

1. INTRODUCTION TO LINUX

1.1. WHAT IS A COMPUTER SYSTEM

1.1.1. *Abstract Definition*

Depending on the user, the system is defined accordingly. It is as a magical box, which lets us to do cool stuff like browsing the Internet or playing games. In this way, it's like a VCR—by putting a tape, pressing a button, and a picture appears on ours TV. On our PC, we click the Internet Explorer icon, type a web address, and a web site somehow appears. The astonishing technical complexity behind these simple procedures isn't important to most people.

1.1.2. *Structural Definition*

The other way of looking at a PC is as a collection of components that are made by various manufacturers. We might be familiar with this way of thinking if we were ever tried to upgrade our PC's hardware. In that case, we will know that our PC consists of a CPU, a hard disk, a graphics card, and so on. We can swap any of these out to put in newer and better components that upgrade ours PC's performance or allow more data storage. What almost no one realizes is that the *operating system* is just another component of ours PC. It, too, can be swapped out for a better replacement.

1.2. WINDOWS AND ITS DRAWBACKS

Windows doesn't come free of charge, and Microsoft isn't performing a public service by providing it. Around \$50 to \$100 of the price we pay for a PC goes straight into Microsoft's pocket. Bearing in mind that hundreds of millions of PCs are made each year, it's not hard to see why Microsoft is one of the world's richest corporations.

It would be difficult to question this state of affairs if Microsoft gave us our money's worth. But it often falls far short. Its products are full of serious security holes, which at best inconvenience us and at worst make us lose data. And that's before we consider the instability of Microsoft products—hardly a day goes by without something unexpected happening. One of the first things people are taught when attending Windows training is how to use the Ctrl+Alt+Delete keyboard combination, which resets the computer after a crash!

Microsoft became rich, and maintains its wealth, by a virtual monopoly over PC manufacturers. While the intelligent computer buyer can choose between components to make for a better PC—deciding between an AMD or Intel processor, for example—we usually have little choice but to buy Windows with a new PC. Try it now. Phone ours favourite big-name computer retailer. Say that we want a PC but we don't want Windows installed. Then listen as the salesperson on the other end of the phone struggles to understand.

1.3. LINUX/GNU

1.3.1. *Linux*

Linux was created a little over a decade ago, in 1991. A decade is considered a lifetime in the world of computing, but Linux actually harks back even further, into the early days of modern computing in the mid-1970s.

Linux was created by a Finnish chap named Linus Benedict Torvalds. At the time, he was studying in Helsinki and had bought a desktop PC. His new computer needed an operating system. Torvalds's operating system choices were limited: there were various versions of DOS and something called Minix. It was the latter that Torvalds decided to use.

Minix was a freely available clone of the popular Unix operating system. Unix was used on huge computers in businesses and universities, including those at Torvalds's university. Unix was created in the early 1970s and has evolved since then to become what many considered the cutting edge of computing. Unix brought to fruition a large number of computing concepts in use today and, many agree, got almost everything just right in terms of features and usability.

Versions of Unix were available for smaller computers like Torvalds's PC, but they were considered professional tools and were very expensive. This was in the early days of the home computer craze, and the only people who used IBM PCs were business people and hobbyists.

Torvalds liked Unix because of its power, and he liked Minix because it was free and ran on his computer. Minix was created by Andrew Tanenbaum, a professor of computing, to demonstrate the principles of operating system design to his students. Because Minix was also a learning tool, people could also view the source code of the program—the original listings that Tanenbaum had entered to create the software.

Minix was lacking in some significant areas. Many people, including Torvalds, found using it very frustrating. Torvalds decided to create from scratch his own version of Minix, but to make it better, avoiding what many considered the pitfalls of Minix. He managed to produce version 0.01 of Linux in just over half a year.

From day one, Torvalds intended his creation to be shared among everyone who wanted to use it. He encouraged people to copy it and give it to friends. He didn't charge any money for it, and he also made the source code freely available. The idea was that people could take the code and improve it.

This was a master stroke. Many people contacted Torvalds, offering to help out. Because they could see the program code, they realized he was onto a good thing. Soon, Torvalds wasn't the only person developing Linux. He became the leader of a team that used the fledgling Internet to communicate and share improvements.

It's important to note that when we talk here about Linux, we're actually talking about the kernel—the central program that runs the PC hardware and keeps the computer ticking. This is all that Torvalds initially produced back in 1991. It was an impressive achievement, but needed a lot of extra add-on programs to take care of even the most basic tasks. Torvalds's kernel needed additional software so that users could enter data, for example. It needed a way for users to be able to enter commands so they could manipulate files, such as deleting or copying them. And that's before we even consider more complicated stuff like displaying graphics on the screen or printing documents.

Linux itself didn't offer these functions. It simply ran the computer's hardware. Once it booted up, it expected to find other programs. If they weren't present, then all we saw was a blank screen.

1.3.2. The GNU Project

Around the time Linus created Linux, there was another project in existence, called GNU. This project team also hoped to create an operating system that used Unix as its inspiration, although avoiding some of the pitfalls that had blighted that operating system, both technically and in terms of its licensing. GNU is a so-called recursive acronym that stands for "GNU's Not Unix," a play on words favored by computer programmers.

GNU's parent organization, the Free Software Foundation (FSF), had been formed eight years prior to Torvalds's effort, and since that time, had produced the majority of the core software that Linux desperately needed. However, as luck would have it, FSF lacked the essential functionality of the kernel. The developers were in the process of creating their own kernel, but it had not come to fruition.

The GNU software was distributed for free to anyone who wanted it. The source code was also made available so users could adapt and change the programs to meet their own needs (in fact, Torvalds had used the GNU model when deciding how to distribute Linux).

Richard Stallman is the man behind GNU and, along with Linus Torvalds, is the second accidental hero in our story. Stallman had been around since the Dark Ages of computing, back when wardrobe-sized computers were "time-shared" among users who used small desktop terminals to access them. Like Torvalds, Stallman started GNU as a personal project, but then found others who were more than willing to join his cause.

Back in Stallman's day at the legendary Massachusetts Institute of Technology (MIT), computer software was shared. If we came up with a program to perform a particular task, we offered it to practically anyone who wanted it. Alternatively, if we found an existing program wasn't adequate or had a bug, we improved it yourself, and then made the resulting program available to others. People might use ours improved version, or they might not; it was up to them.

This way of sharing software was disorganized and done on an ad hoc basis, but came about of its own accord. Nobody questioned it, and it seemed the best way of doing things. There certainly wasn't any money involved, any more than there would be money involved in one friend explaining an idea to another.

1.3.3. GNU and Linux Together

The Linux kernel, developed by Torvalds, and the GNU software, developed by Stallman, were a perfect match. It's important to note that this doesn't mean the two projects joined forces. It simply means that the Linux project took some of the GNU software and gave it a good home. This was done with Stallman's blessing, but there wasn't any official union between the two groups. Remember that Stallman had intended everyone to freely share and use the GNU tools. Linux represented a set of people doing just that. GNU is still working on its own kernel, called Hurd, which may provide an alternative to using Torvalds's Linux kernel.

GNU and Linux together formed a complete operating system, which mimicked the way Unix operated. Other projects and individuals spotted the success of Linux and came onboard, and it wasn't long before Linux realized the potential for a graphical user

interface (GUI), the fundamentals of which were provided by the XFree86 Project. A lot of additional software was also provided by individuals and organizations, all using the same "share and share alike" example set by Stallman, with the GNU tools, and Torvalds, with his kernel.

Many people refer to Linux as GNU/Linux. This gives credit to the GNU Project that provided the majority of tools vital to making Linux into a usable operating system. However, like the majority of people in the computing world, I use the term Linux throughout this book to avoid confusion.

1.4. ADVANTAGES: LINUX

People have been known to exaggerate about Linux when singing its praises, and there's certainly some hyperbole around. But there are a couple of cast-iron facts about its benefits.

1.4.1. Crash-Free

A primary benefit of Linux is that it doesn't crash. In years and years of using Linux, we will never experience our mouse cursor freezing on screen. A strange error box won't appear and not go away until we reboot. It's possible to leave a Linux system running for years without ever needing to reboot (although most desktop SUSE Linux users shut down their PC when they won't be using it for a while, just like the rest of us). Of course, programs that run on top of Linux sometimes crash, but they don't take the rest of the system down with them, as can happen under Windows. Instead, we can clean up after a crash and just carry on.

1.4.2. Security

The next benefit is that Linux is far, far more secure than Windows. Linux is based on years of proven computer science research. It works on the principle of users who have permissions to undertake various tasks on the system. If we don't have the correct permission, then we cannot, for example, access a particular piece of hardware. Additionally, privacy can be ensured because the files on the PC are "owned" by individual users, who can permit or deny others access to those files.

1.4.3. Free and Shareable

Another big benefit is that Linux can be obtained free of charge. Once it's installed, the latest updates for all our programs are also free of charge. Not only that, but if we want any new software, it will also usually be free of charge (and normally just a download away). Is this starting to sound attractive yet?

Because the software is free, we can share it with friends. Suppose that we find a really great image editor. We mention it to a friend, and he asks for a copy. Under Windows, copying the program is strictly illegal—to do so turns us into a software pirate! Unless that image editor is freeware, our friend will need to buy the software himself. Under Linux, sharing software is normally entirely legal. In fact, it's encouraged!

1.4.4. Compatible with Older Hardware

Another benefit of Linux is that it works very well on older hardware and doesn't require a cutting-edge PC system. The latest version of Windows XP requires high-

powered hardware, to the extent that upgrading to that operating system usually means buying a new PC, even if ours old one still works fine!

In contrast, Linux works on computers dating back as far as the early 1990s. This book is largely being written on a five-year-old Pentium II 450 MHz notebook, running SUSE Linux 9.1. There's virtually no waiting around for programs to start. On the same machine, Windows 2000 (which came installed on the computer) grinds and churns, and using it can be a frustrating experience.

With Linux, there's software for just about every type (or age) of computer. For example, we'll find stripped-down graphical user interfaces (GUIs) that are designed specifically for old computer hardware. Linux encourages an attitude of both recycling and making the most of what we have, rather than constantly upgrading and buying new hardware.

In other words, we can pull out that "old" PC and bring it back to life by installing Linux. We might even be able to give it away to a family member or friend who does not have a PC. Perhaps it's time for grandma to get online, or perhaps we can give the kids their own PC so they will stop using yours.

Alternatively, we might consider turning old hardware into a server. Linux is capable of just about any task. As well as running desktop computers, it also runs around 60% of the computers that make the Internet work. Linux is extremely flexible. We could turn an old PC into a web, e-mail server, or firewall that we can attach to a broadband Internet connection. If we were to do this with Microsoft software, it would cost hundreds of dollars, not to mention requiring an advanced computer. It's free with Linux.

1.5. VERSIONS/TYPES OF LINUX

All the pieces of GNU software were available for free download and were therefore free of charge. But this brought its own problems. Not everyone had the know-how to put all the bits and pieces together into a complete operating system. Those who could do this didn't necessarily have the time for it.

Because of this, a number of companies stepped in to do the hard work. They put together versions of Linux, complete with all the software from the GNU Project, which they then sold for a fee on floppy disks, CDs, or DVDs. They also added in bits of their own software, which made it possible to install Linux easily onto a computer's hard disk, for example. They produced their own manuals and documentation, too, and did other things such as bug testing to ensure it all worked well.

What they came up with became known as distributions of Linux, or distros for short. Examples of these companies include Red Hat, SUSE, Mandrake, and many others around the world. Additionally, a number of hobbyists got together and formed organizations to create their own distros, such as Debian and Slackware.

Modern distros are very advanced. They make it easy to install Linux on ours PC, and they usually come with everything we need, so we can get started immediately. Additionally, they have their own look and feel, as well as unique ways of working and operating. This means that SUSE Linux is not the same as Red Hat Linux, for example, although they share a lot of common features and, of course, they all share the core GNU software.