Summary

System programs are fundamentally different from the kinds of programs that most beginning students learn how to write because they access protected resources inside the computer system. What actually happens when a program makes a relatively simple call to print onto the terminal window involves much more than what meets the eye. The sequence of steps includes the use of system calls, which are function calls into the kernel code. The kernel is the core of the operating system, the part that is memory resident as long as the computer is powered on, and is responsible for protecting, managing, and making available the wide range of resources in the computer system.

Unix introduced many novel ideas in the design of operating systems. Some of the most innovative ideas that made it so successful are the following:

- A programmable, interchangeable command line interpreter, called a shell, that runs in userspace rather than as a part of the kernel
- The concept of processes and the method of process creation
- The use of two levels of privilege to provide protection of the kernel and its resources
- Device-independent I/O operations
- The representation of files as sequences of bytes without structure
- I/O redirection and pipes in particular
- The concepts of users and groups and file permissions
- The single directory hierarchy
- The environment concept

The growth and spread of Unix led to many different Unix varieties and distributions and a need for standardization. This in turn led to the creation of a consortium that created the POSIX standards for its interfaces and behavior.

Exercises

1. Who are the authors of the `bash` shell? (Hint: Use the man pages to find out.)
2. What is the return type of the `read()` system call?
3. Using the man pages, find the names of all of the header files that you would need to include to use the following functions in a program. There might be more than one needed for some of these.
   (a) `_exit()`
   (b) `setuid()`
   (c) `fstat()`

4. If your current working directory is */usr/share/gcc/python*, what is the shortest relative pathname of the file */usr/lib32/libc.so.6*?
5. What command can be used to print the creation date of a file? (Hint: This information is part of a file's status.)

1、 Brian Fox, Chet Ramey

2、 ssize_t

3、 (a) #include <unistd.h>
(b) #include <unistd.h>
(c) #include <sys/stat.h>

4、 ../../../lib32/lib32.so.6

5、 stat、