

Post01 - Unix

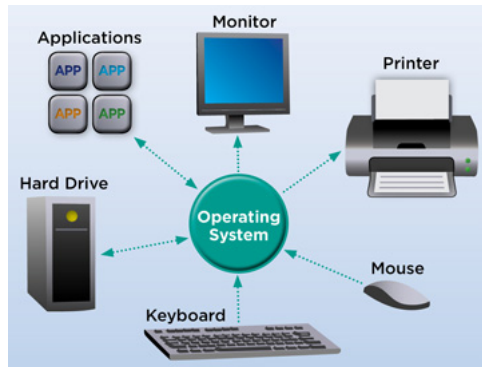
Haibo Fan

The majority of data processing and analysis happens on computers. Knowing how to write functions or manipulate data tables is relatively common (though necessary) knowledge. However, the mechanisms behind the working of those functions and commands - or even the system itself used to analyze data - is not as well known. This post will talk about what's behind the code; more specifically, about the machine that interprets and executes human input. While there are different standards and protocols for handling such input, I will mostly focus on Unix.

Introduction: What is Unix?

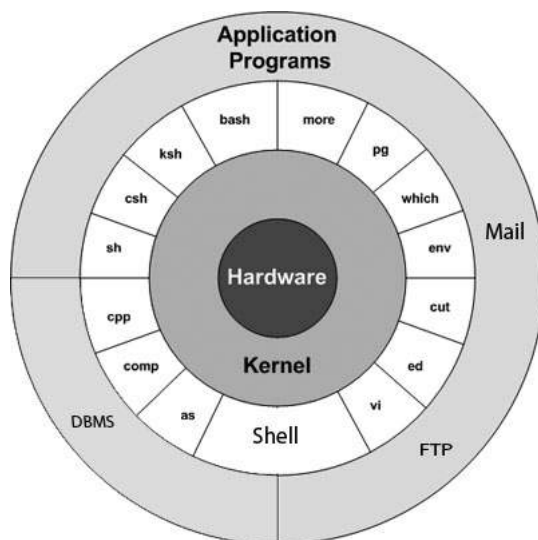
Some History and Background

For a computer to function, it must run on an operating system, which is like an overarching support structure for all of the computer's capabilities. ^[1]



Unix is one operating system, developed around the late 1960s and early 1970s at AT&T Bell Labs and primarily attributed to Ken Thompson and Dennis Ritchie. ^[2] Over time, since it was not at the time owned by any large computing company, it became the first open source OS that basically anyone could acquire and modify. Today, there is a brand called "UNIX" which is a set of specifications for the OS (mostly for the kernel, explained below), owned by The Open Group. ^[3] Since Unix is open-source, there are multiple distributions available (sort of like "flavors"), of which The Open Group decides whether or not to certify them as truly "UNIX". Those distributions not entirely conforming to the UNIX specifications are instead called "Unix-like."

The Two Basic Parts of the Unix OS ^[4]



Strictly speaking, every OS has these parts. But since Unix is an OS, it also has these parts, and a discussion about Unix would thus not be complete without mentioning them.

1. the Kernel ^[5] ^[6]

The kernel of the OS is at the core, interfacing between digital commands and hardware, such as by scheduling or queuing commands and determining how much memory to allocate to certain tasks. It loads at the beginning of the computer's startup, recognizing certain key components (such as the processor), and continues running even when the computer is fully booted up, constantly handling system requests and appropriately assigning hardware space.

2. the Shell ^[7] ^[8]

The shell is one of the most basic user interfaces to the OS - that is, it allows the user to input commands and the like into the operating system. It is named the "shell" because it is the outermost layer of the kernel, essentially communicating between the user and the kernel by arranging commands to be carried out. It is generally a command-line interface, necessitating knowledge of the right commands and syntax by the user to actually perform tasks. Since the shell is right above the kernel, it provides the user a great amount of control over the kernel's tasks. ***

2.a. More about the Shell: Bash and Git

In Stat 133, Git Bash is an often-used tool. But what exactly is it?

Well, Bash is actually just a Unix shell ^[10] - that is, a shell designed for Unix systems. It is the default shell for Linux, a widely popular Unix distribution.

Git is a type of *version control software*, which simply means that it keeps track of changes, specifically for files, sort of like a history for files. ^[11] It is free software and was developed for use with the Linux kernel.

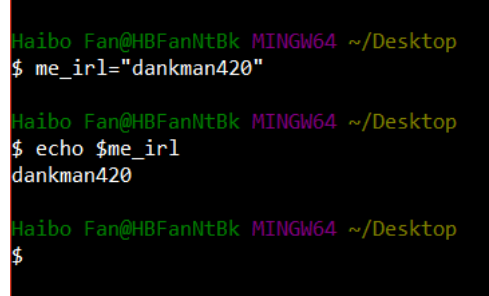
Git Bash is simply software for Git - version control software - packaged with a Bash shell to allow user interface with Git. ^[12] Git was originally developed for Linux, and Bash is the default shell for Linux, so using Bash to interface with Git is reasonable. Since not all OS's run Linux, Git Bash provides an emulation of the Linux kernel and an accompanying shell to allow everyone to work in a Unix-like environment.

2.b. Some Examples of Bash code ^[13]

While in Stat 133 Bash is mostly used to just push and pull files from repositories, Bash is by itself a coding language, just like R or Python, with expected capabilities.

Due to technical difficulties, I can only include screenshots of my Bash console, and not the code itself in this post.

Here, it is possible to assign values to names, then call those names again, using the command “echo” to display the output:

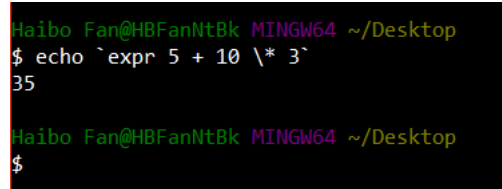


```
Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$ me_irl="dankman420"

Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$ echo $me_irl
dankman420

Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$
```

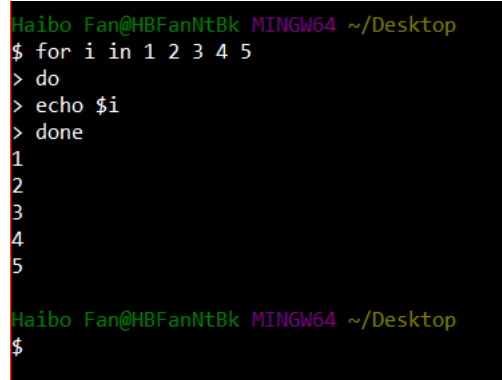
Basic arithmetic is also supported; here is evaluation of $5 + 10 * 3$.



```
Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$ echo `expr 5 + 10 \* 3`
35

Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$
```

Even control flow is possible, as demonstrated by this for loop which prints the integers from 1 to 5 inclusive:



```
Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$ for i in 1 2 3 4 5
> do
> echo $i
> done
1
2
3
4
5

Haibo Fan@HBFanNtBk MINGW64 ~/Desktop
$
```

But why Unix?

So far I've only described OS's in general, with some examples specific to Unix, such as Git and Bash. So why is Unix special?

While Windows is the most popular operating system for most people, Unix is preferred for computing. There are numerous possible reasons: its long history, being developed at a leading scientific research institute; its open-source nature, allowing anyone to customize it to their needs.

How-To Geek summarizes two nice properties of Unix: it has a philosophy of creating small utilities that are especially good at performing specific tasks, allowing large complex tasks to be broken down into smaller ones by way of piping; and Unix only has “files”, no directories like Windows, giving a standard that allows programs to easily communicate with one another. ^[14]

Unix is also known for its command-line interface (CLI), as opposed to a graphical user interface (GUI) like Windows uses. ^[15] As mentioned above, this, combined with the small utilities and piping, allows the user very fine control over what operations he or she wants the computer to perform. A CLI is closer to a shell than a GUI, thus allowing finer control over kernel functions

In general, Unix is also more stable than Windows and less prone to viruses (its open-source nature allows anyone to inspect for potential weak points), as well as being free (for certain distributions, such as Linux), while operating Windows requires a purchase. ^[16] As a student at the number one public university in the world, I can empathize with those who would rather use a free OS than a commercial one.

In addition, Unix's long history means that its architecture and design are very well understood and multiple tools and utilities have been developed for it over the years.

Conclusion

So what is Unix? Unix is an OS, and one with a lengthy history at that, developed at a leading science institute and becoming the multiflavored family of distributions it is today.

Though this post describes some of the basic components of the Unix OS and common tools used in conjunction with it, that does not by itself set it apart from other OS's. What does is its design philosophy of simplicity and long history, providing its users with numerous pre-developed tools to use and experiment with, gives the user finer control over computing and programming than Windows can. Thus it is a common choice for programmers nearly everywhere.

References

- [1] https://www.webopedia.com/TERM/O/operating_system.html³
- [2] <http://searchdatacenter.techtarget.com/definition/Unix>
- [3] http://www.unix.org/what_is_unix.html
- [4] <http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html>
- [5] <https://techterms.com/definition/kernel>
- [6] <http://searchdatacenter.techtarget.com/definition/kernel>
- [7] <https://techterms.com/definition/shell>
- [8] [https://en.wikipedia.org/wiki/Shell_\(computing\)](https://en.wikipedia.org/wiki/Shell_(computing))
- [9] <https://www.tutorialspoint.com/unix/unix-getting-started.htm>
- [10] [https://en.wikipedia.org/wiki/Bash_\(Unix_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))
- [11] <https://en.wikipedia.org/wiki/Git>
- [12] <https://git-for-windows.github.io/>
- [13] <https://www.tutorialspoint.com/unix/unix-using-variables.htm>, and "Next Page"'s
- [14] <https://www.howtogeek.com/182649/htg-explains-what-is-unix/>
- [15] <https://www.quora.com/How-is-UNIX-different-from-Windows>
- [16] <https://www.lifewire.com/operating-systems-why-unix-2200178>