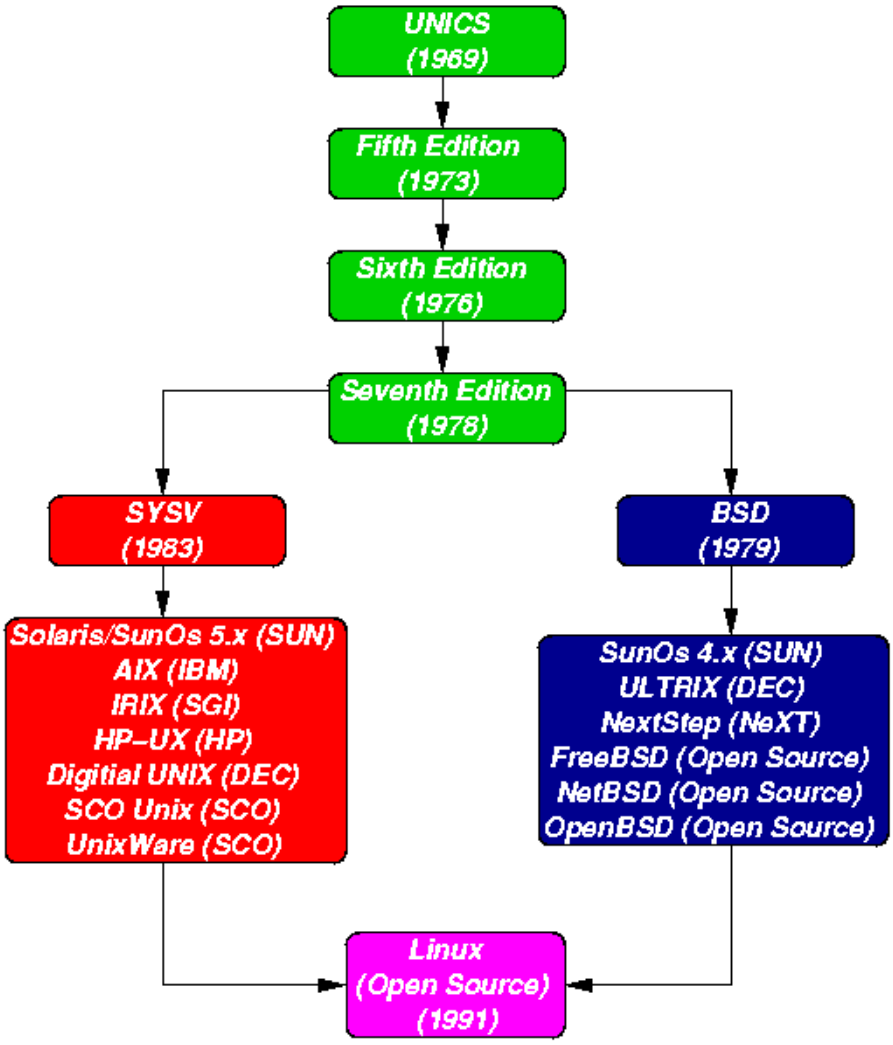
1. A Brief History of UNIX

UNIX has been a popular OS for more than two decades because of its multi-user, multi-tasking environment, stability, portability and powerful networking capabilities. What follows here is a simplified history of how UNIX has developed.



2. Architecture of the Linux Operating System

Linux has all of the components of a typical OS:

Kernel

The Linux kernel includes device driver support for a large number of PC hardware devices (graphics cards, network cards, hard disks etc.), advanced processor and memory management features, and support for many different types of filesystems (including DOS floppies and the ISO9660 standard for CDROMs). In terms of the services that it provides to application programs and system utilities, the kernel implements most BSD and SYSV system calls, as well as the system calls described in the POSIX specification.

The kernel (in raw binary form that is loaded directly into memory at system startup time) is typically found in the file /boot/vmlinuz, while the source files can usually be found in /usr/src/linux.The latest version of the Linux kernel sources can be downloaded from http://www.kernel.org.

Shells and GUIs

Linux supports two forms of command input: through textual command line shells similar to those found on most UNIX systems (e.g., sh, - the Bourne shell, bash - the Bourne again shell and csh - the C shell) and through graphical interfaces (GUIs) such as the KDE, Unity or GNOME window managers. If you are connecting remotely to a server, your access will typically be through a command line shell.

System Utilities

Virtually every system utility that you would expect to find on standard implementations of UNIX (including every system utility described in the POSIX specification) has been ported to Linux. This includes commands such as ls, cp, grep, awk, sed, bc, wc, more, and so on. These system utilities are designed to be powerful tools that do a single task extremely well (e.g., grep finds text inside files while wc counts the number of words, lines and bytes inside a file). Users can often solve problems by interconnecting these tools instead of writing a large monolithic application program.

Like other UNIX flavours, Linux's system utilities also include server programs called daemons which provide remote network and administration services (e.g. telnetd and sshd provide remote login facilities, lpd provides printing services, httpd serves web pages, crond runs regular system administration tasks automatically). A daemon (probably derived from the Latin word which refers to a beneficient spirit who watches over someone, or perhaps short for "Disk And Execution MONitor") is usually spawned automatically at system startup and spends most of its time lying dormant (lurking?) waiting for some event to occur.

Application programs

Linux distributions typically come with several useful application programs as standard. Examples include the emacs editor, gcc (a C compiler), g++ (a C++ compiler), latex (a powerful typesetting language) or LibreOffice.

Redhat Linux also comes with rpm, the Redhat Package Manager that makes it easy to install and uninstall application programs, meanwhile Debian or Ubuntu use deb extension.

3. Logging into (and out of) UNIX Systems

Text-based (TTY) terminals:

When you connect to a UNIX computer remotely (using telnet) or when you log in locally using a text-only terminal, you will see the prompt:
At this prompt, type in your usename and press the enter/return key. Remember that UNIX is case sensitive (i.e. Will, WILL and will are all different logins). You should then be prompted for your password:

```
login: will
password:
```

Type your password in at the prompt and press the enter/return key. Note that your password will not be displayed on the screen as you type it in.

If you mistype your username or password, you will get an appropriate message from the computer and you will be presented with the login: prompt again. Otherwise, you should be presented with a shell prompt, which looks something like this:

```
$
```

To log out of a text-based UNIX shell, type "exit" at the shell prompt (or if that does not work try "logout"; if that does not work press ctrl-d).

Graphical terminals:

If you are logging into a UNIX computer locally, or if you are using a remote login facility that supports graphics, you might instead be presented with a graphical prompt with login and password fields. Enter your user name and password in the same way as above (N.B. you may need to press the TAB key to move between fields).

Once you are logged in, you should be presented with a graphical window manager that looks similar to the Microsoft Windows interface. To bring up a window containing a shell prompt look for menus or icons which mention the words "shell", "xterm", "console" or "terminal emulator".

To log out of a graphical window manager, look for menu options similar to "Log out" or "Exit".

4. Changing your password

One of the things you should do when you log in for the first time is to change your password.

The UNIX command to change your password is passwd:

```
$ passwd
```

The system will prompt you for your old password, then for your new password. To eliminate any possible typing errors you have made in your new password, it will ask you to reconfirm your new password.

Remember the following points when choosing your password:

- Avoid characters which might not appear on all keyboards, e.g. '£'.
- The weakest link in most computer security is user passwords so keep your password a secret, do not write it down and do not tell it to anyone else. Also, avoid dictionary words or words related to your personal details (e.g. your boyfriend or girlfriend's name or your login).
- Make it at least 7 or 8 characters long and try to use a mix of letters, numbers and punctuation.

5. General format of UNIX commands

A UNIX command line consists of the name of a UNIX command (actually the "command" is the name of a built-in shell command, a system utility or an application program) followed by its "arguments" (options and the target filenames and/or expressions). The general syntax for a UNIX command is:

```
$ command -options targets
```

Here command can be though of as a verb, options as an adverb and targets as the direct objects of the verb. In the case that the user wishes to specify several options, these need not always be listed separately (the options can sometimes be listed altogether after a single dash).

6. The UNIX Filesystem

The UNIX operating system is built around the concept of a filesystem, which is used to store all of the information that constitutes the long-term state of the system. This state includes the operating system kernel itself, the executable files for the commands supported by the operating system, configuration information, temporary workfiles, user data, and various special files that are used to give controlled access to system hardware and operating system functions.

Every item stored in a UNIX filesystem belongs to one of four types:

Ordinary files can contain text, data, or program information. Files cannot contain other files or directories. Unlike other operating systems, UNIX filenames are not broken into a name part and an extension part (although extensions are still frequently used as a means to classify files). Instead they can contain any keyboard character except for '/' and be up to 256 characters long (note however that characters such as *, ?, # and & have special meaning in most shells and should not therefore be used in filenames). Putting spaces in filenames also makes them difficult to manipulate - rather use the underscore '_'.

Directories are containers or folders that hold files, and other directories.

Devices to provide applications with easy access to hardware devices, UNIX allows them to be used in much the same way as ordinary files. There are two types of devices in UNIX - block-oriented devices which transfer data in blocks (e.g. hard disks) and character-oriented devices that transfer data on a byte-by-byte basis (e.g. modems and dumb terminals).

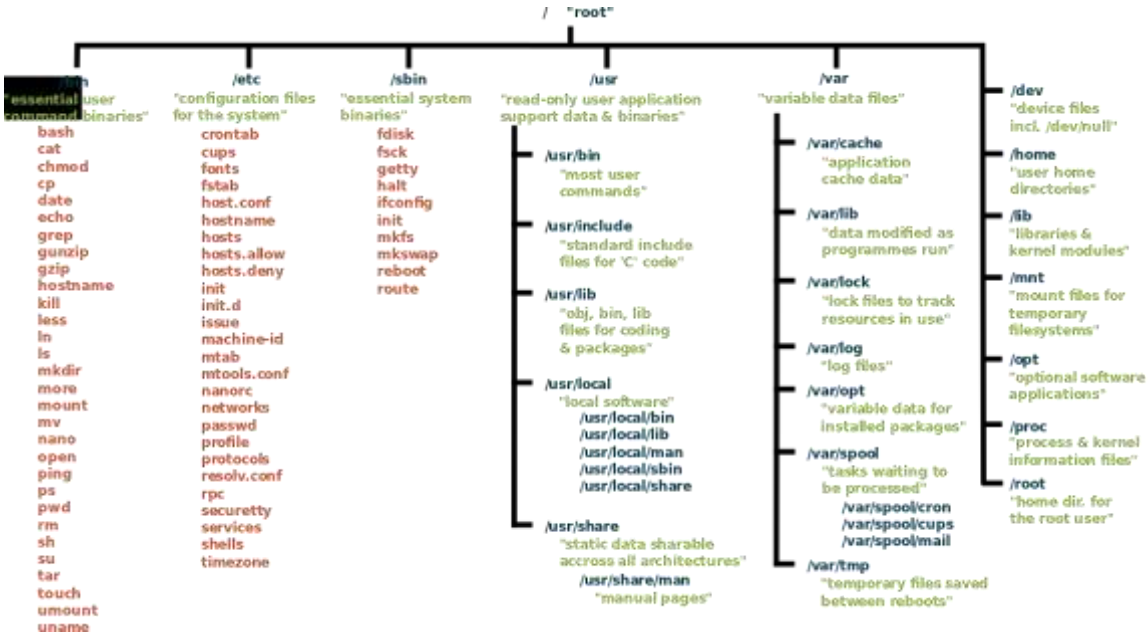
A **link** is a pointer to another file. There are two types of links - a hard link to a file is indistinguishable from the file itself. A soft link (or symbolic link) provides an indirect pointer or shortcut to a file. A soft link is implemented as a directory file entry containing a pathname.

7. Typical UNIX Directory Structure

The UNIX filesystem is laid out as a hierarchical tree structure, which is anchored at a special top-level directory known as the root (designated by a slash '/'). Because of the tree structure, a directory can have many child directories, but only one parent directory.

Certain conventions exist for locating some kinds of files, such as programs, system configuration files, and users' home directories.

Here is a generalized overview of common locations of files on a Unix operating system:



Directory or file	Description
/	The slash / character alone denotes the root of the filesystem tree.
/bin	Stands for binaries and contains certain fundamental utilities, such as ls or cp, that are needed to mount /usr, when that is a separate filesystem, or to run in one-user (administrative) mode when /usr cannot be mounted.
/boot	Contains all the files needed for successful booting process.
/dev	Stands for devices. Contains file representations of peripheral devices and pseudo-devices.

/etc	Contains system-wide configuration files and system databases; the name stands for etcetera.
/home	Contains user home directories on Linux and some other systems.
/lib	Originally essential libraries: C libraries, but not Fortran ones. On modern systems, it contains the shared libraries needed by programs in /bin, and possibly loadable kernel module or device drivers. Linux distributions may have variants /lib32 and /lib64 for multi-architecture support.
/media	Default mount point for removable devices, such as USB sticks, media players, etc.
/mnt	Stands for mount. Empty directory commonly used by system administrators as a temporary mount point.
/opt	Contains locally installed software.
/proc	procfs virtual filesystem showing information about processes as files.
/root	The home directory for the superuser root - that is, the system administrator. This account's home directory is usually on the initial filesystem, and hence not in /home (which may be a mount point for another filesystem) in case specific maintenance needs to be performed, during which other filesystems are not available. Such a case could occur, for example, if a hard disk drive suffers physical failures and cannot be properly mounted.
/sbin	Stands for "system (or superuser) binaries" and contains fundamental utilities, such as init, usually needed to start, maintain and recover the system.
/srv	Server data (data for services provided by system).
/sys	In some Linux distributions, contains a sysfs virtual filesystem, containing information related to hardware and the operating system.
/tmp	A place for temporary files not expected to survive a reboot. Many systems clear this directory upon startup or use tmpfs to implement it.
/usr	It holds executables, libraries, and shared resources that are not system critical, like the X Window System, KDE, Perl, etc. In older Unix systems, user home directories might still appear in /usr alongside directories containing programs, although by 1984 this depended on local customs.
/var	Stands for variable. A place for files that may change often - especially in size, for example e-mail sent to users on the system, or process-ID lock files.