CS-701 Lecture 2

February 1, 2001

Dr. Vickery

Unix Overview

- The kernel
  - Processes
  - System calls
  - Library functions
- The file system
  - Files and Directories
  - Non-file files(!)
- The shells
  - *sh, ksh, bash, zsh*
  - *csh, tcsh, zsh*
  - Read, edit, interpret command lines

Applications

- An example of an application is a program that you write.
  - Takes the form of an executable file.
  - Every application runs as a process.
- Each process has a unique number (pid) that identifies it in the system.
  - When the system initializes, the first process is created (pid = 1).
  - After that, all processes are created by some other process calling the *fork()* kernel function.

Kernel Calls

- The kernel provides a set of functions, like *fork()*, that can be called from applications.
- When kernel functions execute, the CPU is put into a special mode so that it can execute privileged machine instructions.
  - It takes a lot of time to switch in and out of kernel mode.
  - Library functions contain machine language code to do the actual switching into kernel mode.

Library Calls

- To save programmers the work of writing the same utility functions over an over, there is are library files that contain pre-compiled functions that applications can call.
  - Every C program is automatically linked to the Standard C Library, but others can be added.
  - Some library functions also make kernel calls, but they minimize the number of kernel calls as much as possible.
- Example: The only kernel call that does output is *write()*. The library function *fwrite()* calls *write()*, but not necessarily every time it is called.

The File System

- "Everything is a file" in Unix, even if it isn't really.
- There is just one root node (/)
  - Multiple volumes may appear within the tree
    - E.g., /mnt/floppy might be the path to the root of your floppy disk, unlike the "A: drive" concept of DOS/Windows.
  - Device drivers (code in the kernel), processes, and sockets are all examples of non-files that look like files in a Unix file system.

Similar DOS and Unix Commands

Unix	**DOS**	
cp	copy	Copy files
mv	move	Move files
rm	del	Delete files
rmdir	rd	Delete directory
cd	cd	Change currrent directory
ls	dir	List directory contents

Shell Algorithm

- Display a prompt string
- Read a command line
- Substitute environment variables
- Expand aliases and patterns
- Redirect I/O
- Interpret or Execute command(s)
  - Interpret built-in commands like *cd*, *setenv*, *exit*, etc.
  - Create processes to execute files.
- Repeat from the beginning.

Assignment 1

- Use a Unix account to edit a properly documented showenv.c file, build an executable file named *showenv* using the *make* command, and send me a tar file containing your showenv.c file and a typescript of a session in which you build and execute *showenv*.
- Due date depends on when accounts are set up and when I have gone over all the steps involved.