

# Chapter 1

## Introduction & History of \*NIX

So you've decided to take the dive into the world of UNIX system administration? An excellent choice. The UNIX operating system has long held the position of the environment of choice for serious work. Whatever field you look into, from computer animation to fusion reactor simulations, you will find that most of the work is being done on UNIX-based systems. Before we dive in, let's first go over a brief introduction to what UNIX is and how it got to where it is today.

### 1.1 What Is It?

UNIX is a general purpose operating system. It falls under the same software category as Microsoft Windows and Mac OS X. That said, UNIX was not made to be a Windows 'clone'. Beginners often feel overwhelmed by the sheer weight of UNIX and often have trouble comprehending why it behaves so different from Windows. They see the command line with cryptic commands and options and wonder "Windows isn't like this. Why would people design a system like this?"

There is a reason for this. UNIX wasn't originally designed for the entire world. It was created by a group of programmers to run their own programs. It wasn't created to compete with companies like Microsoft. However, the fact that it has managed to gain widespread acceptance without any serious marketing effort (have you ever seen a UNIX commercial?) just goes to show you how powerful it is.

### 1.2 Why Should You Learn It?

So here I am, telling you that UNIX is this amazing system, but why should you need to learn it. You probably already know how to use Windows, isn't

that enough? Well, the answer to that depends on how much control you want over your computer. If you are fine with Microsoft making all the decisions for you, then I guess you can stay with Windows. But if you are reading this, odds are that's not how you feel.

### 1.2.1 GUIs Are Limited

In operating systems like Windows, the user interacts with the operating system through a graphical user interface (GUI). When you go to configure something, you go to the Control Panel and open an icon. Windows then gives you a pretty form with some fields that you can change, and when you click OK, it does everything for you. In Windows, what you see is what you get. However, as is often the case with such systems, what you see is **all** you get. You're going to have a tough time trying to do some custom configurations if the people at Microsoft didn't happen to anticipate what you might be planning and build those options into the GUI. This is fine for most people since the normal options are all normal people will need, but we are system administrators, not normal people.

What's actually happening behind the scenes is that the GUI is taking the input you gave it and is forming commands to give the operating system. Unfortunately, the user often has no idea what these commands are or where they are being sent, so if you wanted to create some custom commands, you wouldn't know how to form them or where to send them. No such problem exists in UNIX. Since all of your interactions with the system can be based on such commands, once you learn how the commands work, you can use them with any options and in any combination to accomplish your task.

I don't mean to say that \*NIX doesn't have GUIs. There are excellent desktop environments available (e.g. GNOME and KDE) with amazing eye candy that surpasses OSX and Windows (e.g. XGL/Beryl). With these environments, it's possible to use \*NIX without every knowing what a command line is. However, unlike Windows, where the GUI is integrated into the OS and is required for normal operation, in UNIX, the GUI is a completely optional package. If you like using a GUI, you are free to install whatever desktop environment you like, but it's not required. UNIX runs the same without a desktop installed (better actually, because the system doesn't have to deal with generating graphics).

### 1.2.2 Understanding the Computer

GUI based systems like Windows try to keep the user as far away from the inner workings of the computer as possible. You as a user interact with the GUI and everything beyond that is accomplished by the magic smoke that powers all computers. While there are benefits associated with keeping users away from the inner workings of the system (mainly, keeping them from screwing it up), it also limits your understanding of the system. This in turn limits what you are able to do with it.

Learning UNIX system administration will help you to understand how the components of the system are laid out and how they work together. Once you understand how the components work together, you will be better equipped to take full advantage of the power of the operating system. The understanding of the operating system also allows you to gain insight into related systems. For instance, once you learn about network interfaces and how to configure networking on your system, you will have a better understanding of how networking works in general.

### **1.2.3 The Major Players Use UNIX**

As stated earlier, UNIX is the environment of choice for getting real work done. If you are hoping to do any serious work with computers, you will more than likely be doing it on a UNIX based system. From computer animation to aerospace simulations to web hosting, the preferred operating system is UNIX. In order to be successful in such fields, it is important that you understand the system on which most of the work is done.

## **1.3 Why Should You Actually Like It?**

People tend to see the command line and the cryptic commands and think “Why am I being forced to learn this stuff? Why couldn’t Windows be the industry standard instead of UNIX?” Well, the simple answer is that there is an incredible amount of power in the UNIX commands and the command line you run them from.

### **1.3.1 Built-In Scripting Language**

The ‘command line’ is the prompt you get when you log into a UNIX system. Its the interactive part of the ‘shell’ program that is started as soon as you log in. The command line gives you complete control of your system, so long as you know the commands to give it. However, once you learn the commands, it doesn’t end there. The shell, while having many features to make interactive use easy, also contains features that make it an excellent scripting language. Do you find your self typing in the same commands over and over? Well dump the commands into a file and run it as you would any other command. The shell scripting facilities make this and more complex tasks easy because you have already learned the language.

### **1.3.2 Regular Expressions**

The UNIX shell and many other utilities support a pattern matching feature called regular expressions. While there is a somewhat steep learning curve associated with this system, once you learn it you will be able to save a lot of time while editing files and write scripts and programs to manipulate data with ease.

### 1.3.3 Automation

The commands in UNIX are, for the most part, non-interactive. Unlike Windows, you don't start a program and click through 10 menu selections and prompts to get things done. You give the shell a command and the shell executes it. If there is an error, it tells you. If not, it just prompts for your next command. This sort of system lends itself to the development of automated systems. Things that just shouldn't be done in Windows can be done in no time in UNIX.

For example, consider the task of converting a thousand BMP files to JPG files for use on the web. If you ask a Windows user to do it, the reaction won't be pleasant. He will think of the tens of thousands of times he will have to click through the same interactive windows of some graphics program, cringe in disgust, curse at you for giving him such a task, and run away. However, if you ask the knowledgeable UNIX user, he will say "sure, no problem", write 3-4 commands, and be done with it.

### 1.3.4 Its Free

Tired of getting completely h0zed by Microsoft? Every couple of years they come out with the new 'latest and greatest', and you're forced to shell out \$200 for a new version of Windows and \$300 for a new version of Office. Every time you want to add software to your machine you have to shell out another \$100+ just for the cheap programs.

UNIX is complete free. This is the free as in 'free beer', free. If you go to a frat party which has free beer, you get beer without having to pay anything. Such is the same with UNIX. You can download it, install it, and run it without paying a dime. The programs that come with it, and those that come from the UNIX community, are also free. Need to work on an Microsoft Word document? The community has created a free Office clone called 'OpenOffice'. This program has the ability to read and write Microsoft Office documents in addition to having many of the same features of Microsoft Office. OpenOffice actually offers an additional type of freedom, which will be discussed in the next section.

## 1.4 How Did It Get Where It Is Today?

### 1.4.1 In The Beginning

UNIX was originally designed in the 60's at Bell Labs. Though it was originally written in machine specific assembly code, it was later rewritten in C code, allowing it to be easily ported to other platforms and architectures. It turned out to be quite popular because it was well-designed and had a set of very innovative features that no other operating system had at the time.

### 1.4.2 The Clone Wars

Unfortunately, UNIX was proprietary. It was like Microsoft Windows today – you had to buy a license to use the software, you couldn’t modify it, and you couldn’t distribute it. As a result, companies and universities began making UNIX-clones. Eventually there were so many clones that UNIX was no longer the name of a single operating system, but the name of a category of operating systems. This became a problem, because programmers would write code for one UNIX system and wouldn’t be able to run it on others. In order to solve this, all the clones got together and decided to make a standard for UNIX compatibility called POSIX (Portable Operating System Interface). Thanks to this, as long as the code they wrote was POSIX compliant, it would work on any POSIX compliant system.

### 1.4.3 Solaris

Sun’s Solaris is one of the most popular UNIX clones today. It has a number of (primarily technical) features and enhancements over most UNIX clones. Unfortunately, up until a couple of years ago, Solaris was still a proprietary version of UNIX – it was just as limited, in terms of freedom (which we will talk about in the next section), as the original version of UNIX. Nevertheless, Solaris is one of the most reliable and powerful versions of UNIX out there today, and many companies use it to run their infrastructure. Solaris administrators are the highest paid administrators out there.

### 1.4.4 BSD

Berkeley Software Distribution, or BSD, is a UNIX clone that was developed here, at UC Berkeley, back in the 80’s. BSD’s main benefit compared to all the other UNIX clones is that it’s completely free, in both senses of the word. In addition to being free as in ‘free beer’, it is also free as in ‘freedom’. You don’t have to buy a license to use BSD, you can modify and improve it in any way you see fit, and you can redistribute it any way you want (with minor restrictions). In fact, there is some BSD code in versions of Microsoft Windows 2000 and newer. Mac OS X is also based on a UNIX core called Darwin, which is a derivative of BSD. Unfortunately, BSD has been declining in popularity lately, in favor of Linux.

### 1.4.5 GNU, Linux, and the Free Software Foundation

The Free Software Foundation is a non-profit corporation founded by Richard Stallman in October 1985 with the goal of providing completely ‘free’ software, in both senses of the word. The FSF provides the public license (GPL) under which free software is published. The GPL grants the recipients of any program released under the GPL the right to freely obtain, modify, and redistribute the program as they see fit.

GNU is a UNIX-like operating system consisting of entirely of free software. The name GNU is actually a recursive acronym for “GNU’s Not Unix”, so chosen because while it is UNIX-like, it differs from UNIX because it is made of entirely free software and doesn’t use any of the original UNIX code. GNU was founded by Richard Stallman and was the original focus of the Free Software Foundation. The GNU system contains everything needed for an operating system except for a working kernel. The official kernel the GNU Project is called GNU Hurd, but it is still being developed, slowly. To make a complete system, most people use the third-party Linux kernel. The combination of the GNU Project and the Linux kernel is referred to as “GNU/Linux”, though it is often shortened to “Linux”.

Linux has acquired a rapidly growing community of users and contributors. Like BSD, Linux is free to everyone, and anybody can modify it according to their needs (there are slight differences in the licensing between BSD and the GPL). Linux has found its way into all sorts of devices, including PDAs, cell-phones, routers, etc.

Unfortunately, Linux generally sacrifices stability for flexibility, and, as a result, is not viewed as a very reliable UNIX platform (compared to Solaris). Nevertheless, Linux is favored by computer scientists and system administrators because of its rapid development and bleeding-edge feature-set.