Assignment 1

CPSC 1280

Dr. Bagheri

May 22, 2014

Ryan Woodward

100201137

1. An operating system provides services for the hardware by accessing it in order to be able to run programs. It creates a process, allocates memory for the program, loads the cpu registers with control information, takes system calls from programs if they need to use the hardware, and after a program has finished executing it cleans up the memory and registers making them available for the next program. For the user the services provided are creating files and directories, and acting as an interface between the user and the hardware basically.
2. If a process (not program) needs to read a file it sends a system call to the operating system. The operating system will create a process to read the file and send the information to the other process via message passing.
3. Yes a process (program) will complete it’s time quantum before it makes way for another process (program) to use the CPU’s resources. A time slice is the amount of time that a task is allowed to use the CPU’s resources for. By scheduling interrupts, if a process is still running but gathering information from an i/o device for example, the kernel will remove that process from the registers and load a new process in. Therefore maximizing the CPU’s resources.
4. Multiprogramming means that multiple programs are allowed to reside in the computers memory at once. They are loaded in blocks as to not take up the entire memory for themselves. Multi-user means that multiple people can all be working on the same UNIX system. Multitasking is when multiple processes are run concurrently. Some can be running in the foreground, or in the background.
5. A UNIX command is non-interactive and doesn’t have headers since the output for that command could become the input for another command. Therefore without needing a information from a user, or cluttering up the output with header information, multiple commands and be strung together using piping.
6. System calls are when a process asks for help from the operating system. This is when a process needs to make use of the computer’s hardware, and needs the operating system in order do that. C is programming is so much different in the UNIX environment since the programmer will have access to the entire library of system calls. On windows the programmer will have to use library functions.
7. This is false. A system call is the processes method of asking for help from the operating system. In two different UNIX systems the system calls from the processes will be the same, perhaps the difference will only occur on the operating system end if the two different UNIX systems have different hardware.
8. The three commands to log yourself out of the system are ctrl-d, typing logout, or typing exit into the command line. Depending on which version of the shell you are using sometimes ctrl-d or typing logout will not work. However typing exit works in most cases.
9. In both cases of echo > README, and echo > readme, two different files were created as evidenced when ls was run. This makes me conclude that the UNIX environment is case sensitive.
10. Command Observations
    1. who
       1. displays all the current users logged in, the date and time they logged in, and which port they are logged in from
    2. tty
       1. this returned to the display console /dev/pts/1. This meanas that from the physical devices folder, and the ports folder, the port that the computer is connected to the UNIX system is 1.
    3. tput clear
       1. this command and it’s option [clear] clears the terminal screen.
    4. id
       1. this will print out all the real and the effective user id’s to the display console
    5. ps
       1. this command lists all the current processes on the display console
    6. echo $$
       1. this command lists the process id of the current shell.