1 What is UNIX?

UNIX is a computer operating system, that is, a collection of programs which allows users to manipulate information and manage resources on a computer. (UNIX is not an acronym for anything, but rather a little joke by the original developers. They were working on a system called "MULTICS", which stood for Multiplexed Information and Computing System, and were not very happy with its performance. So they created a two user system which they refered to as "UNICS", i.e. Uniplexed Information and Computer System. Somewhere along the road "UNICS" became "UNIX", and the rest, as they say, is history.) What makes UNIX different from other operating systems is that it was not designed specifically to run on just one computer. Most operating systems will run only on a particular brand of hardware, so if you made a commitment to an operating system, you were making a commitment to one type of hardware, from one particular vendor. Since the source code for UNIX is written in C, instead of some computer-specific machine language, UNIX can run on any computer with a C compiler, which nowadays means any computer. If you learn UNIX, then, you'll have a tool which you can use on just about any computer you encounter in the course of your work. And if you also choose to learn the C programming language, you can extend the capabilities of the UNIX operating system in any direction you choose.

The basic core routines which comprise UNIX are known as the kernel. In the early days of UNIX, users weren't considered to be serious unless they "hacked the kernel", that is, made their own personal changes to the heart of the operating system. (Of course, UNIX was the only operating system which would allow this kind of thing, since the others consisted of tightly guarded proprietary machine language.) However the vast majority of UNIX users now never have to deal with the kernel, and, in fact, might not even know of its existence. This is because of the existence of programs known as shells or command interpreters. These programs provide an easier way to access the power of the UNIX operating system than dealing with the kernel directly. Shells are discussed in more detail in a later section of this article, and will be the topic of future articles as well.

2 Files in UNIX

Along with portability, one of the major advances of the UNIX operating system is in the way it organizes and accesses files. Files are the basic units of information on a computer and can include text, graphics, programs, sounds, or just about anything else. Under UNIX, files are named in a hierarchial fashion, using a slash (`/`) to separate the different levels of organization. Each level in the hierarchy is called a directory. For example, many installations store locally developed information in a directory called `/usr/local`; within that directory might be sub-directories for source code (`src`), executable commands (`bin`), help files (`help`), and other information. (The term `bin` is often used for directories containing commands because they are usually binary files, as opposed to text files.) Therefore the full name of a local help directory would be `/usr/local/help`; a particular helpfile, say for text processing, might be found in `/usr/local/help/textprocess`.

Another innovation in the UNIX file system is that all external devices, like disk drives, tapes, CD/ROMs, etc. are all treated like files. For example, a magnetic tape drive might be called `/dev/rmt0`. In this way, dealing with a wide variety of peripherals is simplified because programs which access them do so in the same fashion as they would access a regular file.

3 Shells in UNIX

There are a wide variety of shells available within UNIX. Some of the names which you may hear when people talk about shells include the Bourne shell, the C-shell and the Korn shell. The core functions which the shells provide are pretty similar, but each offers some special features of its own. One feature common to all the shells eliminates the need to specify or even know the full name of a file, which includes all the pieces of the hierarchy described above. (This full specification is often refered to as a file's pathname.) When working in UNIX you are said to be "in" a particular directory, known as your current or working directory, and it is the shell that keeps track of this and lets you change your working directory. When you refer to a filename which doesn't begin with a slash (`/`), it is interpreted relative to your current directory. So even if your files are stored in the directory `/disk0/users/files/me`, you could refer to a file called `my.file` (whose complete pathname is `/disk0/users/files/me/my.file`) as simply `my.file`, if your current directory is set to `/disk0/users/files/me`. By default, when you log in, your working directory is set to what is known as your home directory, where your personal files are stored. You can create as many levels of additional directories as you wish in your home directory.

One of the basic functions of the shell is to execute the commands you type. This means that the shell needs to know about where commands are found on your system (that is, which directories commands may be found in). As with most of the other shell services, this is generally transparent to the user. You just type a command like `ls` (to list the names of files in a directory) or `date`, to tell you the time and date, and the shell figures out where to find the command.

4 Graphical User Interfaces

Recently, with the introduction of computer monitors capable of displaying a wide variety of text and graphics, graphical user interfaces (GUIs) have become increasingly popular. Instead of limiting your communication with the computer to just typing in commands in response to a prompt, these graphical interfaces allow you to perform many tasks by manipulating icons (a small object on the computer screen that represents a program or file on the computer), or making choices from menus. Two of the more common interfaces are Open Look and Motif. It's important to remember that these interfaces under UNIX serve the same function as the shell; they allow you to communicate more easily with the kernel, which is the heart of the operating system. In keeping with the UNIX tradition, the kernel itself doesn't need to be modified in order for a graphical interface to be used. Furthermore, one of the basic pieces of all the UNIX GUIs is a command shell, that is a window on the screen which allows you type UNIX commands into the shell of your choice, so you don't need to forego the command line to use a GUI.