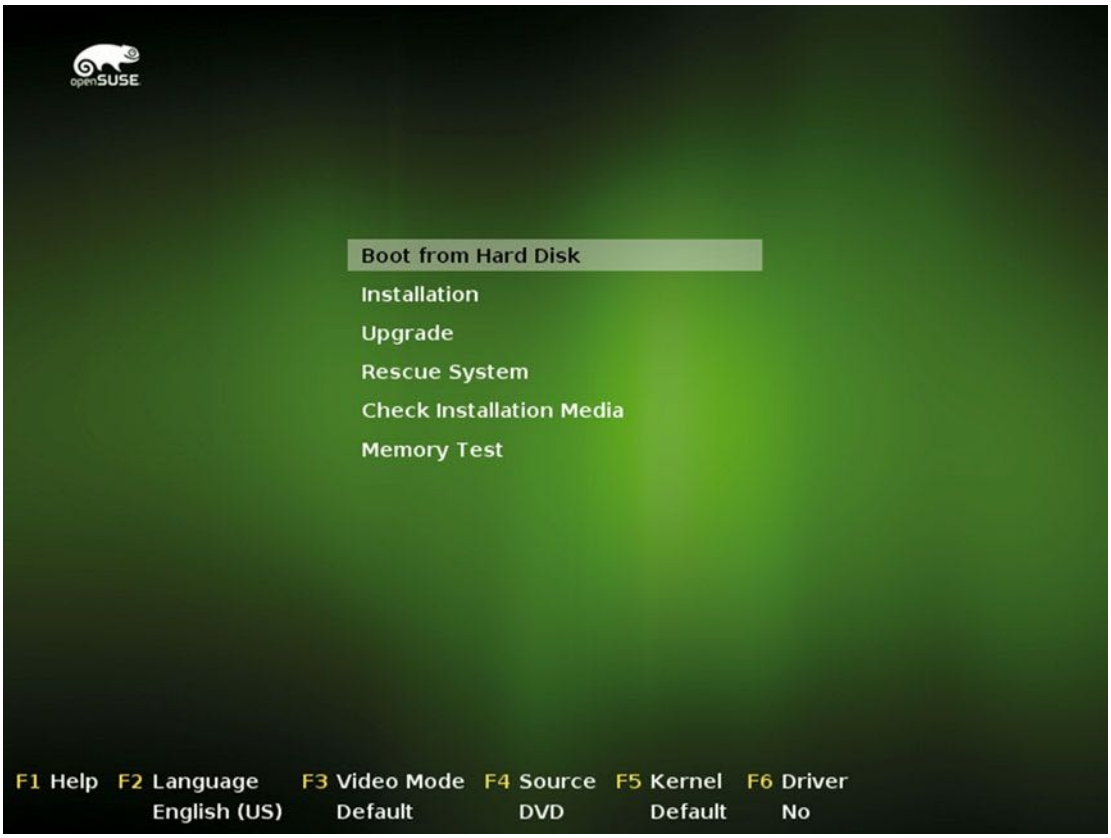


Figure 7-3. *The text installation menu screen of openSUSE*

In this classic screen, you can see (apart from the beautiful openSUSE logo) your options. By default, the "Boot from Hard Disk" menu entry is selected because of how YaST ends the installation, without soliciting or ejecting the DVD at the end. Thus in the first reboot after the first installation, openSUSE will boot directly to your new OS. The next menu entry is the one that interest us: "Installation" is going to start the installation process as soon as you choose it. "Upgrade" is intended to do an offline upgrade of a current openSUSE installation. The rest of the menu entries are the same as you saw in other distros.

Note the lines at the bottom of the screen. One line lists the usual advanced options; the line below it shows the current values selected (in this case, they are the default ones). It is worth mentioning that if you press Esc and then OK in this screen, you will be sent to the text version of this screen (the installer still is a graphical one).

So press the Installation option and then begin the installation on the first screen of YaST (Figure 7-4).

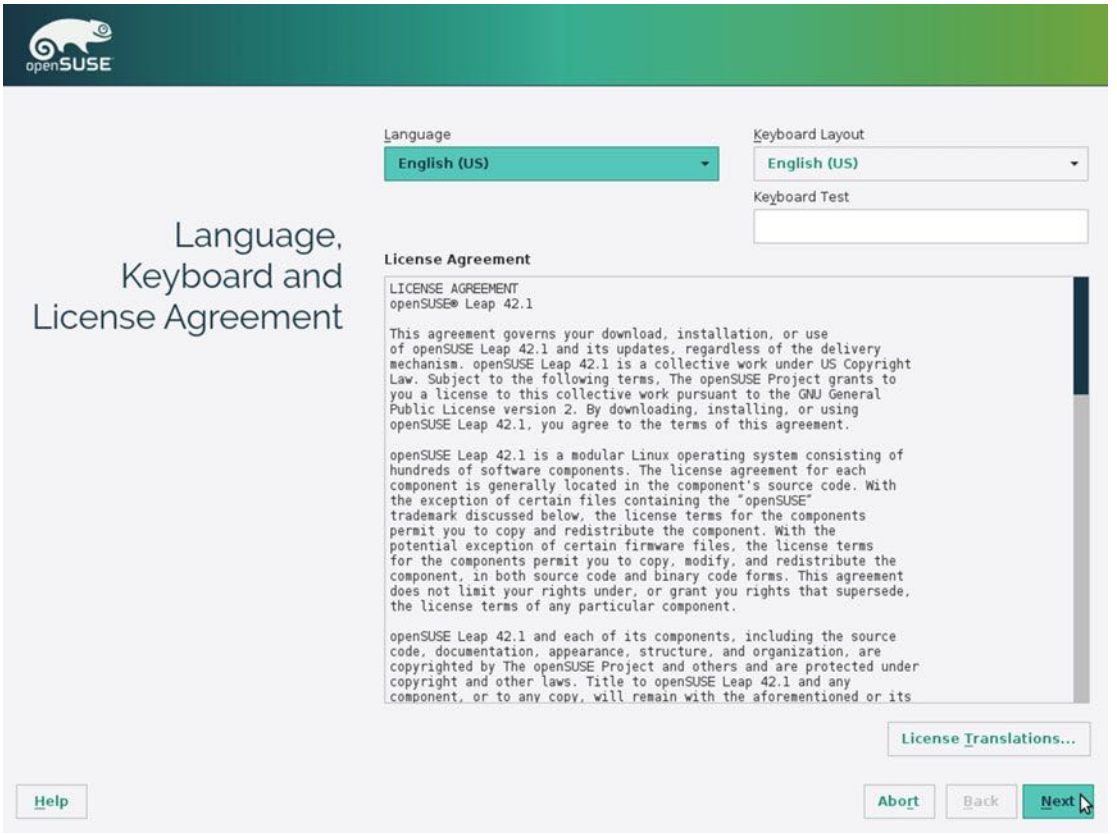


Figure 7-4. *The first screen of YaST is where you select the language, keyboard layout, and agree to the license*

On this first screen you see the traditional options to select a language and a keyboard layout, a Help button, and for the first time, a License Agreement. This is a peculiarity of openSUSE; it is very rare to see this in a Linux distribution, since this is Free Software under a GNU license and the majority of distros want to avoid this step to differentiate themselves from other OSes and their EULAs.

Select your language and keyboard layout, and press the Next button to jump to the next step, shown in Figure 7-5 (after a brief hardware system auto-detection process).

Figure 7-5. *Source installation options*

This screen offers the option to choose additional sources of packages for the installation, like the default repositories or other media (online or offline). If you have a good Internet connection, add the online repositories to get always the most recent versions of the packages. The next step is shown in Figure 7-6.

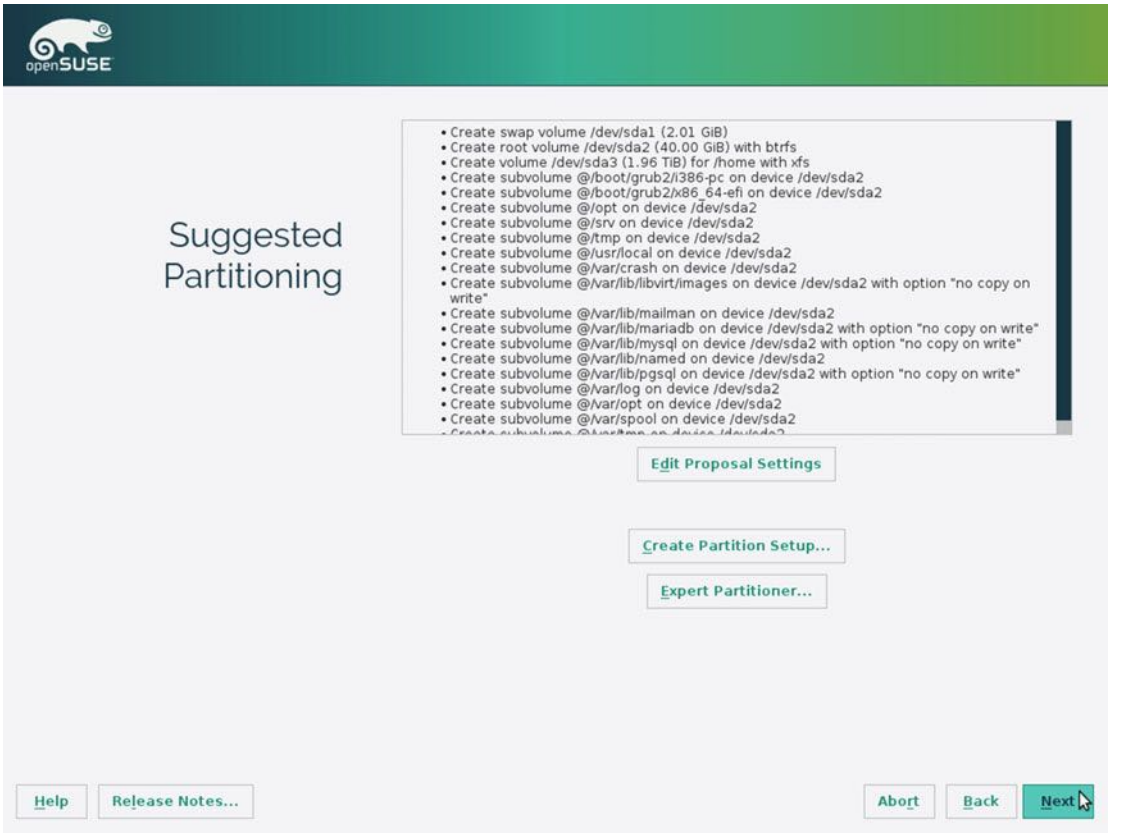


Figure 7-6. *The suggested partitioning by YaST*

This is the drive partitioning step. The suggested partition from openSUSE is far more complex than you've seen so far in this book. This is because it uses by default the btrfs file system in the root partition. openSUSE uses a tool called Snapper that takes snapshots (manual or automated) of that partition. This tool works with btrfs, ext4, and thin-provisioned LVM volumes. And the reason why xfs is used for the `/home` directory is because SUSE adopted this file system because it offers advantages like error handling and support of big volumes of data that suit corporate environments better. Plus, it creates a lot of subvolumes, making use of the advantages of btrfs to do so.

There are three ways to edit the settings that are proposed to you; the first two are semi-automated ones and the third is completely manual.

- **Edit Proposal Settings**: This allows you to make a few changes to the already proposed scheme, like opting to use LVM and encryption and the file systems of the `/` and `/home` partitions.
- **Create Partition Setup**: This is very similar to the previous option, but here you can choose the disk and partition/free space (which is useful in cases when other OSes are already installed) where you want to install the OS.
- **Expert Partitioner**: This is the completely customized manual procedure. Here you can see one of the best things about YaST; the partitioner program is one of the best and most powerful in all Linux distros. You can take a glance at it in Figures 7-7 and 7-8.

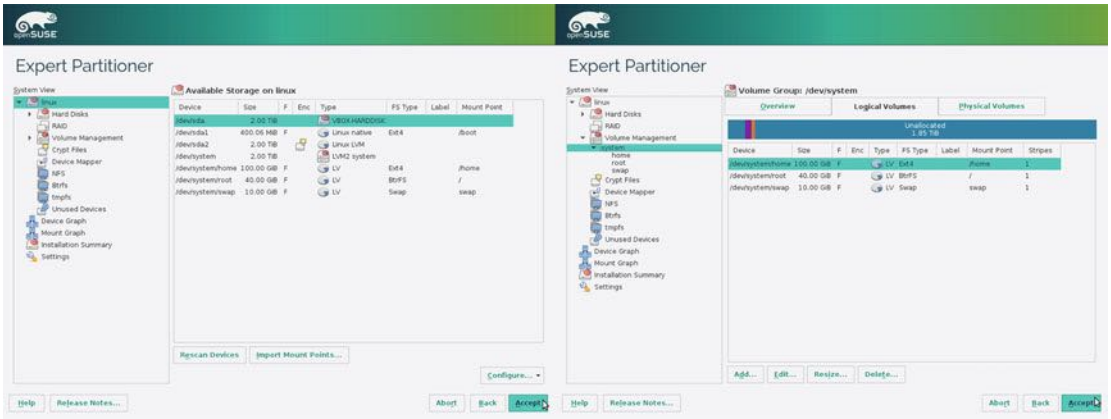


Figure 7-7. *The Expert Partitioner from YaST*

Figure 7-8. *Another view of the Expert Partitioner*

If you are an expert or want to completely customize your partitions, the last option is for you; otherwise the other two are fine. Instead of dealing with the Expert Partitioner, which I think that is an awesome tool, let's customize the default scheme a little by choosing the first button. If you press the Edit Proposal Settings button, you will see a dialog like the on the left in Figure 7-9. It's the default setup, but you're going to change it a little to see how small changes can create a perfect setup.

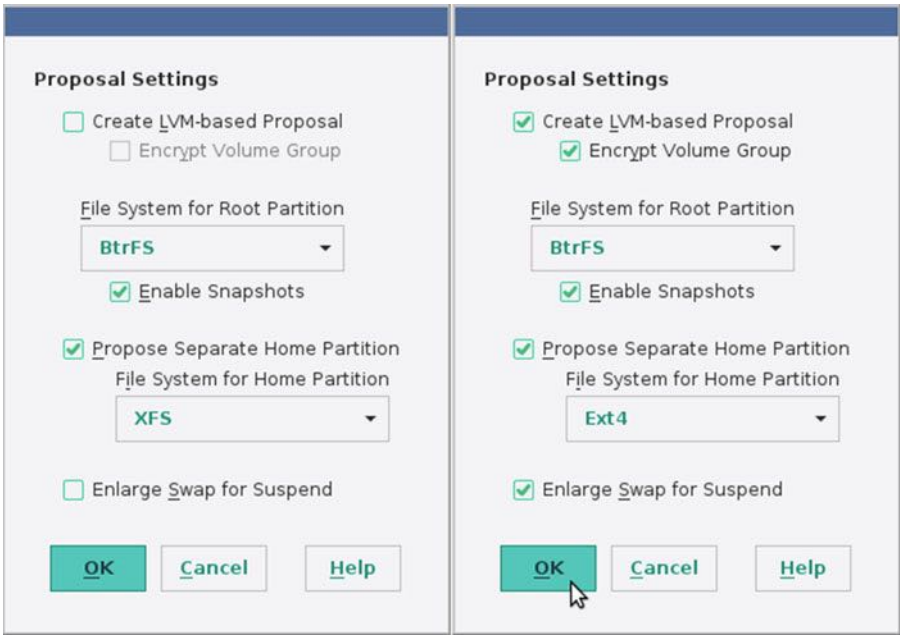


Figure 7-9. *Changing the partitioning proposal settings*

First, you want to enlarge the swap partition for suspend, so you can store the contents of your RAM memory on the disk when it is suspended. Next, change the `/home` partition file system to ext4 instead of XFS; I think this is the best choice for a home user. Finally, you are going to use a LVM partition (for the `/home` mount point) and encrypt it. When you select encryption, it will ask you for a password, and this is serious. If you forget this password, forget about your data: it's gone (you can't access it without that password). So, choose a good one and use a method that helps you to remember it (you can use the one that I showed you in Chapter 4). At the end you have a setup similar to the one shown on the right in Figure 7-9.

Press the OK button and you will see the changes reflected in the summary (Figure 7-10). Press the Next button to go to the next step.

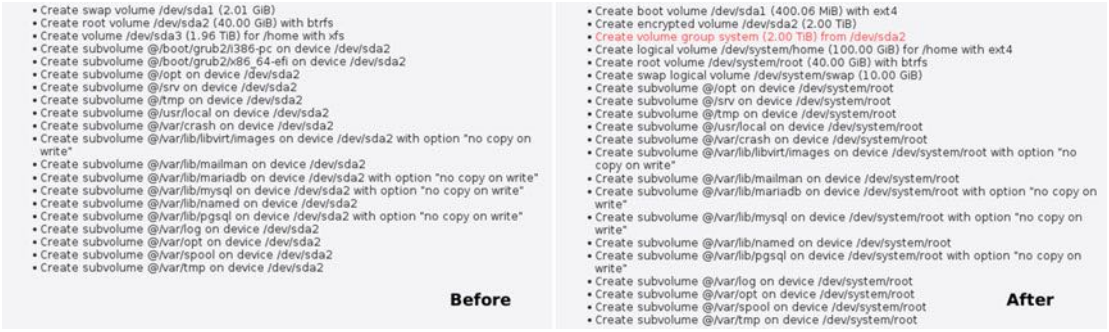


Figure 7-10. *The summary of partition actions before and after change the proposal settings*

This is the usual step where you configure your clock and your time zone (Figure 7-11). As with Fedora, you can also customize advanced options like synchronization with an NTP server via the Other Settings button. You should know already how to complete this step. When finished, press the Next button again.

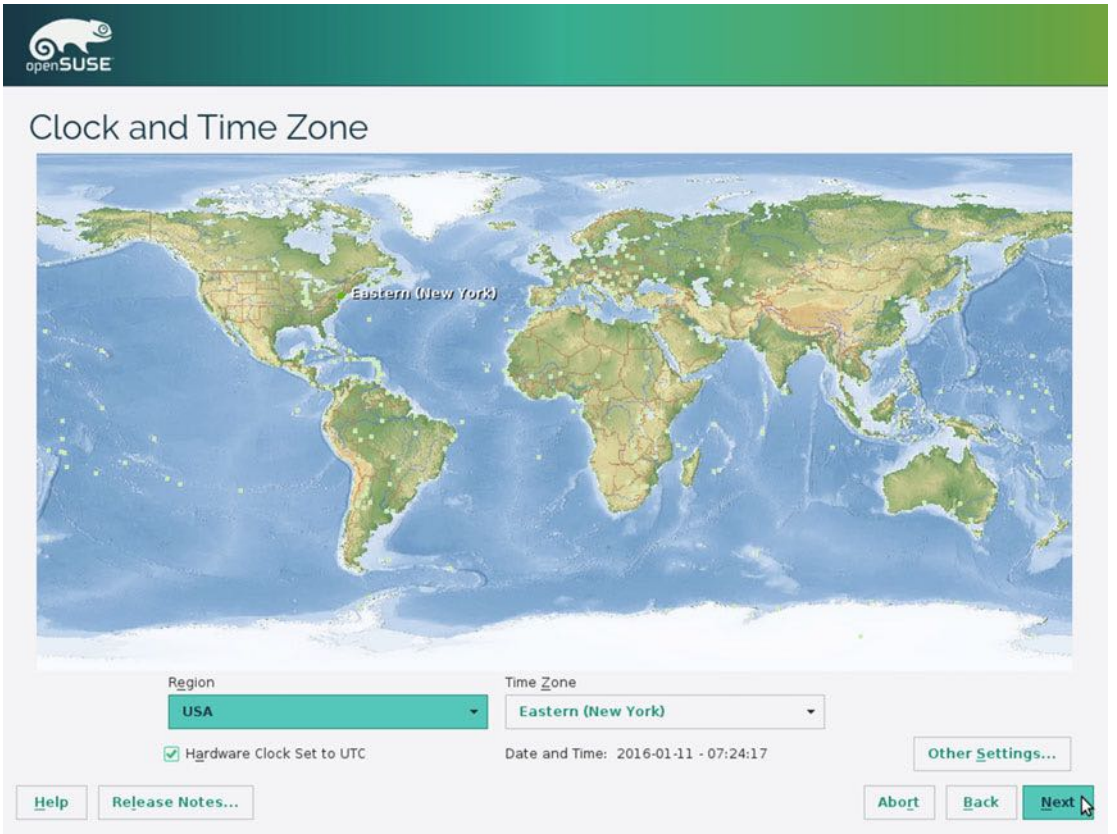


Figure 7-11. *The clock and time zone settings*

If in the first step you added the online repositories as suggested, now you must select which ones you want to use here to do the installation (Figure 7-12). I suggest that you choose the first four, which are selected by default. If you are not going to compile any packages, you don't need the sources (or read the source code), and the debug ones are intended for developers of openSUSE. Select them and go to the next step.

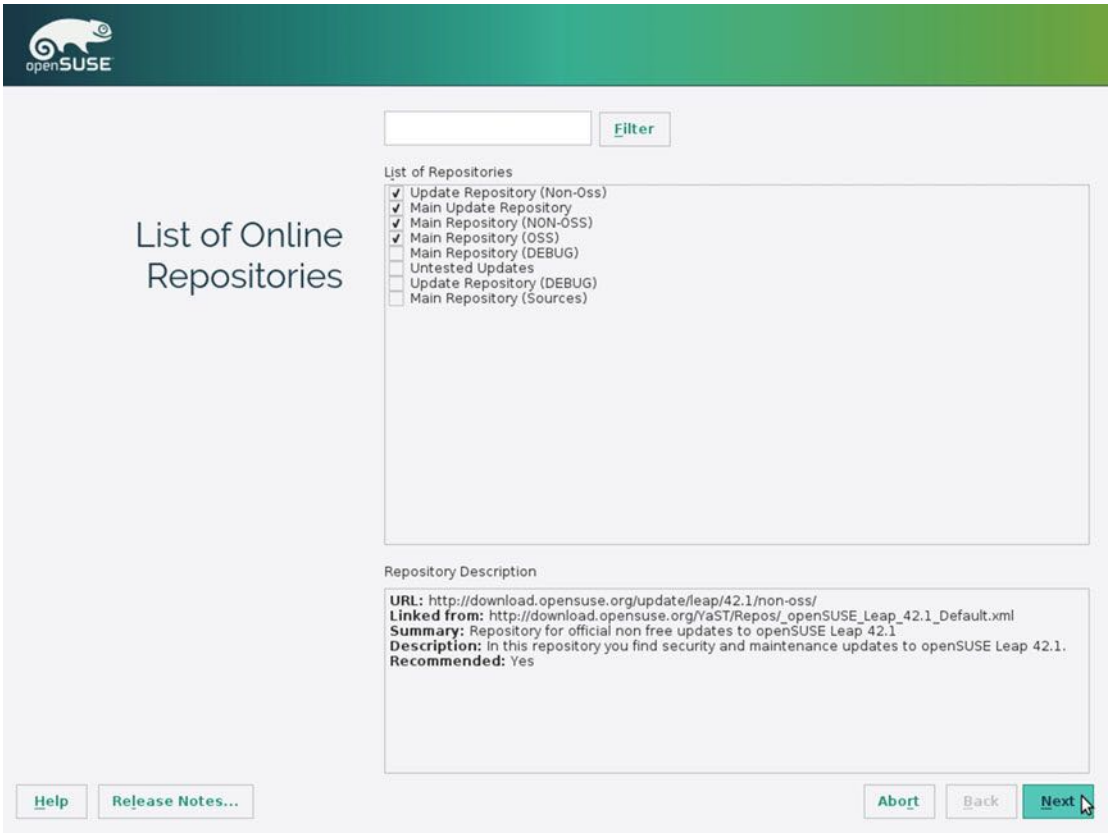


Figure 7-12. *Selection of online repositories*

After a few moments, during which it reads the contents of the online repositories, it will jump to another EULA license agreement, as you can see in Figure 7-13. I understand this appears because the contents of the main repository may have changed from the ones on the DVD, but I suspect that you will find this as annoying as I do. Regardless, it's due to the philosophy of the distro, so press the Agree button and continue with the process.

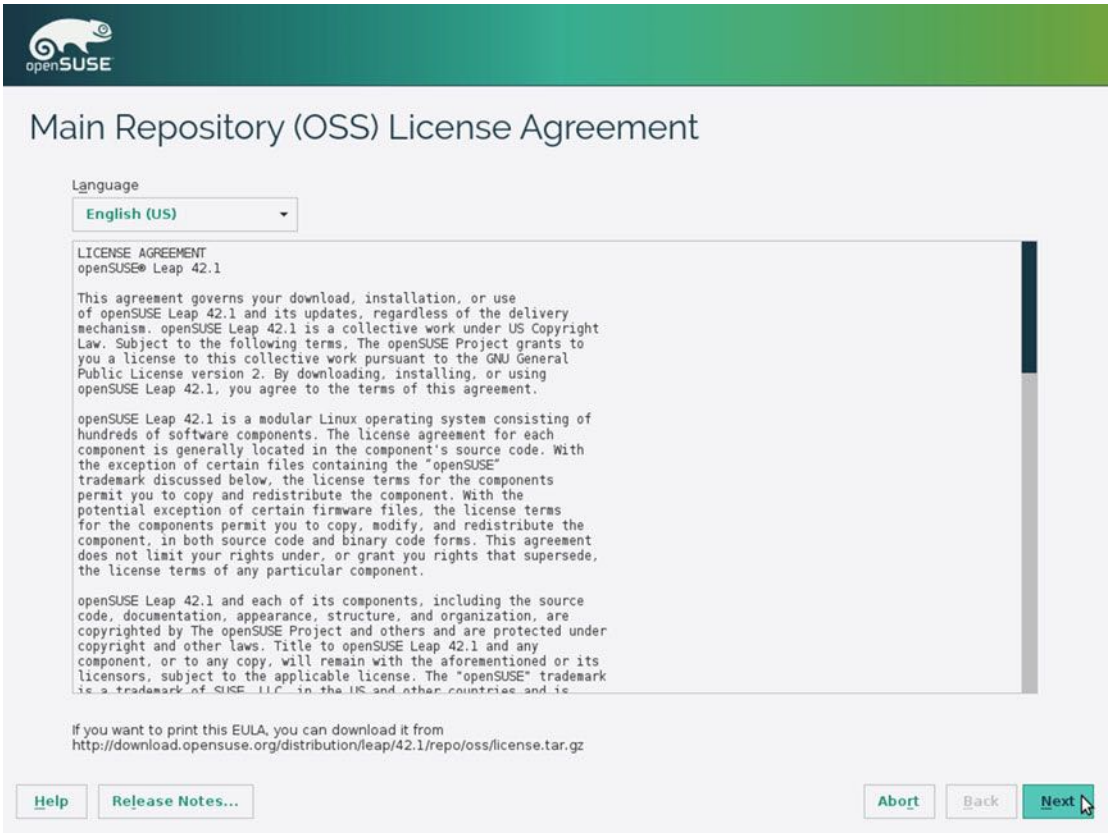


Figure 7-13. *Another license agreement screen*

After adding the repositories, you get to choose the desktop environment (Figure 7-14) that you want to use for the installation. The default one is KDE (the openSUSE tools are written for this environment; if you use another, it will install the minimal components of KDE, Qt, to be able to use it), but you can choose Gnome or other. Let's select KDE here. Now for the last steps!

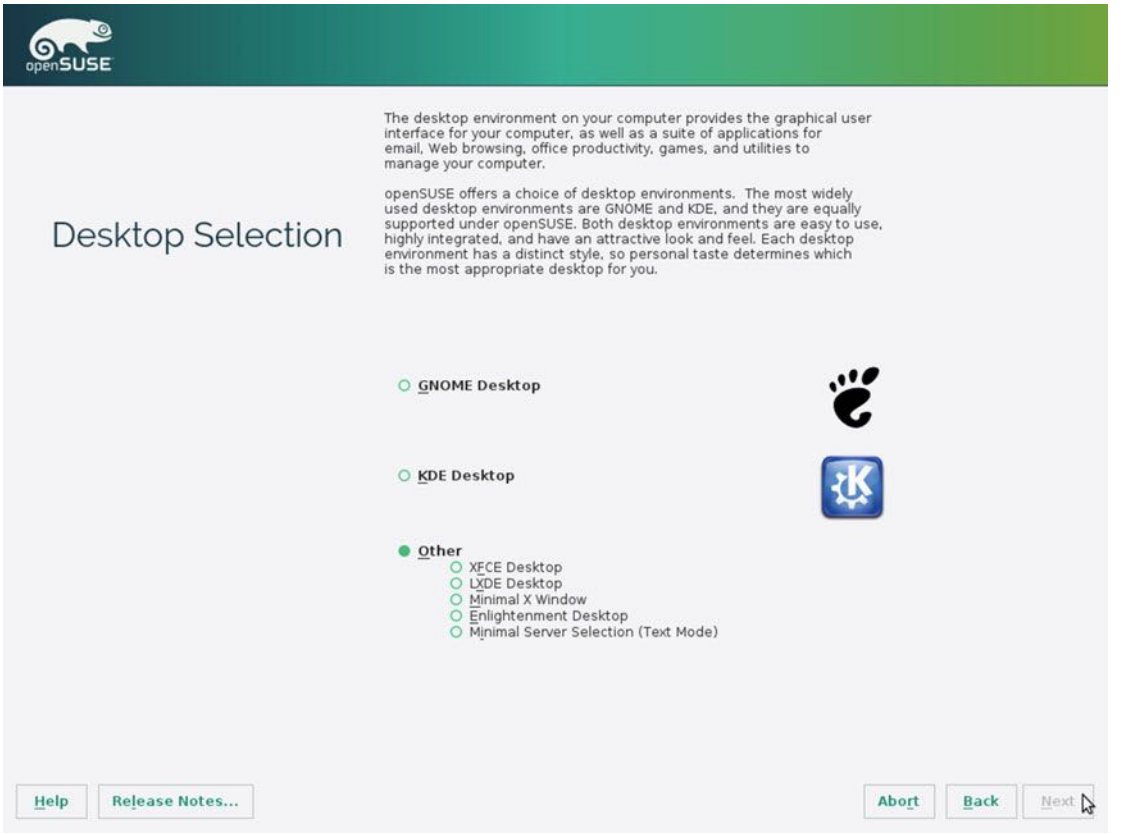


Figure 7-14. *The desktop environment selection screen*

As you can see in Figure 7-15, now you must create the user that you are going to use in the OS, as usual. As in Fedora, there are a series of advanced options. The first one is "Receive System Mail," to be able to receive the mail destined to the root user into this account. Also, you can change the authentication method with the Change button. The default options in this case are the most secures ones, so I suggest that you don't change them. The option "Use this password for systemd administrator" is exactly that it says; don't confuse this with the sudo command. So fill in the fields with your name and the password (please use a strong one) and then you can continue.

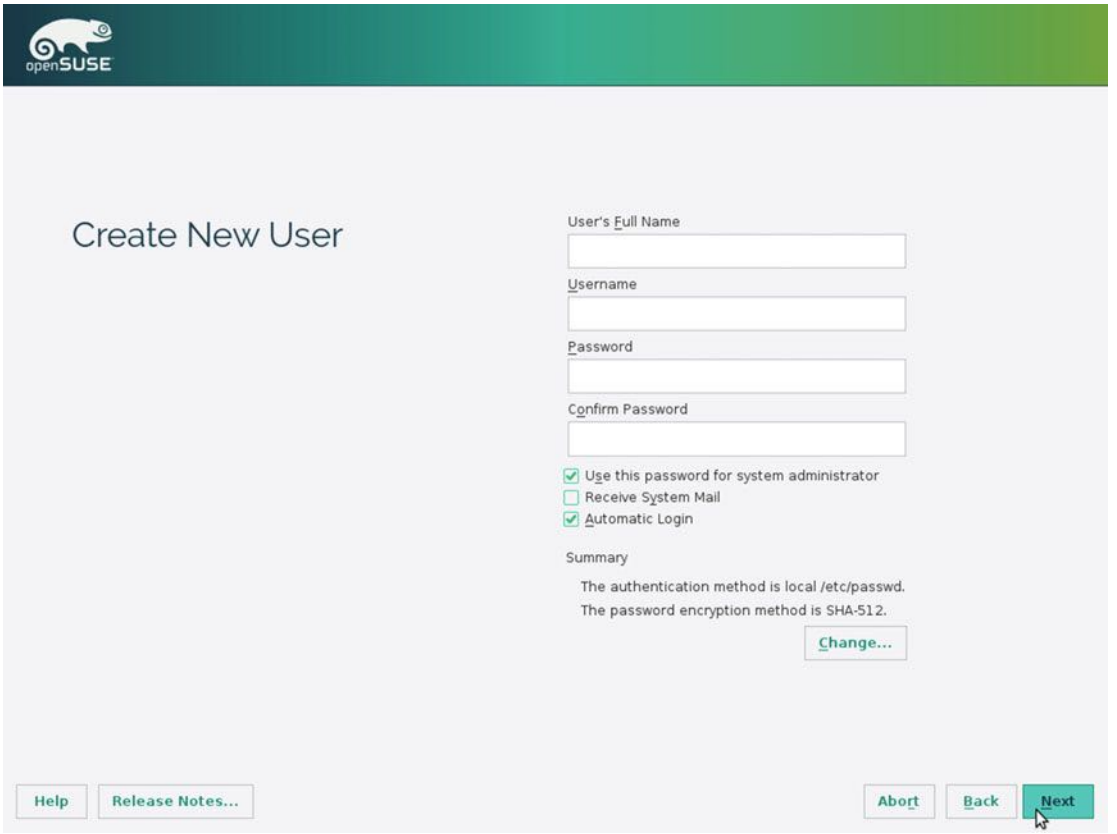


Figure 7-15. *The user creation screen*

■ **Note** I'm using a virtual machine manager, VirtualBox, to install the distros that I'm showing you in this book, for two main reasons. The first one is convenience; I can create a new machine or delete it in a matter of seconds (so I can have several machines for the same distro with different scenarios). If I had to do that with a real machine, it would be impossibly tedious. The second reason is because it's much easier to take screenshots in this scenario.

In order to manage several machines and distros in a reasonable way, I'm using the same password in all of them. And from the beginning I intended to use a short and easy password, because it helps to make the task less annoying. Also, by using a weak password I can test how the different distros manage this situation. Until now, only Fedora considered my password as weak. But openSUSE was the first one (Figure 7-16) to advise me against using a weak password. It was also the first one to alert me that it was very weak because it was based on a dictionary word (and that is true). If you are curious, the password is `pas$wOr.d`. So never, ever use a password like this or you will be owned very fast. Kudos to openSUSE for taking care of things like this.

Figure 7-16. *The weak password alert dialog*

If you didn't previously check the option to use the same password for the root user (I hope not; it is a security weakness that can bite you in the future), then it will now ask you for the root password, as seen in Figure 7-17.

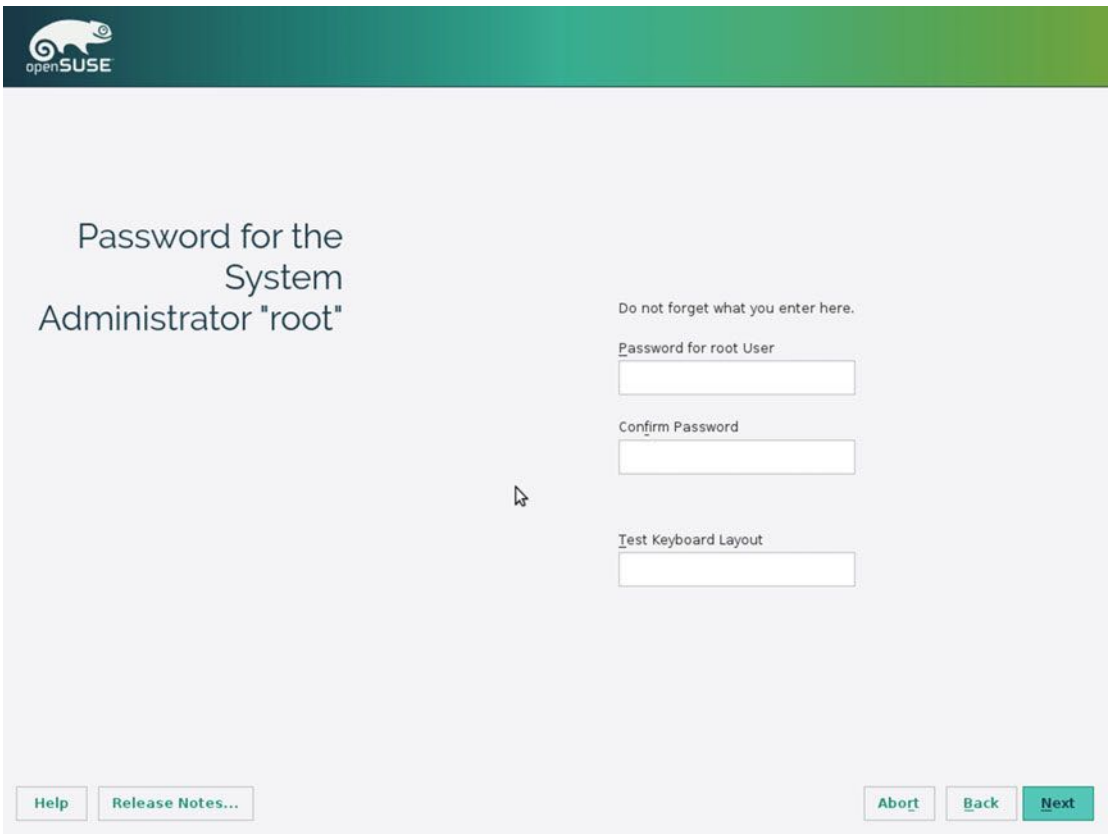


Figure 7-17. *The administrator password form*

Before beginning the installation, you will get a summary of the installation settings (Figure 7-18). Note one important point here—this is an interactive summary, so you can click the links (underlined in green) to access to some options and change them. For example, you can change the software that is going to be installed (Figure 7-19), an option that you didn't have before. Also, you can export this configuration to be able to do other installations exactly like this one in a complete unattended and automated way with AutoYaST. One thing worth mentioning is that you can see the size of the packages that you are going to download from Internet. If you don't have a fast connection and the size is too big, you can always go back and deselect the repositories.

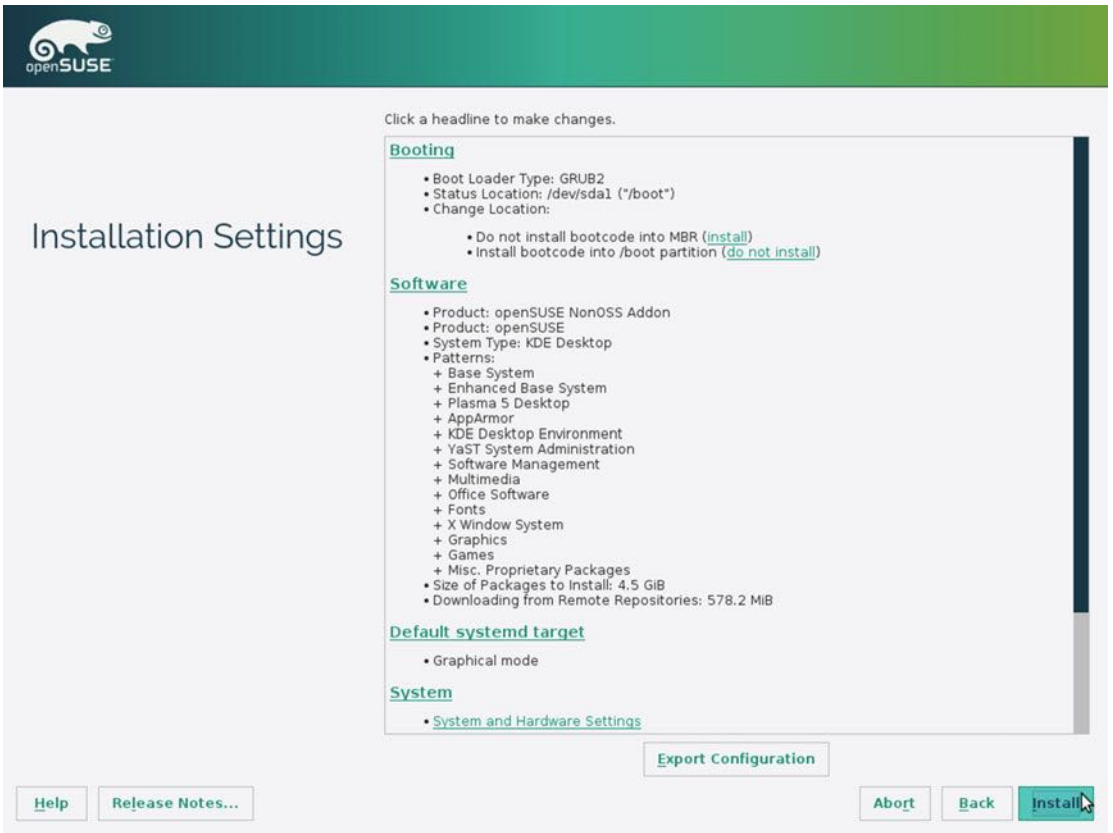


Figure 7-18. *Summary of the installation settings*

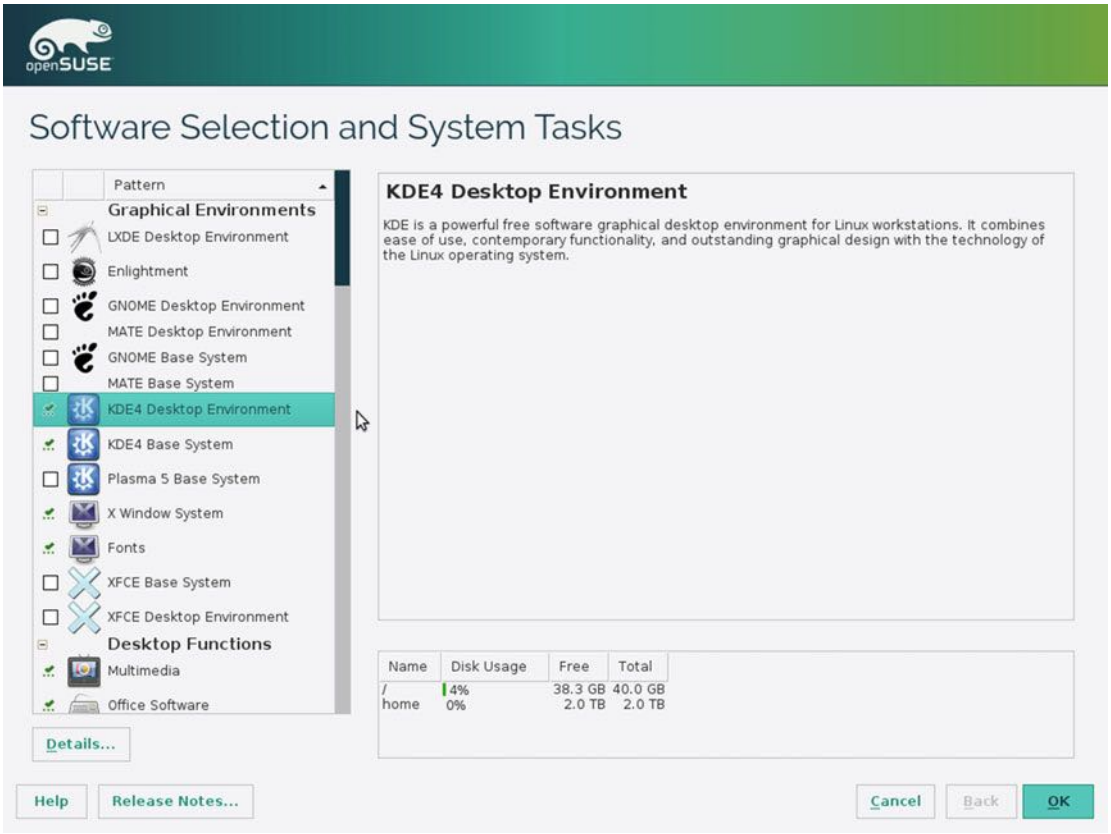


Figure 7-19. *The software selection of the openSUSE installation*

Since a confirmation is required to begin the install process, accept it. While the operating system is being installed on the disk, you can see a screen with a progress bar in the bottom and a series of messages about the distribution above (Figure 7-20). There two additional tabs where you can follow the details of the operations that are executing at that moment and the notes on this specific release.

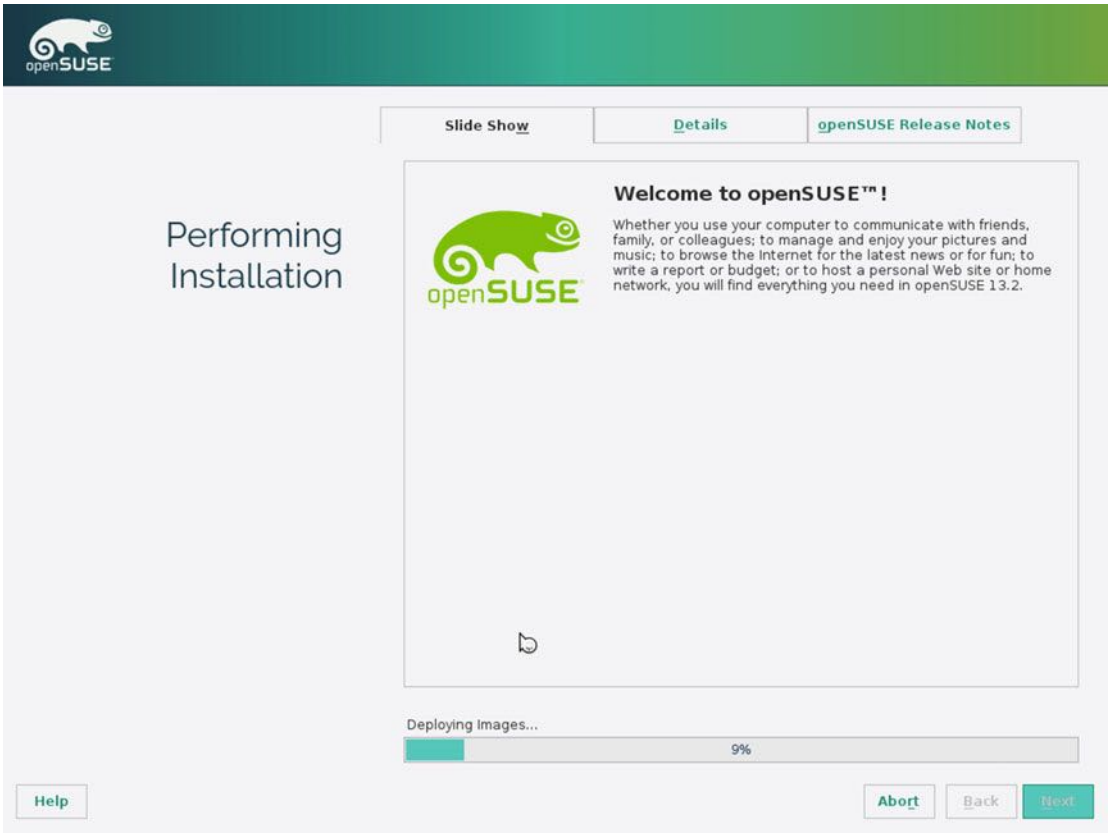


Figure 7-20. *openSUSE is installing*

When the process of installing all the packages on the disk is done, it will tell you that it is going to reboot. After the reboot, you will end up on the first screen again. If you do nothing, it will boot from the hard disk, thereby booting your fresh install of openSUSE. When booting from disk, the first screen that you will see is the already-familiar Grub boot manager, shown in Figure 7-21.

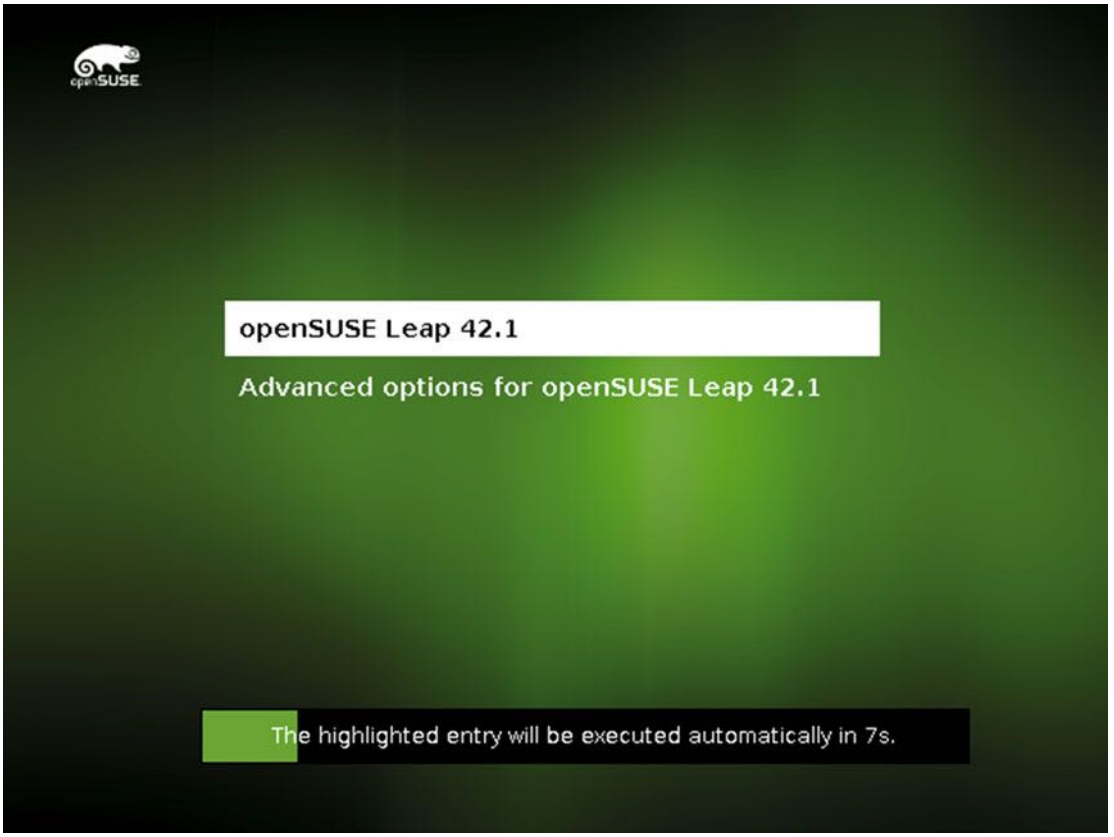


Figure 7-21. *The Grub boot loader of openSUSE*

The next screen is the most critical one: the encryption password to decrypt the contents of your hard disk. If you forgot that password, bad luck! You can't continue (Figure 7-22).

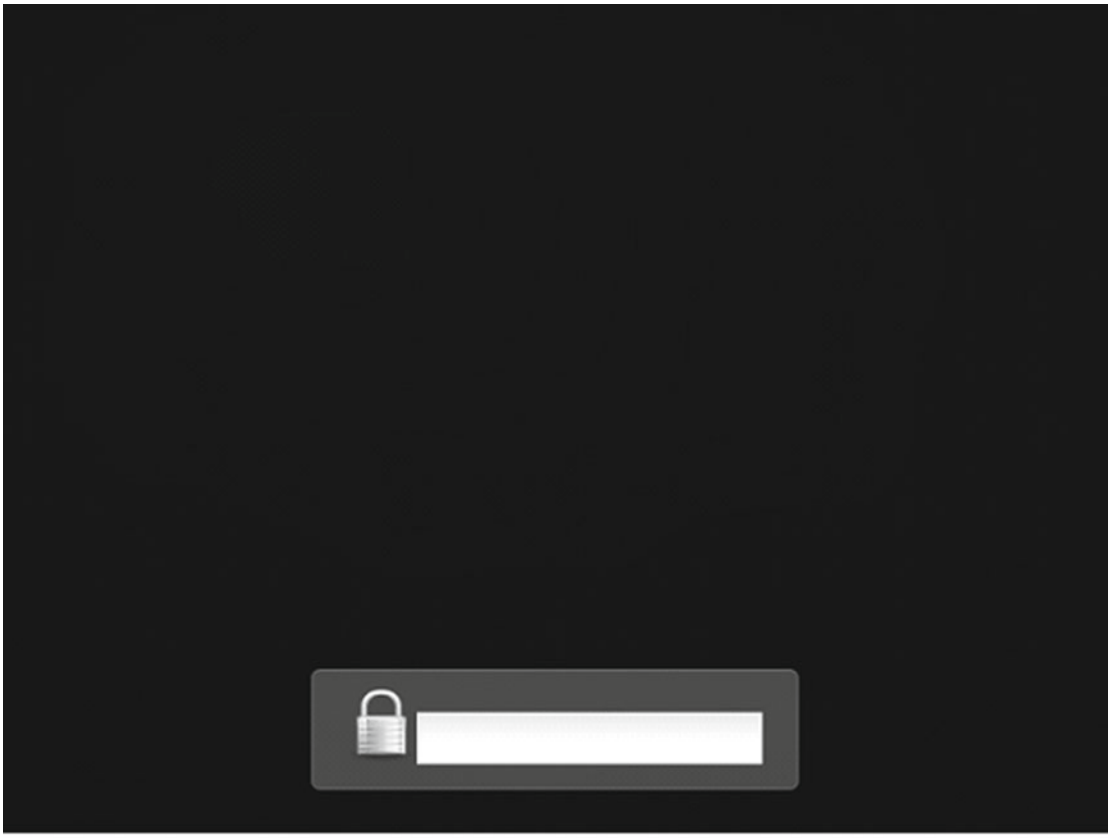


Figure 7-22. *The decrypt password for your files. I hope you remember it!*

If you successfully entered the password, the booting process will continue. And if you did not select the automatic login (I hope not), then you will see the screen in Figure 7-23. Once you introduce your name and your password, you can start your first KDE session.

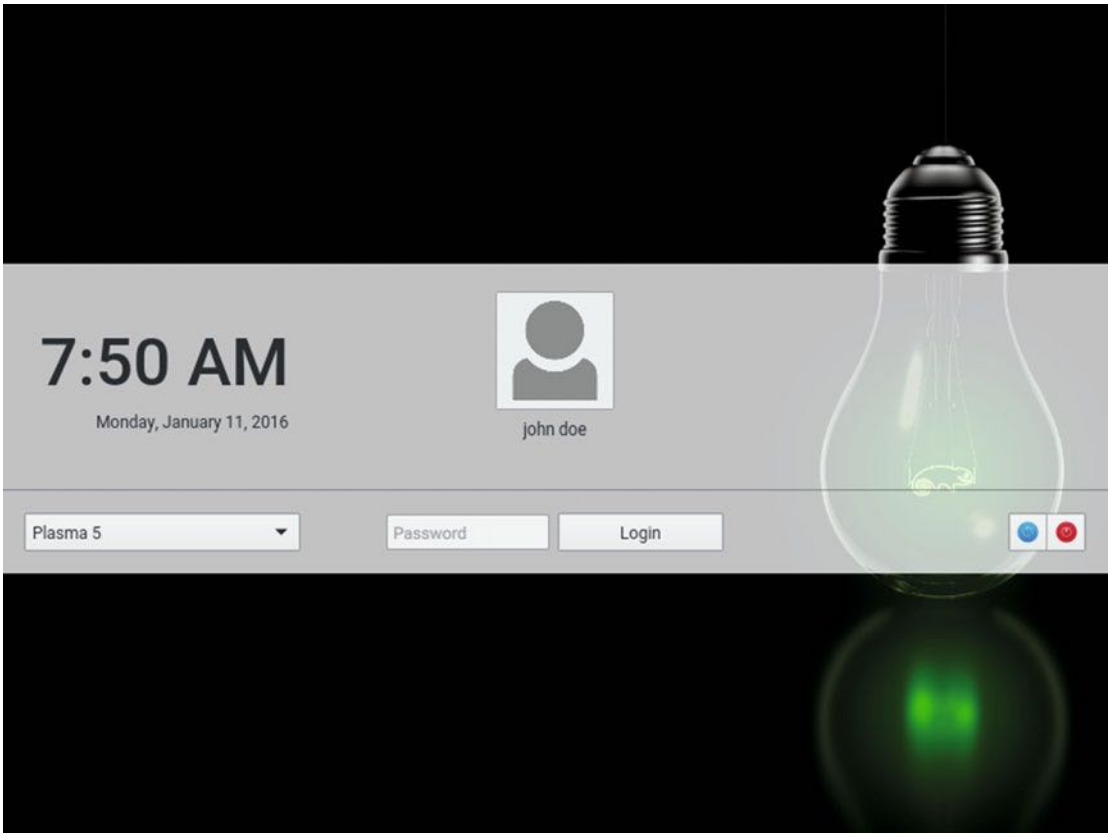


Figure 7-23. *The openSUSE login screen*

And voila! You can enjoy your new operating system (Figure 7-24).

Figure 7-24. *The openSUSE KDE desktop*

Maintenance

Well, in a manner between the Ubuntu one and the Fedora one, openSUSE is intended to be maintained in an easy way, centralized in the YaST tool. So, let's see how these tasks are performed in openSUSE.

Updating and Managing Apps

Although you can use the command line and the tool zypper to update and install/remove your apps, the default goal is to do all of this using YaST. YaST has a complete section for manage software, as you can see in Figure 7-25.

Figure 7-25. *The software section of YaST*

openSUSE, like Ubuntu, will alert you when new updates are available (Figure 7-26), so you can use that dialog to perform the updates directly from it.

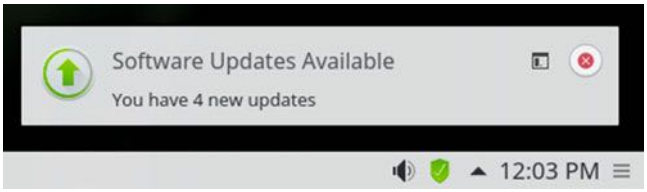


Figure 7-26. *New updates available notification*

But you can always check the updates manually by using the Online Update option from YaST (Figure 7-27) to perform the update. It is the regular software manager from YaST, but only the patches tab is selected by default. This is the same tool that you must use to remove or install software. It is a very powerful tool indeed, but I think that it's not intuitive, so beginners may find it hard to use at first. Personally, I find this tool and Synaptic (one that you can use in Debian or Ubuntu) to be very poor in terms of usability; I prefer to use command line utilities for these tasks because they are easier to control and use once you know how. However, openSUSE is focused on the corporate world and advanced users, so these tools are what their users expect to find.

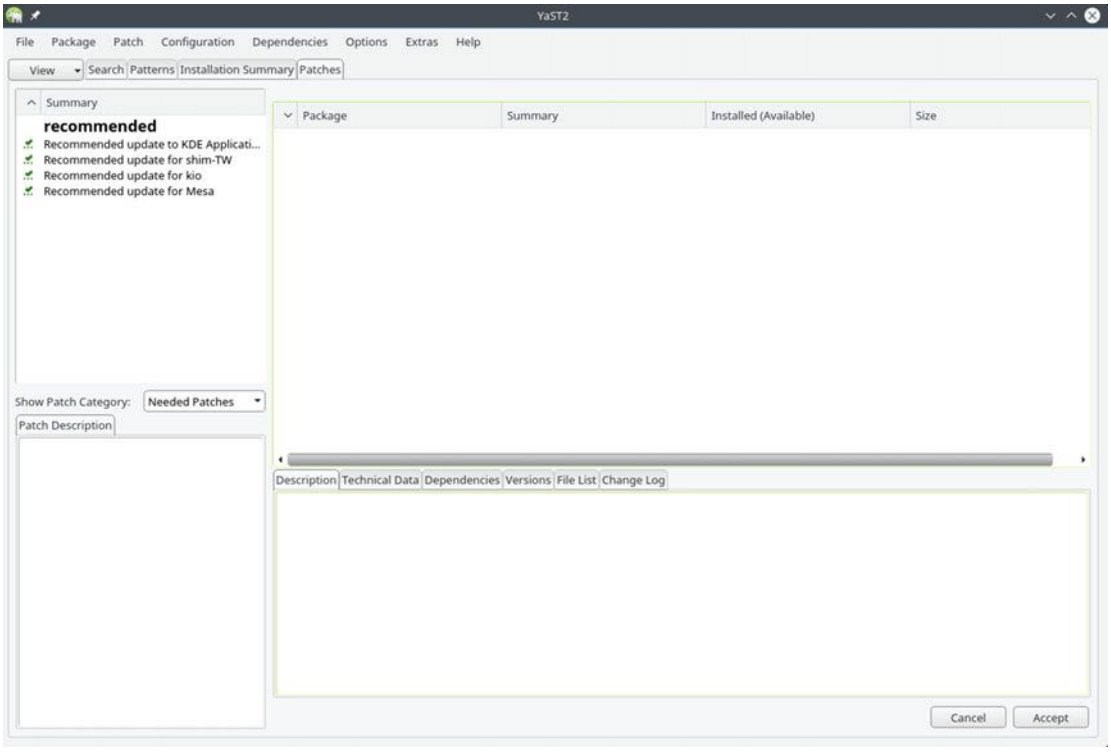


Figure 7-27. *The YaST software updater*

Upgrading

There are two ways to perform a release upgrade in openSUSE: the online method and the offline one. To perform the online method, use the command line, the zypper command, and the online repositories. The offline method only needs an ISO image of the new release, and then you pick the Upgrade option on the first screen instead the Installation one. The command to perform the online method is very easy to use at first sight, `# zypper dup`, but you need to perform a series of operations first to change the repositories, update your packages, and so on[2]. So, if you are a new user to the distro, I recommend you choose the offline method and follow the instructions.

Pros and Cons

The following is a list of some things that I personally see as pros and cons of the openSUSE distribution. Of course there's always room for discussion in this matter, but I've done my best to be as objective as possible.

Pros

- YaST is one of the best and powerful tools to configure and install a Linux OS.
- openSUSE is a secure distribution and can be hardened directly from YaST.
- It takes care of a lot of little things.
- You have the option of a rolling release version if you like to have the latest updates or you have the option of a very stable version.
- openSUSE supports only free and open source software but lets you access proprietary software easily if you want.
- The developers contribute and collaborate with the upstream versions of packages and are great contributors to the kernel.
- It is an original distribution, not a derivative.
- It uses one of the most extended package formats, rpm.
- SUSE is behind openSUSE, and openSUSE is part of the base of SUSE.
- It is well documented, and if you know German, you even can buy the boxed version and get a printed manual.

Cons

- openSUSE is focused on the corporate world and advanced users. It's not user friendly for beginners.
- YaST is powerful but not intuitive, and it is not the best way to configure some things.
- The hardware support by default is not as good as other distros; you may have to manually use proprietary drivers.

[2] https://en.opensuse.org/SDB:System_upgrade

- The regular release scheme of the main distro is not as regular as the Ubuntu or Fedora ones, but it's not as bad as the Debian one.
- It has no commercial support.
- It only supports one architecture.
- The installation base is not as big as Fedora, Debian, or Ubuntu.

Summary

Another distro, openSUSE, was put under the microscope. It takes a different approach to many common matters. There is no equivalent to YaST in any of the distros seen in previous chapters, and I assure it is not going to be the last new thing that you will see in this book.

Thus, in the next chapter you will see how Linux Mint approaches various topics.

CHAPTER 8

Linux Mint

Linux Mint is the new sensation. It's been the most popular distro for the past four years, even more popular than Ubuntu, the distro upon which it is based. This is ironic, because Ubuntu was the most popular distribution for the prior six years and it was based on another one, Debian[1]. However, there's popularity and there's install base, and in the latter, Ubuntu still has a bigger number by far. So, what makes Linux Mint so successful? The main reason is that Linux Mint goes a step further than Ubuntu in making the distro easier to use, and it listens to its user base about what direction to take in its development.

History

Linux Mint was started by a French developer, Clement Lefebvre, in 2006. It was based on Kubuntu (only for the original release; after that one, Ubuntu was the base distro). Lefebvre's goal was to create a more elegant distro. After its release, it was slowly but continuously adopted by a growing community, largely based in Europe. However, two events would make a big difference in expanding the popularity and adoption of Mint in a short period of time.

Around 2011, the Gnome desktop environment made a radical change in its version 3. A big number of Gnome users did not like that change, and began to search for an alternative. At that time, KDE was also transitioning from version 3 to 4, with frequent bugs due to the immaturity of the new version. Of the few distros that did not change to version 3 of Gnome, Mint was the friendliest, most stable, and most polished. The other event was when Ubuntu switched from Gnome to Unity, a desktop environment even more hated and unwelcomed that Gnome 3; this provoked a great migration of users from Ubuntu to Mint.

Mint was able to take advantage of this avalanche of users coming into its community and develop its own desktop environment, Cinnamon, as an alternative to Gnome 3; it also adopted and helped develop MATE as an alternative to Gnome 2. And Mint listened to its users again: it created an alternative version based on Debian Stable that works as a semi-rolling release version of Mint.

Through the years, Linux Mint would usurp the throne as the top European distro from the venerable openSUSE distribution.

Philosophy

When Clement Lefebvre defined the goal of Linux Mint as an elegant distribution, what he meant was a distro focused on ease of use, appealing design, stability, and feedback from its users. The focus on usability is reflected in things like the full multimedia support out of the box, something that is not easy to achieve in

[1]Popularity ranks based on DistroWatch (http://distrowatch.com)

J. Dieguez Castro, *Introducing Linux Distros*, DOI 10.1007/978-1-4842-1392-6_8

other distros. Listening to feedback from its users is the reason behind the development of Cinnamon and the adoption of MATE and other software contributions. Also, Linux Mint is known as a very polished distro, where everything works and the little details are taken care of.

As a result, in terms of ease of use, Linux Mint is probably one of the top three. Another thing worth mentioning is that Linux Mint is focused only on the desktop user, and this obviously helps it end up with a more polished product.

Distro Selection Criteria

Now that you know a little history of Mint, let's see how this particular distro fares on the section criteria from Chapter 2.

Purpose and Environment

Linux Mint is a general purpose distro focused only on the desktop user. It comes in two versions:

- **Linux Mint**: The main distribution, it is a regular release version and is based on Ubuntu. There are different images for each desktop environment supported (Cinnamon, MATE, KDE, Xfce). Moreover, the two main DEs offer versions with no codecs, for use in countries where multimedia codecs are patented.
- **LMDE**: A version based on Debian (LMDE means Linux Mint Debian Edition), it has a semi-rolling release cycle.

Support

Obviously, without a company behind it, Linux Mint offers only community support, and since it has the smallest community of all the distros seen in this book so far, the support is not as good. The available channels to get this support are the following:

- **Documentation**: `www.linuxmint.com/documentation.php`
- **Forum**: `http://forums.linuxmint.com/`
- **Mailing Lists**: `http://librelist.com/browser/linuxmint/` `linuxmint@librelist.com`
- **IRC**: `#linuxmint-help at irc.spotchat.org`

Because Linux Mint is based on Ubuntu, some things can be resolved via Ubuntu's support channels, so you can always try those channels if you don't find help in the Mint ones.

The feedback from the users is collected, discussed, and often implemented from the Ideas section of the community web site, `http://community.linuxmint.com/idea`.

User Friendliness

As mentioned, Linux Mint is one of the easiest Linux distros to use, and you can see this from the start. It takes the best parts of Ubuntu and adds its own features to make the general experience a very pleasant one. There are a lot of little details that intend to minimize friction with the OS, things like the domain blocker (not having to deal with the firewall to set parental controls) or the Upload Manager (for managing FTP and SCP services). In the "Maintenance" section, you'll see how far Linux Mint goes to make certain things easier. This is one of the distros that I would recommend to beginners.

Stability

Linux Mint is a very stable distribution. I think it's more stable than its mother one, Ubuntu, because Mint uses the Long-Term Support releases of Ubuntu, and it only offers "safe" updates by default. And by "safe," the Linux Mint developers mean that these updates will not break your OS. Thus, this distro does not have the latest packages (in fact, it is usually more outdated than Ubuntu), but it offers a better balance between stability and freshness than Debian Stable, for example.

Mint follows a standard release model (at least with the Ubuntu-based one), and it is based on the LTS releases of Ubuntu. Thus, after an Ubuntu LTS release (every two years) is released, so is a major version of Mint, and after each regular Ubuntu release (every 3 months), a minor version of Mint (without too many changes) is released. Each version has a code name, which is always female name ending with "a." They follow an alphabetical order and the first letter of the code name is in the alphabet position that corresponds to the version number.

Obviously, because it depends on the Ubuntu releases, Linux Mint cannot be as regular as Ubuntu because its developers need some time to prepare their own release, and this amount of time varies per release.

Hardware Support

Being an Ubuntu derivative, the hardware support is almost as good as Ubuntu's, but Mint always gives priority to open source drivers over private ones. (You can always install private ones later through the driver's manager.)

Aesthetics

A good global aesthetic is one of the major goals of Linux Mint. Mint develops its own desktop environment, which is made with design in mind. You can see this in every corner of the distro, from the colors, the desktop, the logo, the theme, the icons, to the wallpaper. There is a general consistency throughout the desktop and it is easy on the eyes. Linux Mint is often cited as the most appealing of all of the distros (of course this is a very subjective matter, but there is a global consensus about it).

Desktop Environment

There are two main official desktop environments for Linux Mint: Cinnamon and MATE. The first one is a fork of Gnome Shell (Gnome 3) and the second a fork of Gnome 2. As mentioned, Cinnamon was developed by the Mint community; it is the most popular, and it is the de facto official Mint DE. Other desktop environments like KDE and Xfce are provided in separate ISO images.

Init System

Currently Linux Mint still uses Upstart (and SysV in LMDE) as its init system, but this will change in the next release (which will happen soon after the next Ubuntu LTS release) when it will adopt systemd, something the majority of distros have done already.

Package Management System

Since it's based on Ubuntu, obviously the package management system is dpkg, so it uses .deb packages. And even though Linux Mint has a repository for its own packages, it uses the Ubuntu repositories as well, so it gets the benefits of the packages available for Ubuntu. You can configure other popular sources for Ubuntu packages like the PPAs or the Getdeb repository. Linux Mint offers a good graphical tool to manage those packages, the Software Manager (mintinstall) but you can also use Synaptic for the same task.

Architecture

Mint only supports the two major architectures: Intel and AMD in 32- and 64-bit versions.

Security/Anonymity

Mint is based on Ubuntu and so it is reasonable secure by default. However, it is not as secure as Ubuntu because it does not implement AppArmor out of the box. Also, the levels update policy can make the OS a bit unsafe because the kernel updates (some of them with security patches) are not made by default. As always, there's a balance between ease of use and security; normally you need more knowledge to maintain a more secure distro.

Principles and Ethics

Linux Mint gives preferences to free software over proprietary software, but it does offer proprietary software like Adobe Flash support and multimedia codecs. Even though the graphic drivers are set by default to open source drivers, it has tools like the Driver Manager to deal with proprietary drivers, and you can always find some in the repositories.

Live CD

Like Ubuntu, the ISO images of Linux Mint are also Live DVD images.

Professional Certification

Linux Mint does not offer any professional certification.

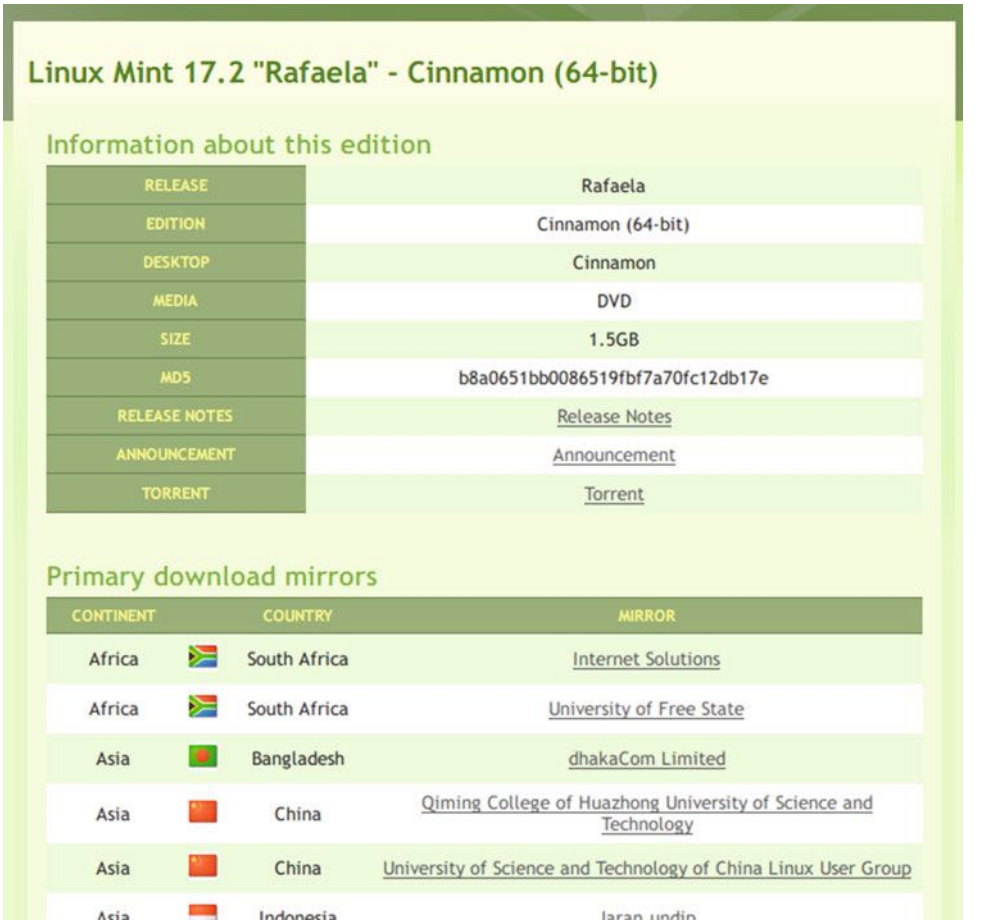
Installation

Because Linux Mint is based on Ubuntu, the Mint installation process is very similar to the Ubuntu one (in fact, it's the same one with little, aesthetic changes), so I'm going to introduce another scenario here to show you something different from the installation in the Ubuntu chapter.

The first of all, as usual, go to the download page of the distribution, `www.linuxmint.com/download.php`, where you will see something similar to Figure 8-1.

Figure 8-1. *The Linux Mint downloads page*

Select the main version (the Ubuntu-based one) that is current; at the time of writing, it was the 17.3 release. You can select different images depending on the desktop environment that you want to use; Cinnamon is the default one. Also, there are different versions for each of the main DEs (Cinnamon and MATE), so you can choose if you want a version without multimedia codecs or an OEM image. Let's choose the most popular one, the Cinnamon 64-bit version. After you click that link, you will see the screen in Figure 8-2.



Linux Mint 17.2 "Rafaela" - Cinnamon (64-bit)

Information about this edition

RELEASE	Rafaela
EDITION	Cinnamon (64-bit)
DESKTOP	Cinnamon
MEDIA	DVD
SIZE	1.5GB
MD5	b8a0651bb0086519fbf7a70fc12db17e
RELEASE NOTES	Release Notes
ANNOUNCEMENT	Announcement
TORRENT	Torrent

Primary download mirrors

CONTINENT	COUNTRY	MIRROR
Africa	South Africa	Internet Solutions
Africa	South Africa	University of Free State
Asia	Bangladesh	dhakaCom Limited
Asia	China	Qiming College of Huazhong University of Science and Technology
Asia	China	University of Science and Technology of China Linux User Group
Asia	Indonesia	Jaran undip

Figure 8-2. *The download options for the ISO image of Linux Mint*

The first option that you see here is to download the ISO image via Bit Torrent, the most suitable option for a Linux distribution like this, with little economic resources. Below this is a long list of mirrors where you can download the ISO image through HTTP. It is up to you, but I chose to download the torrent file for the image.

Once you have downloaded the image, you can start the installation. In my case, I'm going to install Linux Mint in a machine that has a 2TB HDD with a default Windows 8 installation occupying the entire hard drive. The goal is to have the two OSes installed at the same time on the machine and have the ability to choose the one you want to use at boot up time.

The first screen that appears when you boot from the Linux Mint ISO is a little different than in other distros (a ten-second countdown). If you don't wait those ten seconds, a live session of Linux Mint starts; otherwise, if you press any key, you will see the screen shown in Figure 8-3. It's the common menu that you see in other distros, but here Mint hides it in order to make the installation process easier and less intimidating.

Figure 8-3. *The boot-up menu of Linux Mint*

Once you are in the Live session of Linux Mint (Figure 8-4), you can play around a little; at the end, click the "Install Linux Mint" icon to start the actual installation (see Figure 8-5).

Figure 8-4. *The Linux Mint Live session*

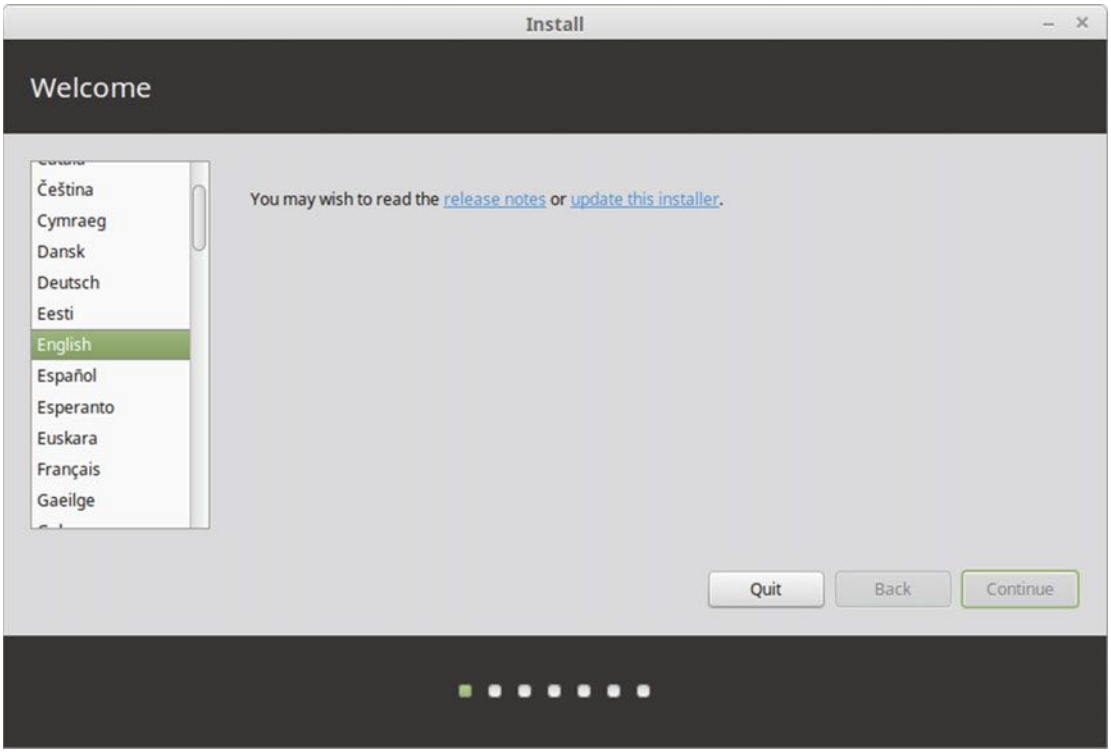


Figure 8-5. *The first step of the Mint installation*

The first step is the one where you must select the language to use in the process and your OS. You can also read the notes on the release or update the install software itself (this option only appears if necessary, after you check for an upgrade). Then comes the easy part (Figure 8-6).

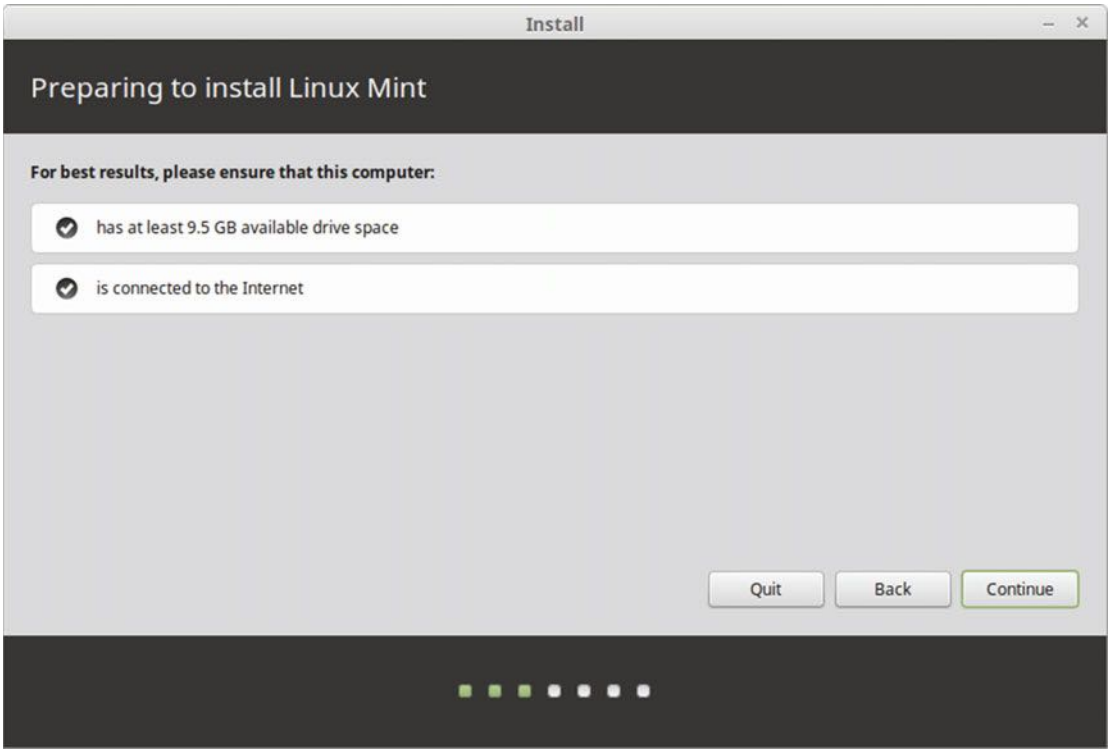


Figure 8-6. *Checking for prerequisites*

Nothing is required of you here. It's just testing the system to make sure that it has the hardware necessary for the installation. Continue to the next step.

You can see in Figure 8-7 that the Windows OS was detected; you are offered the option to install Mint alongside it or delete all of the data on the hard disk and install Mint. As I said at the start, the intention is to have both OSes available at the same time. This option is selected by default, so press Continue to go to next step (Figure 8-8).

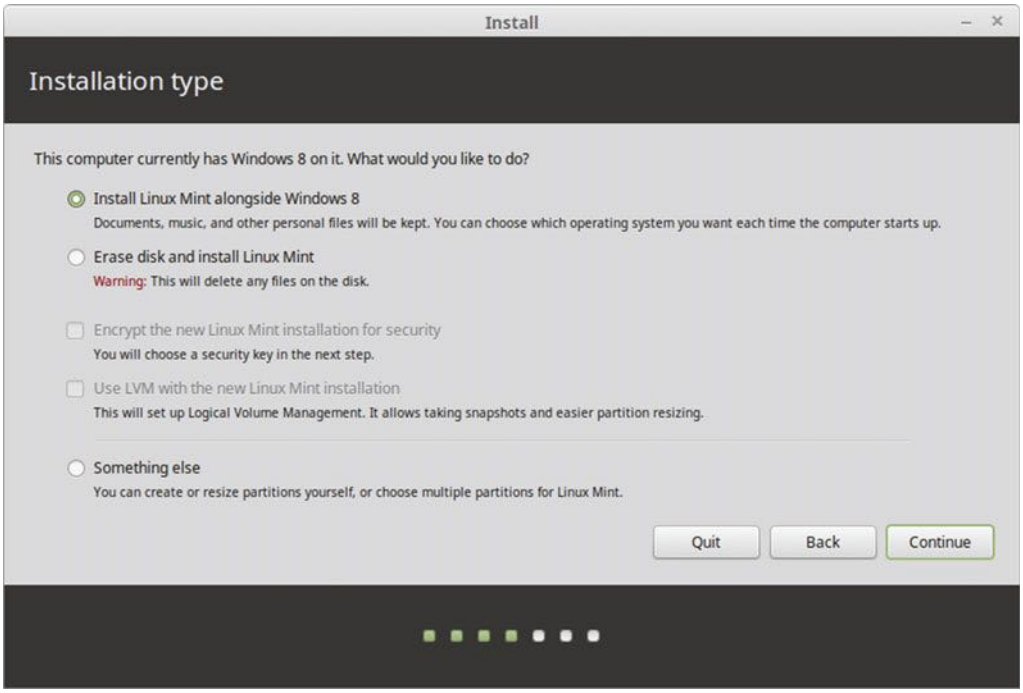


Figure 8-7. *Installation type screen*

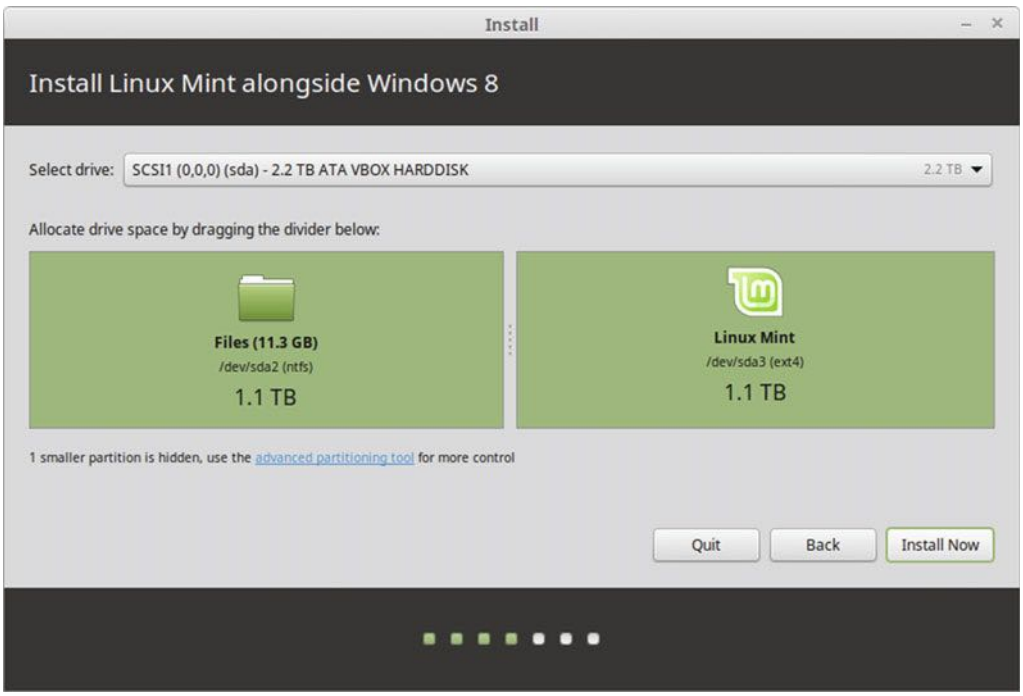


Figure 8-8. *Size of the disk distribution for both operating systems*

Now you must decide how much disk space to allocate for each operating system. By default the installation program uses a 50/50 split, but you can modify this by simply moving the bar that appears between both green squares. You can also access the advanced partitioning tool via a link.

You can see the current status of my disk in Figure 8-9. There are currently two partitions, a very small one for the Windows 8 loader and a big one for the OS and the data.

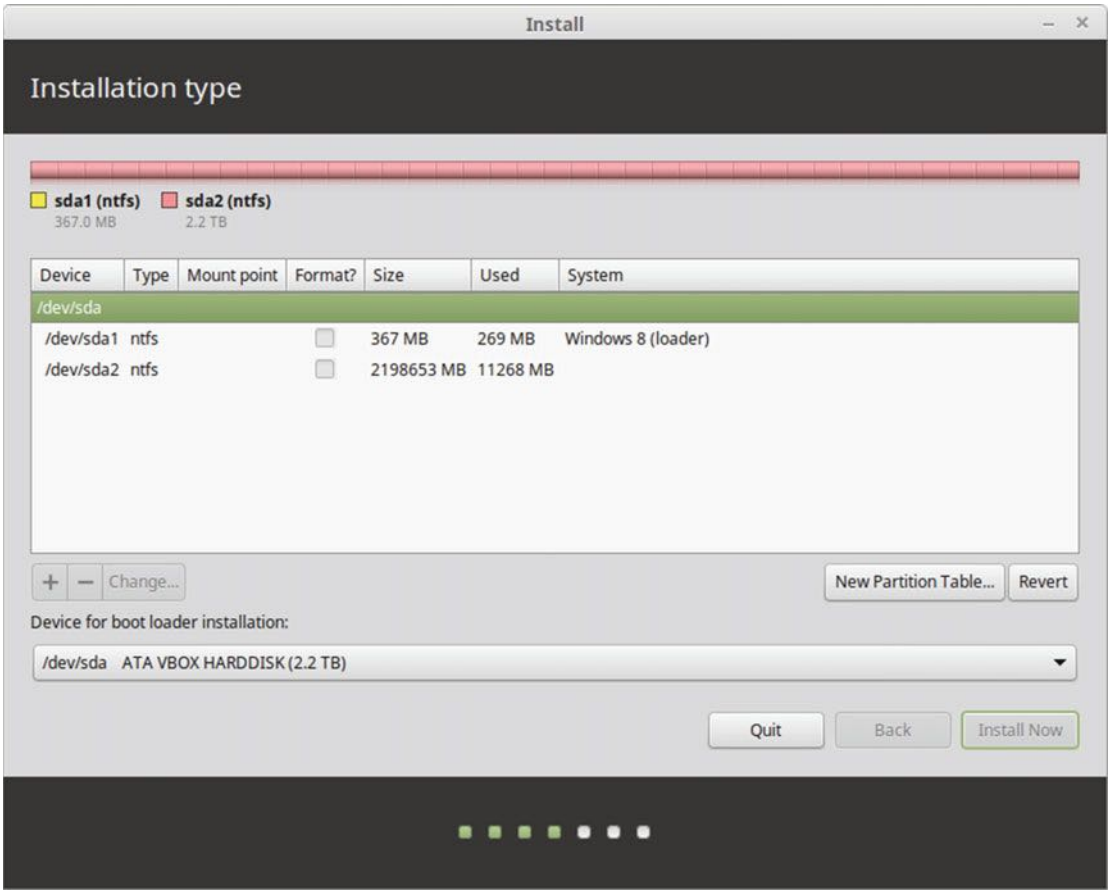


Figure 8-9. *The current partitions of the disk with Windows 8*

Return to the previous screen (Figure 8-8). If you press the Install Now button, the dialog shown in Figure 8-10 will pop up and inform you that some disk operations are going to be performed, in particular the resizing of the big ntfs partition of Windows. Of course, at this point you should have already made a backup of your data, because this operation is delicate and something could go wrong (it's rare, but you know, stuff happens). If you are ready, press the Continue button.

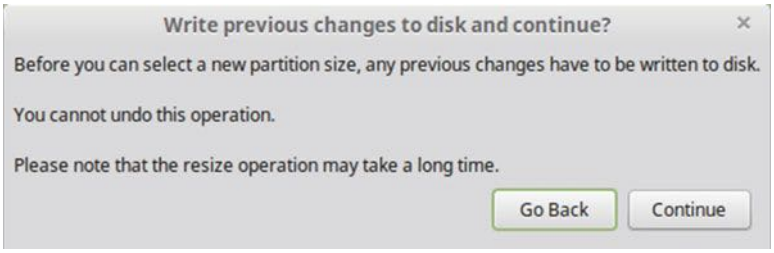


Figure 8-10. *The warning dialog about the resize operation*

Once the partition is resized, and before it creates the new partitions to allocate your Linux Mint installation, a pop-up dialog appears to show you the changes that will be made and ask for your confirmation (Figure 8-11). At this point, you should know that the default proposed partition (a root partition and a swap one) is not the best plan for the long term. So, instead of continuing from here, let's go back and set up a better partition scheme.

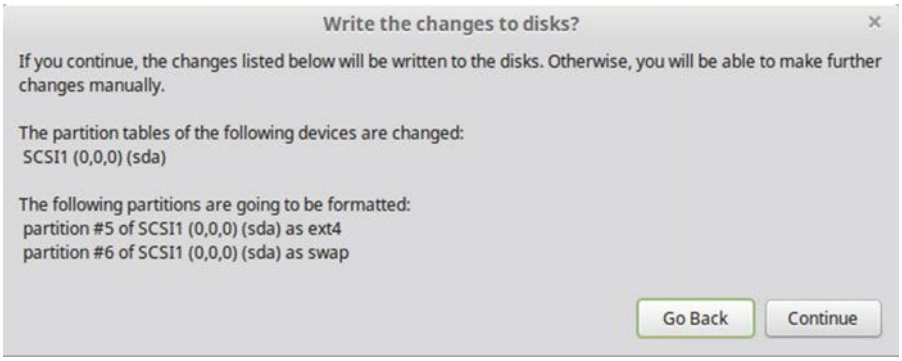


Figure 8-11. *Are you sure you want to write the changes to the disk?*

If you go back, you'll land in the Figure 8-7 screen again. This time, choose the "Something else" option to create your own partition scheme. The advanced partitioner now shows something like Figure 8-12, where the Windows partition was already made smaller. So now you have half of the disk as available space. In that space, let's create new partitions similar to Figure 8-13. Yes, they could be different, but this is good enough without being too complex.

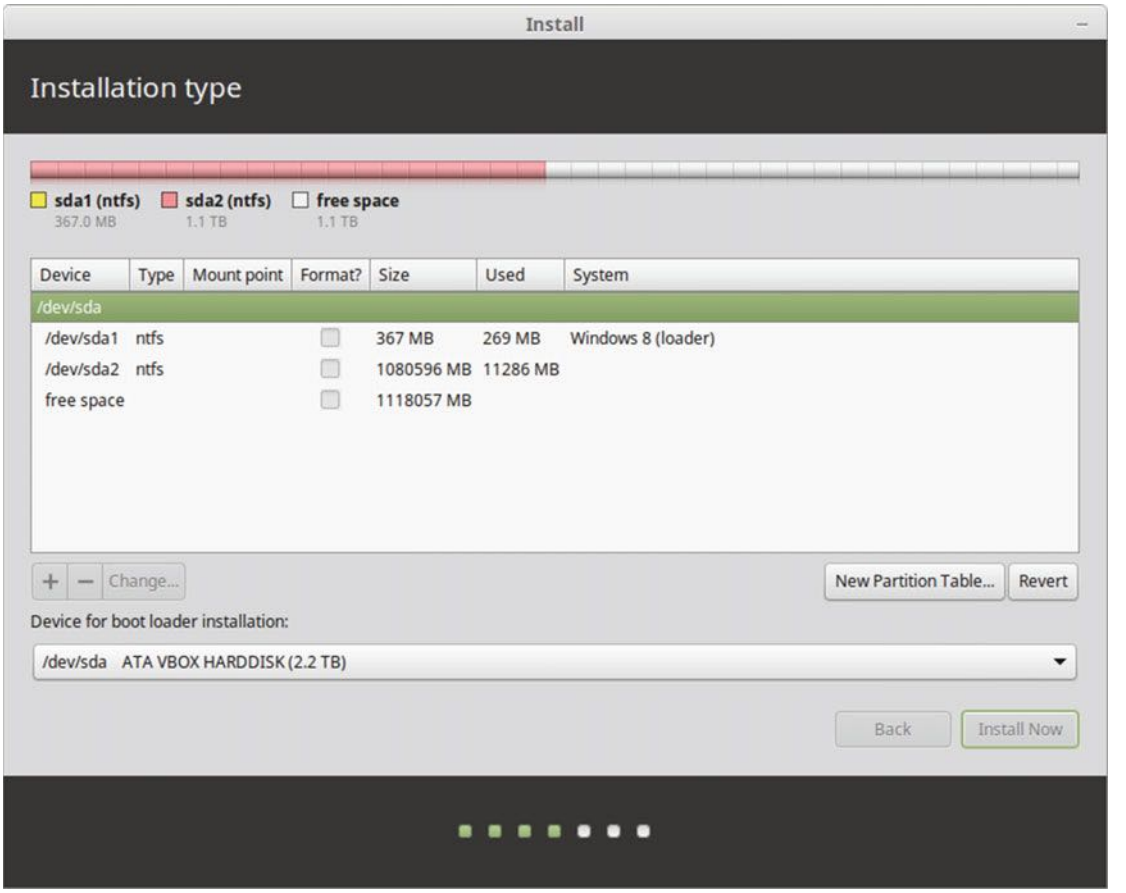


Figure 8-12. *The Windows partition already resized*

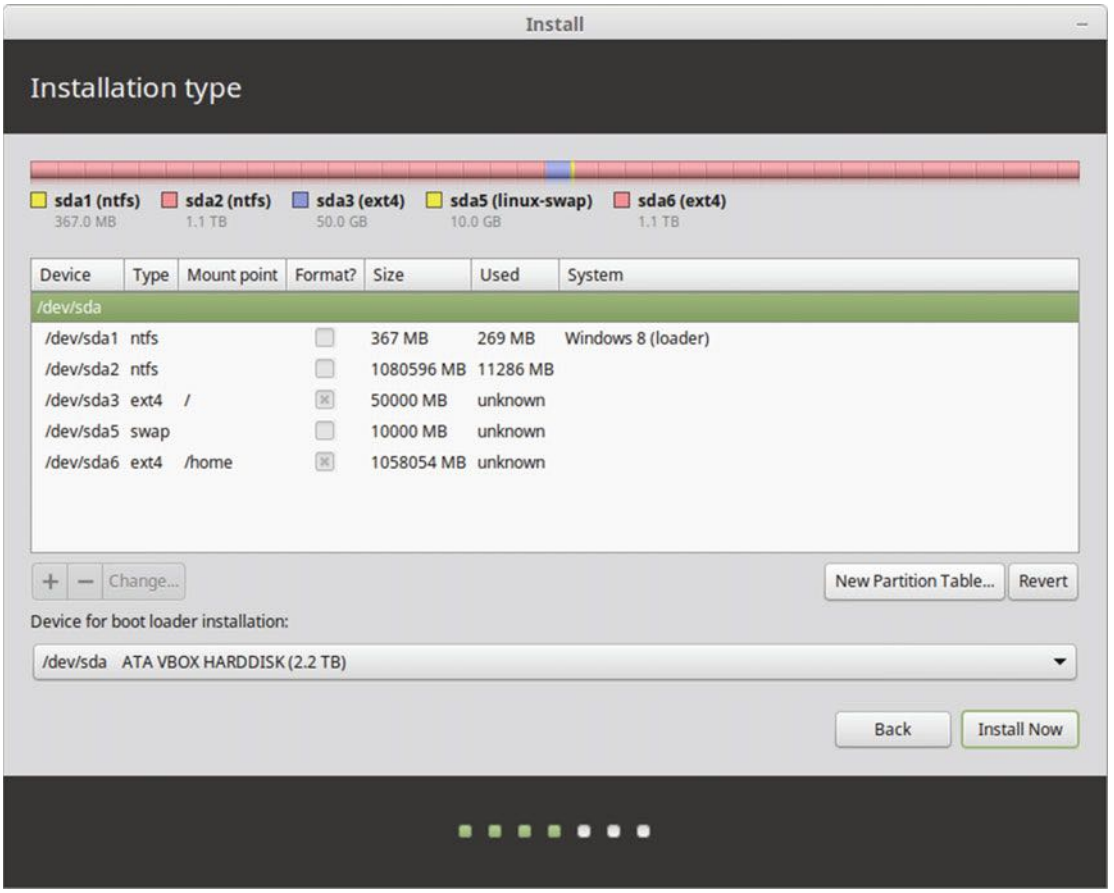


Figure 8-13. *The new partition scheme to install Linux Mint*

If you press the Install Now button, you'll see the summary shown in Figure 8-14.

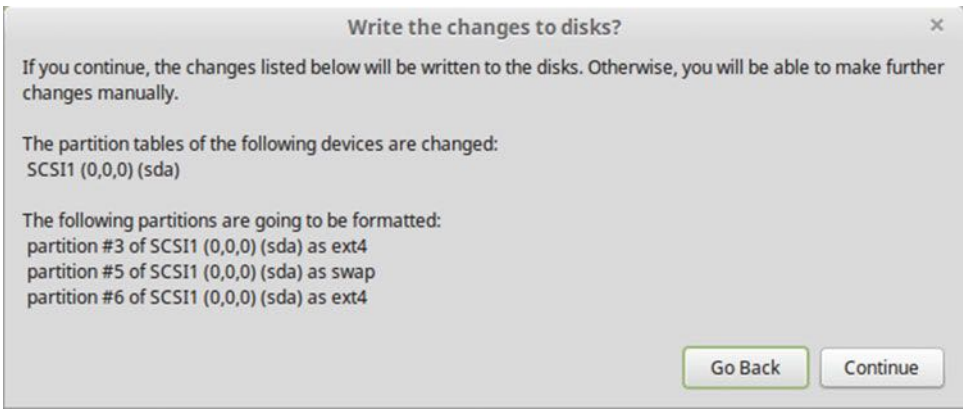


Figure 8-14. *The summary of the new partition scheme*

If you continue, the changes will be written to the disk and the installation process will go on. The next steps are to set the current time zone where you are located (Figure 8-15) and the keyboard layout (Figure 8-16).

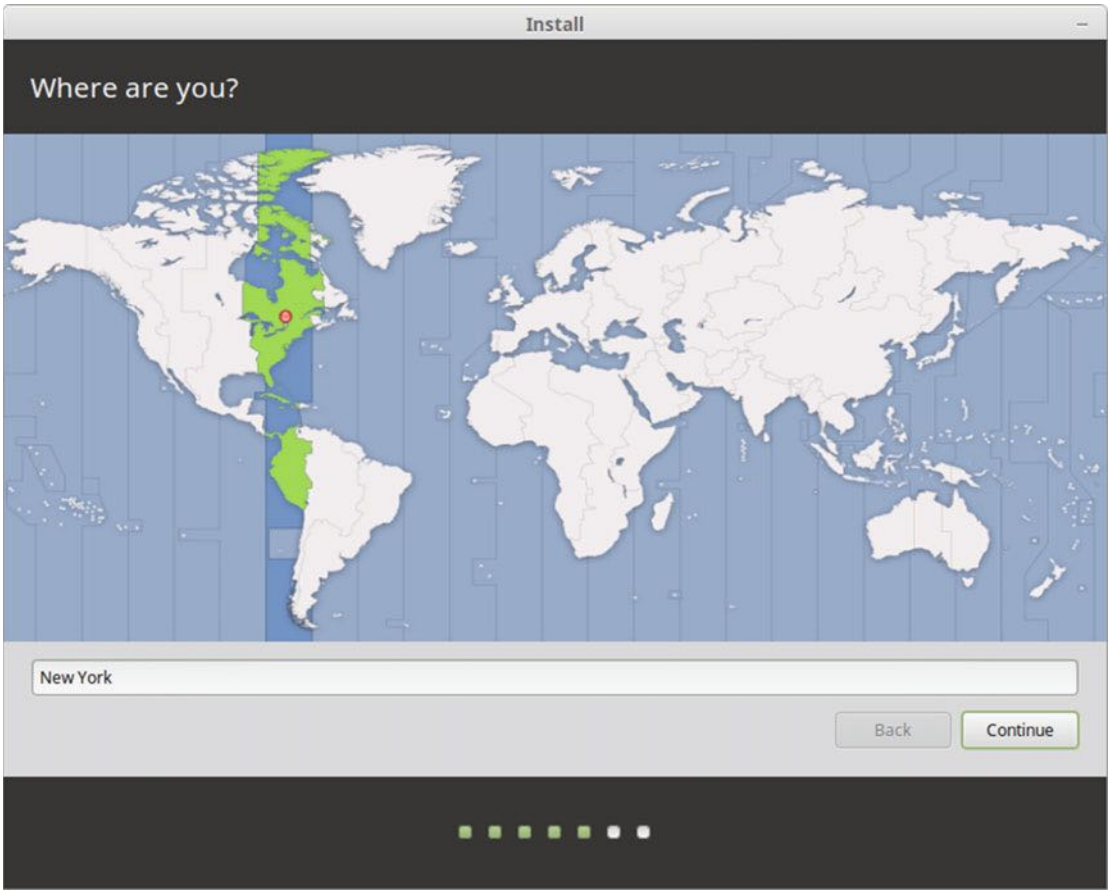


Figure 8-15. *Time zone settings*

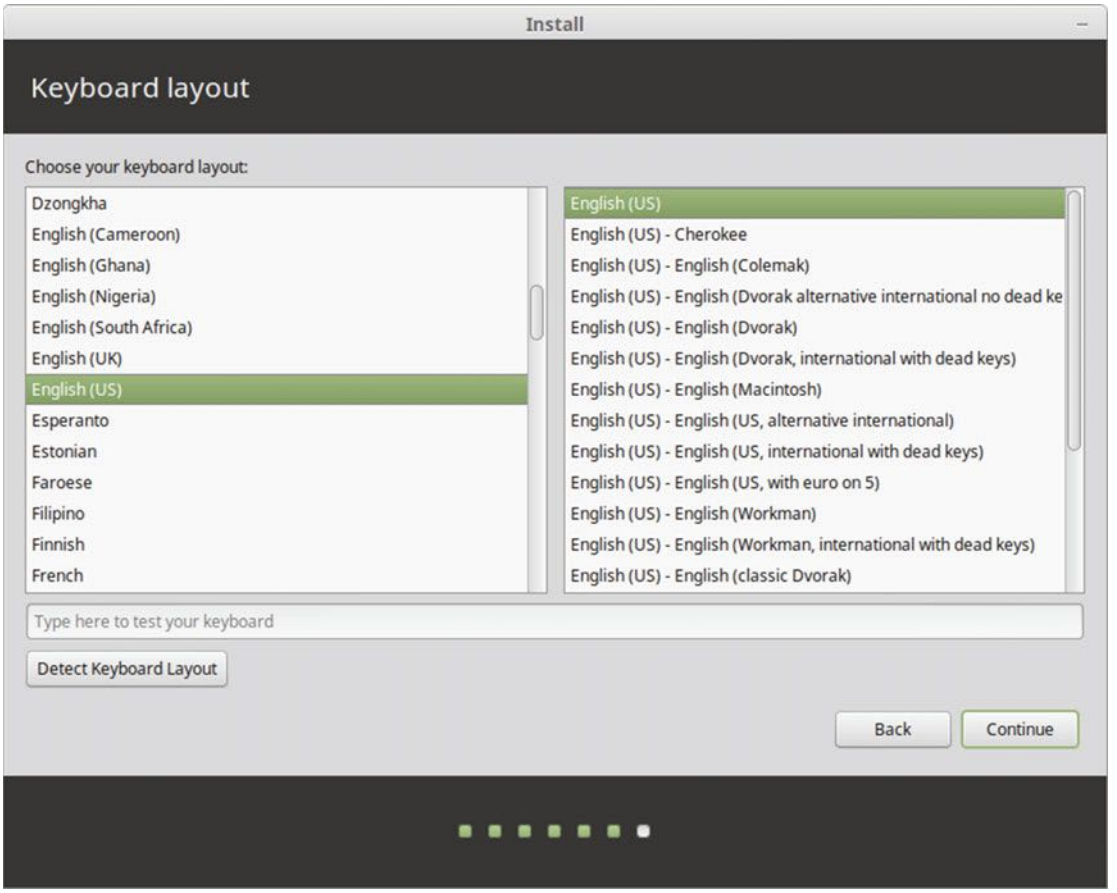


Figure 8-16. *The keyboard layout settings*

In Figure 8-17, you can observe the user settings; these are the same ones I have used all along. However, the password that Fedora and especially openSUSE found as weak is seen here as a fair one. Well, press the Continue button anyway.

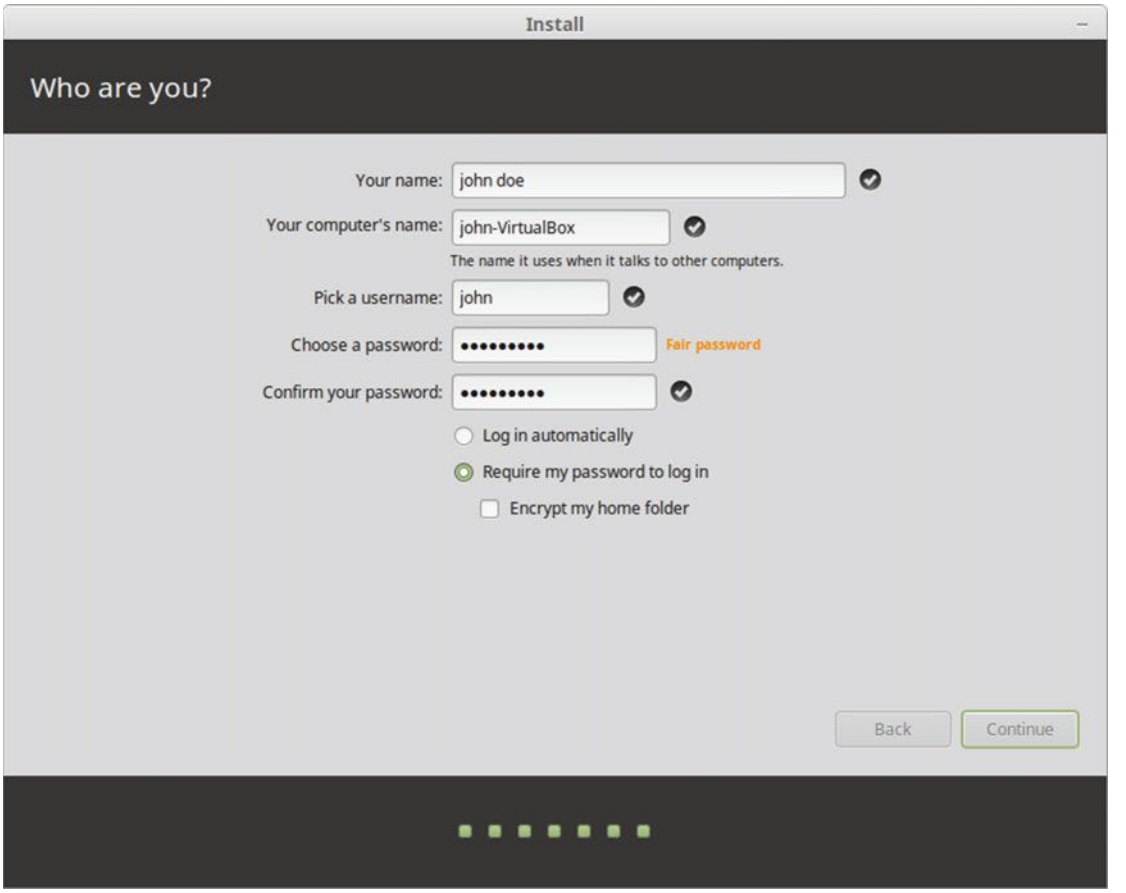


Figure 8-17. *The user settings*

The boring part of the installation, the copying of the files of the OS to the disk, will begin. At the end, you get the dialog shown in Figure 8-18. When you press the Restart Now button, the Live session will shut down and ask you to remove the ISO image file from the system (Figure 8-19).

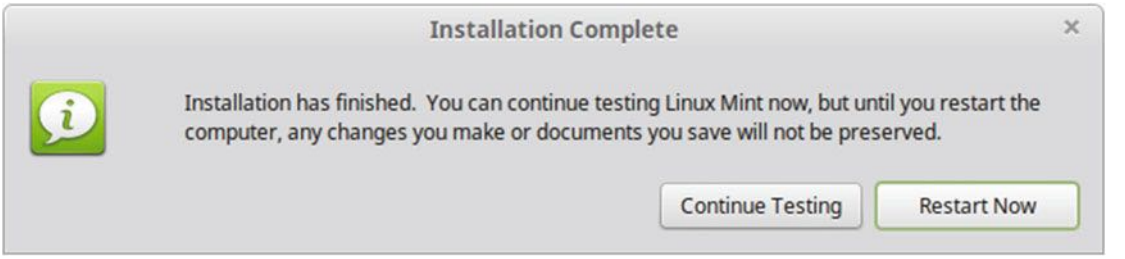


Figure 8-18. *The installation is complete*

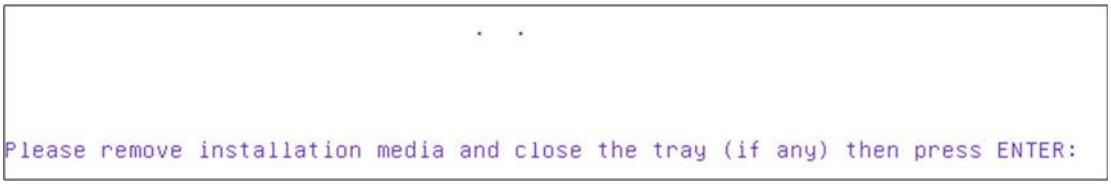


Figure 8-19. *Remove the image from the DVD/USB to restart the system*

The OS is now installed and this is the first time that you will boot up the machine with it. The first screen, as usual, is the Grub one, shown in Figure 8-20.

Figure 8-20. *The Linux Mint Grub screen*

You can observe various things here. First, at the top, in the very first line, you can read that the version of the Grub used here is an original Ubuntu package, a little detail that the Linux Mint people left behind. Also, you can see that you can opt to boot to Linux Mint (as the default option) or you can boot to the Windows 8 OS. If you choose Windows 8, it will boot as usual, as if you hadn't made any changes to the disk. And finally you should notice that there are two options to boot Mint; the second one is shown as "Advanced options ..." (see Figure 8-21). In reality, the usual two entries that are seen in almost all distros are the normal and the recovery ones. I suppose that the Mint developers took this approach to make it easier for newcomers to use.

Figure 8-21. *The advanced options for Linux Mint*

Choose the first option, and start to boot your new OS for the first time. The login screen is shown in Figure 8-22. Then you get to Linux Mint's Cinnamon desktop (Figure 8-23). As with other distros, it shows you a welcome screen with a few possibilities to start to use the system. Congratulations! You have installed a new Linux OS.

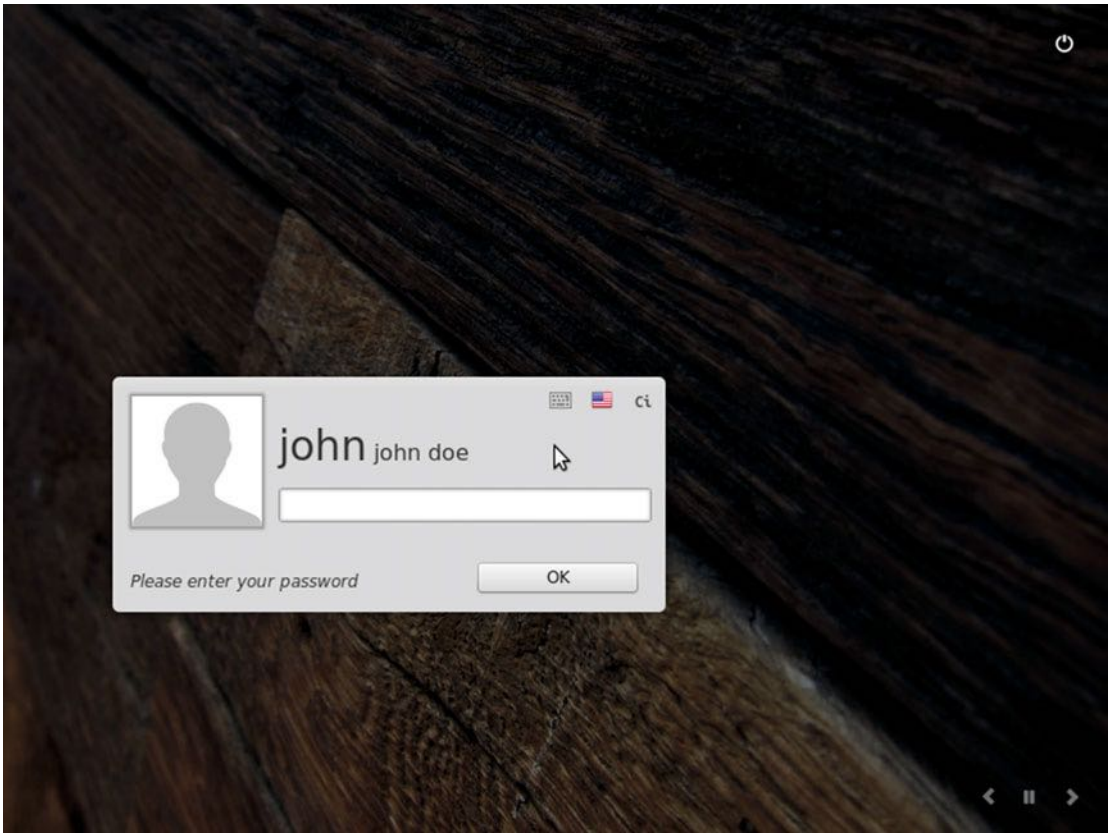


Figure 8-22. *Mint's login screen*

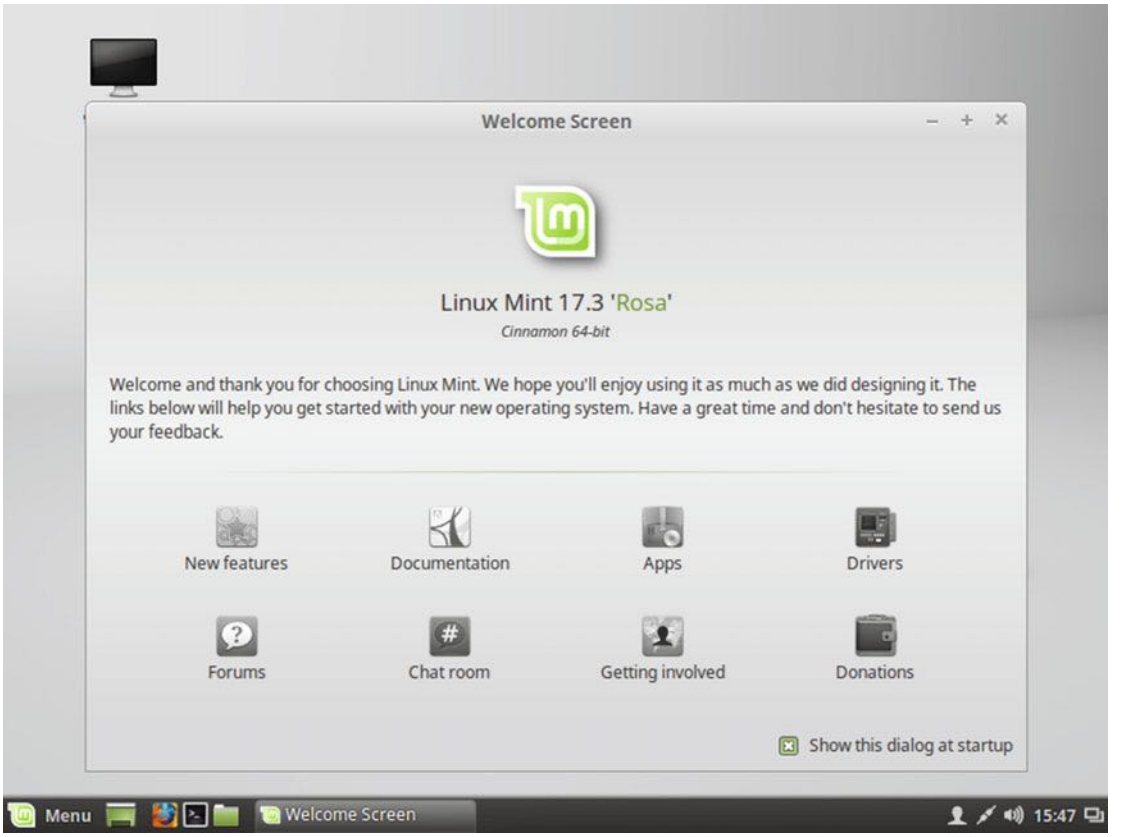


Figure 8-23. *The Linux Mint 17.3 Cinnamon screen*

Maintenance

As per its founding philosophy, Linux Mint is one of the easier distributions to maintain. You can do it all from graphical interfaces, which makes Linux Mint well suited for beginners.

Managing Apps

If you want to install or remove applications in Linux Mint, you have the same alternatives as in Ubuntu: you can do so via a dedicated graphical application, via the Synaptic tool, or in the terminal with the common command line tools for .deb packages (apt-get, aptitude, dpkg). The only difference here is that the dedicated graphic app is different from the Ubuntu one.

The Software Manager app (Figure 8-24) is a simple, easy-to-use, and intuitive application, in the same spirit as the Ubuntu equivalent, grouping apps by category. But once you start to use it, the first thing that you will notice is that the search and navigation is much faster than its Ubuntu counterpart; in fact, it has less bugs and works a little better. The Mint developers made a good thing with this app.

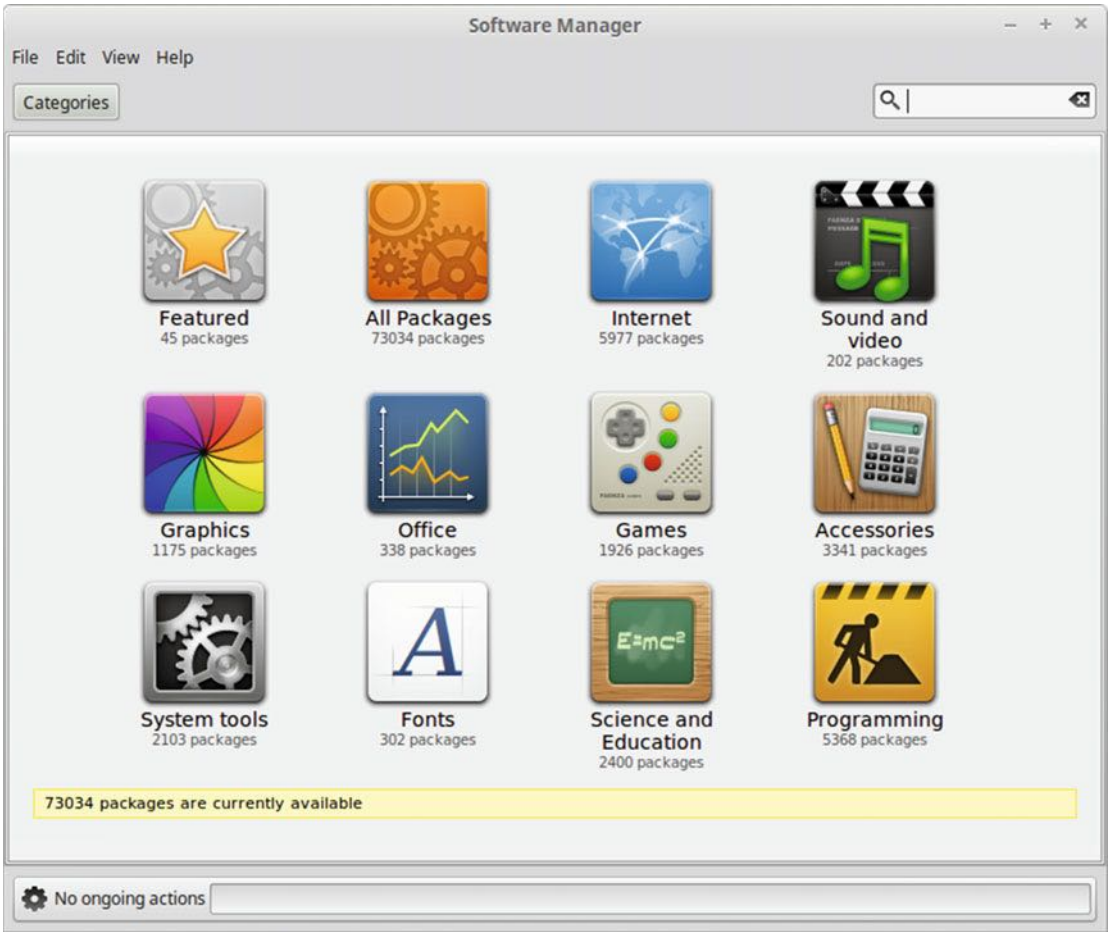


Figure 8-24. *The Software Manager application*

This application will help beginners manage their apps in an easy way, without having to deal with the terminal.

Updating and Upgrading

In Mint, the package updates and distro upgrades are managed with the same tool: Update Manager (mintupdate), shown in Figure 8-25. Linux Mint has a particular way of managing updates; it doesn't follow the same manner as Ubuntu, even though it is based on that distro and use the same packages and repository. Mint divides the updates into various levels based on function, stability, or security. This is not a popular classification system with many people because they consider kernel updates unsafe to update by default (because they can break some configurations like hardware drivers). I personally think that this is good thing for newbies to Linux and to keep the OS more stable, but it can be a security risk. If you don't like this policy, you can always configure it to show those updates; it is up to you (you can see this in Figure 8-26). Also, you can even see if a kernel version solves a particular security issues or has some regressions (Figure 8-27), and if a particular kernel version is recommended. I think this is a better way to manage things for beginners.

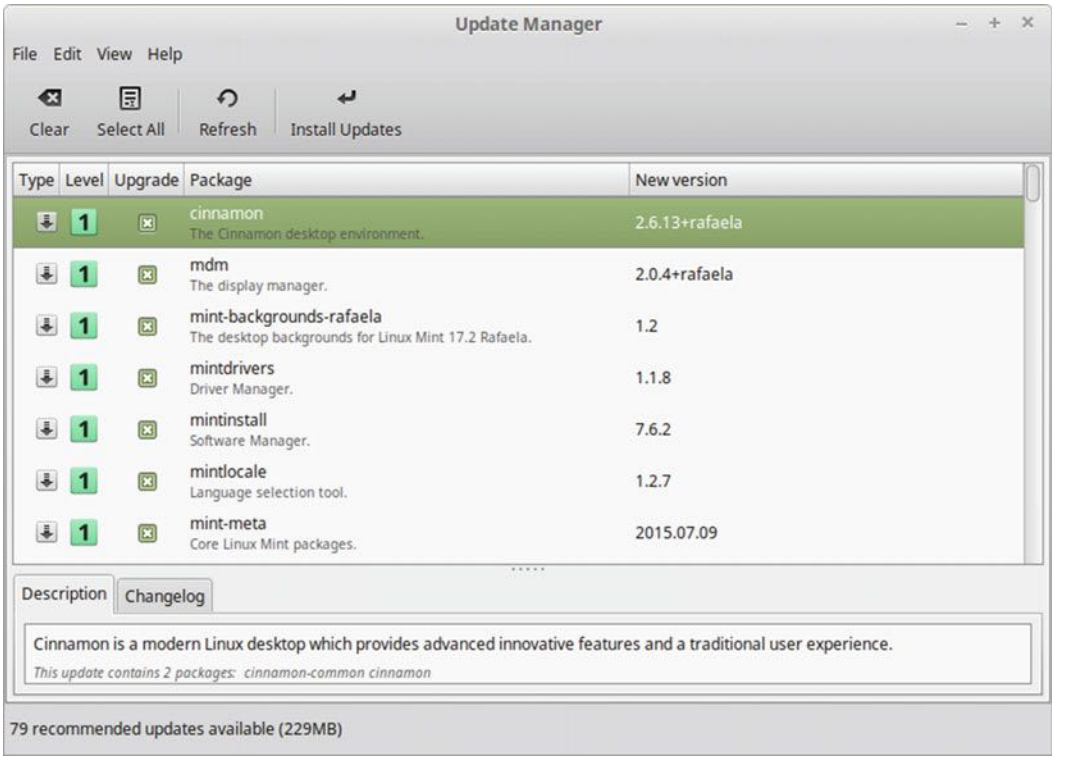


Figure 8-25. *The Update Manager application*

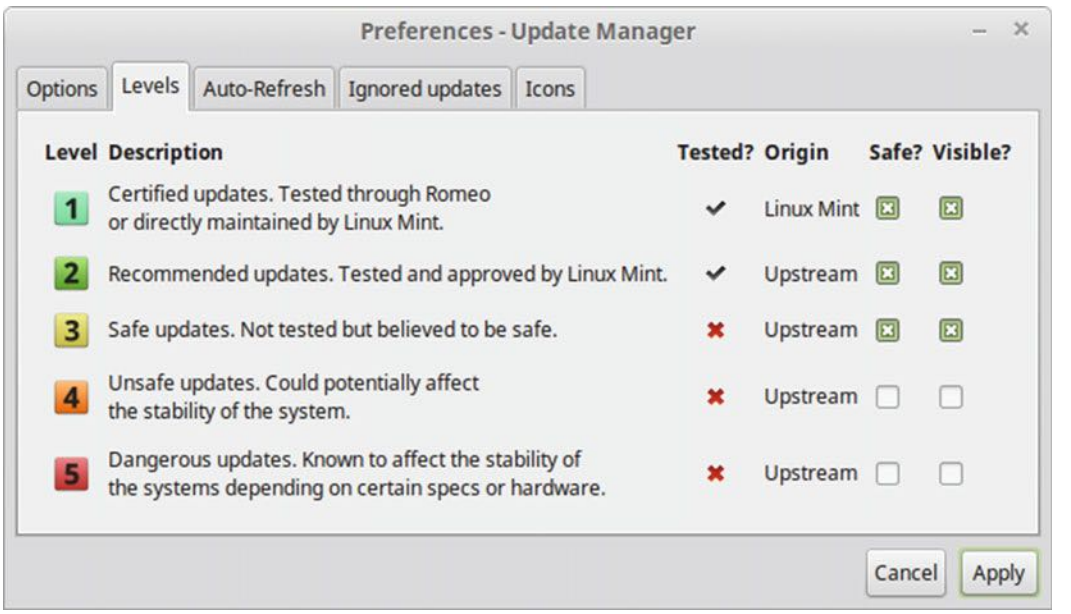


Figure 8-26. *The update levels of the Update Manager*

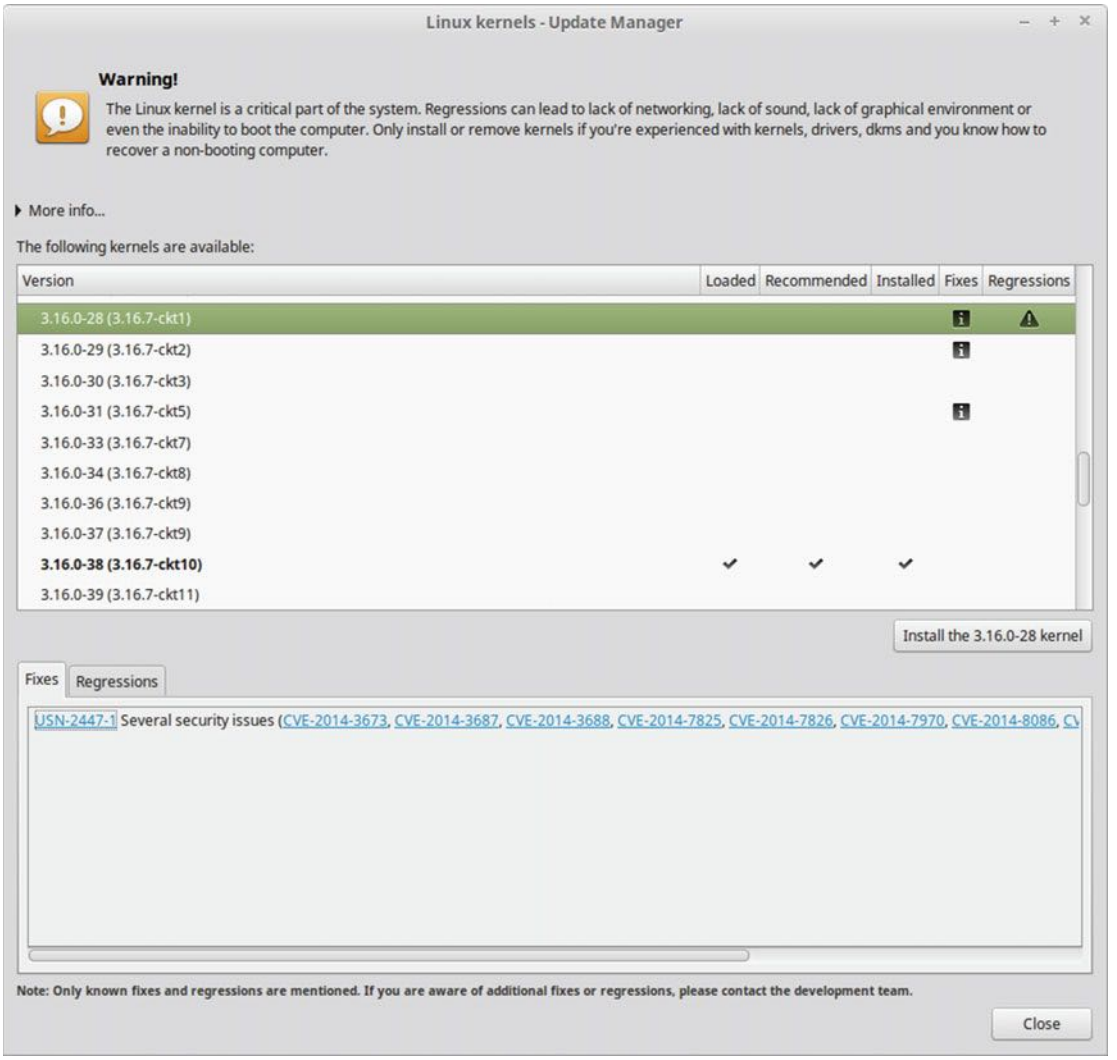


Figure 8-27. *The kernels section of the Update Manager*

The Update Manager can also deal with the distro upgrades; when a new release is available, you will see a new option to upgrade the distro in the Edit menu. If a beginner follows the Linux Mint way and does not update any unsafe update, it will be a safer upgrade than the Ubuntu regular one (but always remember to make a backup first!).

Pros and Cons

The following list contains some things that I personally see as pros and cons of the Linux Mint distribution. Of course there is always room for discussion in this matter, but I have tried to be as objective as possible.

Pros

- Linux Mint is a very easy-to-use distribution and it's recommended for beginners.
- It has its own desktop environment, Cinnamon, which is more popular than Unity and the Gnome Shell itself.
- It is based on a solid and popular distribution, Ubuntu.
- It offers community-driven development, and it follows the ideas that its community suggests.
- It offers multimedia support out of the box.
- It cares about design and aesthetics, and keeps things simple.
- It's a very stable distro, based on Ubuntu LTS releases and "safe" updates.

Cons

- Some people do not like its update levels, which imply a certain security risk.
- Linux Min is focused only on the desktop (but some people see this as advantage).
- The packages are a bit outdated.
- If you want only free software, this is not your distro.
- The Cinnamon and Mate desktops are still not perfectly polished and lack some features.

Summary

With Linux Mint you can see how an entirely community-based distro with few resources can become even more popular that its base distro. Things like using alternative desktop environments or a different approach to updates/upgrades can be enough to entice users to switch away from another distro. In all OSes, there is always room for improvement, even in the best ones, and thanks to the Free Software you can do this in Linux. Can you imagine that would happen if you could do the same with Windows 10 or OS X?

In the next chapter, you see another different approach: Mageia.