UNIX : PROS and CONS

UNIX is an operating system that is most commonly found in universities, research laboratories and large government institutions. It was written in 1969 by Kenneth Thompson at Bell Labs. It differed from other operating systems of the time in several ways. Unlike most of the other operating systems, UNIX was free, as well as machine independent, meaning that it could be run on any machine.

UNIX Pros

UNIX has a lot of pluses besides being free and runnable on almost any machine. You have more control over the computer with UNIX. Operating systems like Mac OS and Windows are icon and mouse based; using them, you can activate only what the operating system allows you to activate. On UNIX, you can do any of the things that the operating system can do. This means, or course, that you can really mess things up if you don't know what you are doing. Another freedom, and danger, that you have with UNIX is that you can change the operating system to make it more compatible with what you want to do. If you know what you are doing, UNIX can be powerful and customizable, and now that it has been around for such a long time, most of the bugs are out and it is very reliable.

UNIX is multi-user

UNIX is multi-tasking

UNIX provides file security

UNIX is non-hardware dependant (portable)

UNIX does not impose file format or naming

UNIX provides powerful scripting language

UNIX provides program development kits (C/C++)

- Very capable and stable OS. Unix/Linux servers have been known to have continuous uptimes in the order of decades!
- Very fine level control on nearly every aspect of the system. You can control the behvaiour of the system in a very detailed manner.
- Greater security - On linux/unix, only the root user can make system level changes, unlike windows. Linux is also generally safer.

- Open source freedom - With loads of open source software available, you can have lots of options in any kind of software required.
- Can run on lot lower resources - Linux is very tame when it comes to resource usage. The system can run with very little RAM and CPU available. This is evident from the fact that most embedded systems are Linux based.
- Great degree of customisation - Since the source code to the Linux kernel is open, you can download it and customise it at the very kernel level. This makes it great for experimentation, and rapid prototyping.
- Simple to use - You can do almost everything in Linux from the command line. This is a big plus point for developers and admins.
- Full multitasking with protected memory. Multiple users can run multiple programs each at the same time without interfering with each other or crashing the system.
- Very efficient virtual memory, so many programs can run with a modest amount of physical memory.
- Access controls and security. All users must be authenticated by a valid account and password to use the system at all. All files are owned by particular accounts. The owner can decide whether others have read or write access to his files.
- A rich set of small commands and utilities that do specific tasks well -- not cluttered up with lots of special options. Unix is a well-stocked toolbox, not a giant do-it-all Swiss Army Knife.
- Ability to string commands and utilities together in unlimited ways to accomplish more complicated tasks -- not limited to preconfigured combinations or menus, as in personal computer systems.
- A powerfully unified file system. Everything is a file: data, programs, and all physical devices. Entire file system appears as a single large tree of nested directories, regardless of how many different physical devices (disks) are included.
- A lean kernel that does the basics for you but doesn't get in the way when you try to do the unusual.
- Available on a wide variety of machines - the most truly portable operating system.
- Optimized for program development, and thus for the unusual circumstances that are the rule in research.

- already exists for over 30 years
- available for almost any hardware platform
- made to keep on running
- secure and versatile
- scalable

UNIX Cons

Using UNIX successfully is going to mean having a UNIX expert on site. Even something as simple as installing new products and updates can be complicated if you are not a UNIX expert. The command nature of the UNIX system is difficult, especially for beginners, so if you have a shop where newbies need to get up to speed quickly, UNIX might not be a good choice. This is why UNIX is most popular in places where there are sophisticated users.

Because UNIX is customizable, different dialects of UNIX have sprung up. For example, when the Berkley UNIX group didn't like the way UNIX classifies users, the Berkley users changed the OS code. Standard UNIX users use "cgup" as the command to change groups; the Berkley users use "NewGroup." This means that an expert UNIX user at one location might be facing a steep learning curve at a new place.

UNIX is case sensitive.

UNIX is not user friendly.

Disadvantages:

- many different dialects
- on proprietary systems: bundling and system specific implementation of commands/packages
- not user friendly, confusing for beginners
- proprietary hardware is expensive
- The traditional command line shell interface is user hostile - designed for the programmer, not the casual user.
- Commands often have cryptic names and give very little response to tell the user what they are doing. Much use of special keyboard characters - little typos have unexpected results.

- To use UNIX well, you need to understand some of the main design features. Its power comes from knowing how to make commands and programs interact with each other, not just from treating each as a fixed black box.
- Richness of utilities (over 400 standard ones) often overwhelms novices. Documentation is short on examples and tutorials to help you figure out how to *use* the many tools provided to accomplish various kinds of tasks.