

UNIX as an Operating System

What is an Operating System (OS)?

The operating system (OS) is the program which starts up when you turn on your computer and runs underneath any other programs you run - without it nothing would happen at all. In simple terms, an operating system is a manager. It manages all the available resources on a computer from the CPU to memory to hard disk accesses.

One could break down the overall tasks that an OS must perform into 3 main categories...

- **Control Hardware** - The operating system controls all the parts of the computer and attempts to get everything working together.
- **Run Applications** - Another job the OS does is run application software. This would include word processors, web browsers, games, etc...
- **Manage Data and Files** - The OS makes it easy for you to organize your computer. Through the OS you are able to do a number of things to data including copy, move, delete, and rename it. This makes it much easier to find and organize what you have.

UNIX History

The UNIX operating system was born in the late 1960's. It originally began as a one man project lead by Ken Thompson of Bell Labs and has since grown to become the most widely used operating system.

In the time since UNIX was first developed, it has gone through many different generations and even mutations. Some differ substantially from the original version, like Berkeley Software Distribution (BSD) or Linux. Others, still contain major portions that are based on the original source code.

An interesting and rather up to date timeline of these variations of UNIX can be found at <http://www.levenez.com/unix/history.html>.

UNIX as an Operating System (OS)

In general most UNIX operating systems have the following characteristics...

- **Multi-user & Multi-tasking** - most versions of UNIX are capable of allowing multiple users to log onto the system each capable of running multiple tasks. This is standard for almost any OS now-a-days.
- **Over 30 Years Old** - there is something to be said for the fact that UNIX is over 30 years old and it's popularity and use are at all time highs. Over these 30 years many variations have spawned off and many have died off, but much of today's modern UNIX systems can be traced back to the original versions. It has endured the test of time. For reference, Windows at best is half as old (Windows 1.0 was released in the mid 80's, but many would argue that it was not by any means stable or complete enough until the 3.x family which was released in the early 90's).
- **Large Number of Applications** - although it may not seem it, there is an enormous amount of applications available for UNIX operating systems. They range from commercial applications such as CAD, Maya, Word perfect, to many free applications.
- **Free Applications and Even Free Operating System** - of all of the applications available under UNIX, many of them are free. The compilers and interpreters that we use in most of the programming courses now these free versions that can be downloaded off of the Internet free of charge. As well, most of the development that we do in programming courses is done under the Linux OS. Since Linux is one of the free versions of UNIX, it too can be downloaded and installed free of charge.
- **Less Resource Intensive** - in general, most UNIX installations tend to be much less demanding on the system resources. In many cases the old family computer, that can barely run whatever version of windows is more than sufficient to run the latest version of Linux.
- **Internet Development** - Much of the backbone of the Internet is run by UNIX servers of one type or another. Many of the more general web servers out there also run UNIX with the Apache webserver - another free application.

Parts of the UNIX Operating Systems

Most UNIX systems can be broken down into several parts...

- **The Kernel** - handles memory management, input and output requests, and program scheduling. Technically speaking, the UNIX kernel "is" the OS. It provides the a basic software connection to the hardware. The kernel is very complex and deals with the inner workings of these things, and is thus beyond the scope of this course.

- **The Shell** - basic UNIX shells provides a "command line" interface which allows the user to type in commands which are translated by the shell into something the kernel can comprehend and then is sent off to the kernel for it to act upon.
- **The Built-in Utilities** - are programs that allow a user to perform tasks which involve complex actions in one fell swoop. Utilities provide user interface functions that are basic to an operating system but which are too complex to be built into the shell. Examples of utilities are programs that let us see the contents of a directory, move & copy files, and remove files.

Also, not part of the operating system (OS) directly, but common on almost all systems are...

- **Installed Software & Utilities** - are additional programs that are not bundled with the operating system (OS) distribution. These can range from additional or different versions of basic utilities, to full scale commercial applications.

Flavors of UNIX

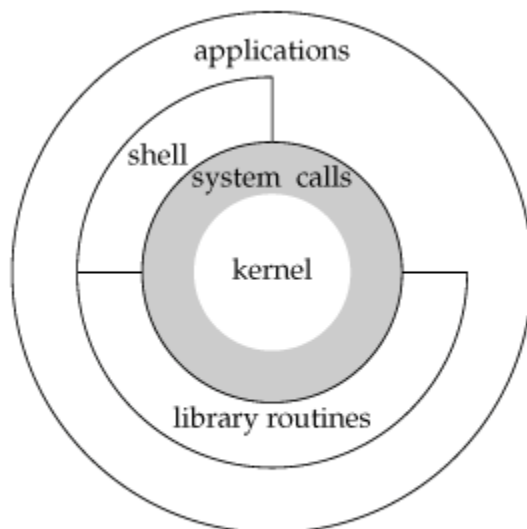
As I have mentioned, over the 30 years that UNIX has been around there have been many different variants. There just as many ways to categorize these variations. For the purpose of this course, we will classify into 2 sets of categories...

- **Open Source vs. Proprietary –**
- **Open Source** means that the source code that is the operating is readily available and is free for all to install and use.
- **Proprietary** means the software's use, redistribution or modification is prohibited, or requires you to ask for permission, or is restricted so much that you effectively can't do it freely (meaning it was developed for sale and profit, and thus you should pay for it).
- **Proprietary UNIX Examples**
- [Solaris](http://solaris.gl.umbc.edu) - Access to a Solaris UNIX server (solaris.gl.umbc.edu) via SSH access.
- [IRIX](http://irix.gl.umbc.edu) - There are a couple of IRIX machines in the basement of the library, as well as server access (irix.gl.umbc.edu) via SSH access.
- [Mac OS X](http://macosx.gl.umbc.edu) - no access currently available at UMBC.

- and many others...
- **Open Source UNIX Examples**
- [Free BSD](#)
- [Linux](#)
 - [RedHat](#) - RedHat is also one of the most popular linux distributions.
 - [Caldera](#)
 - [Mandrake](#)
 - [Debian](#)
 - [SuSE](#)
 - [Slackware](#)
 -
- and many others...

UNIX ARCHITECTURE

In a strict sense, an operating system can be defined as the software that controls the hardware resources of the computer and provides an environment under which programs can run. Generally, we call this software the kernel, since it is relatively small and resides at the core of the environment. Figure given below shows a diagram of the UNIX System architecture.



Architecture of the UNIX operating system

Architecture of the UNIX operating system

The interface to the kernel is a layer of software called the system calls (the shaded portion in Figure). Libraries of common functions are built on top of the system call interface, but applications are free to use both. (We talk more about system calls and library functions in next assignments.) The shell is a special application that provides an interface for running other applications. In a broad sense, an operating system is the kernel and all the other software that makes a computer useful and gives the computer its personality. This other software includes system utilities, applications, shells, libraries of common functions, and so on.

For example, Linux is the kernel used by the GNU operating system. Some people refer to this as the GNU/Linux operating system, but it is more commonly referred to as simply Linux. Although this usage may not be correct in a strict sense, it is understandable, given the dual meaning of the phrase operating system.

WINDOWS AND UNIX

As far as operating systems go, to some it would seem as if UNIX has a clear advantage over Windows. UNIX offers greater flexibility than Windows operating systems; furthermore, it is more stable and it does not crash as much as Windows. To some, UNIX is just as easy to use as Windows, offering a GUI interface as well as command line. But there are users out there that believe UNIX is for only for computer gurus only, claiming that the fragmentation of the UNIX GUI is its greatest competitive weakness.

One thing that has been established though, UNIX is quite a bit more reliable than Windows, and less administration and maintenance is needed in maintaining a UNIX system. This is a huge cost saver for any organization. Rather than employing many individuals to maintain a Windows based system, one part-time employee would be needed for the upkeep of a typical size UNIX system. One key difference between UNIX and Windows is the implementation of multiple users on one computer. When a user logs onto a UNIX system, a shell process is started to service their commands. Keeping track of users and their processes, a UNIX operating system is able to keep track of processes and prevent them from interfering with each other. This is extremely beneficial when all the processes run on the server, which demands a greater use of resources - especially with numerous users and sizeable applications.

Another main difference between UNIX and Windows is the process hierarchy which UNIX possesses. When a new process is created by a UNIX application, it becomes a child of the process that created it. This hierarchy is very important, so there are system calls for influencing child processes. Windows processes on

the other hand do not share a hierarchical relationship. Receiving the process handle and ID of the process it created, the creating process of a Windows system can maintain or simulate a hierarchical relationship if it is needed. The Windows operating system ordinarily treats all processes as belonging to the same generation.

UNIX uses daemons, Windows has service processes. Daemons are processes that are started when UNIX boots up that provide services to other applications. Daemons typically do not interact with users. A Windows service is the equivalent to a UNIX daemon. When a Windows system is booted, a service may be started. This is a long running application that does not interact with users, so they do not have a user interface. Services continue running during a logon session and they are controlled by the Windows Service Control Manager.

UNIX has a novel approach to designing software. Since UNIX is open-sourced, it attracts some very intelligent programmers who develop many applications free of charge. With this in mind, many designers choose to resolve software problems by creating simpler tools that interconnect rather than creating large application programs. In contrast, Windows applications are all proprietary and costly. With UNIX, each generation extends, rather than replaces the previous like Windows it is rarely necessary to upgrade - old and new Unix are all compatible. The main reason for this is the way UNIX is built, which is on a solid theoretical foundation. There are many advantages to this, for instance, a book written 20 years ago that discusses programming UNIX can still be used today. Imagine trying to figure out how to run Windows XP with a Window 3.1 manual - it can't be done.

One argument to be made about UNIX is its lack of standardization. Some feel there are too many choices to be made regarding which GUI to use, or which combination of UNIX hardware and software to support. UNIX operating systems make great high-performance servers, but for end-users, every application on each arrangement of UNIX platform requires a different set, and each application has a different user interface. Microsoft has "the" Windows operating system; there simply isn't one standardized UNIX operating system, or for that matter, a single standardized UNIX GUI. One could argue and say this is a downfall for UNIX, but on the other hand, these variations add flavor and versatility to a solid, reliable operating system.

In summary, the best way to choose between UNIX and Windows is to determine organizational needs. If an organization uses mostly Microsoft products, such as Access, Front Page, or VBScripts, it's probably better to stick with Windows. But, if reliability, universal compatibility, and stability are a concern, UNIX would probably be the way to go.

Simply stated, the main difference is Windows uses a GUI (Graphical User Interface) and UNIX does not. In Windows one uses the click of a mouse to execute a command where as in UNIX one must type in a command. There are GUIs that can be used in a UNIX environment though very few UNIX users will stoop that low to use one.) Before there was a Windows environment, DOS (Disk Operating System) was used on PCs. DOS was based on and was similar, but only a poor subset, to the UNIX system.

Differences between UNIX and WINDOWS: Unix is safe, preventing one program from accessing memory or storage space allocated to another, and enables protection, requiring users to have permission to perform certain functions, i.e. accessing a directory, file, or disk drive. Also, UNIX is more secure than Windows on a network because Windows is more vulnerable than UNIX. For example, if you leave a port open in Windows it can be easily used by a hacker to introduce a virus in your environment.

Unix is much better at handling multiple tasks for a single user or for multiple users than windows. For each user, Unix in general, and especially Sun's Solaris provides many more utilities for manipulating files and data than windows does. For a corporate environment, Unix (especially Solaris) provides much more control for the administrator than windows does. Solaris, for example, enables the administrator to mirror or stripe data across several disks to minimize risk or optimize performance without 3rd party products. In general, for a programmer or for an administrator, Unix provides more power and flexibility than windows. For the less sophisticated user, Windows can often more easily be installed and configured to run on cheaper hardware to run a desired 3rd party product. In short -- Unix is better, Windows is easier for less sophisticated users.