Processes

CSC 510

Unix Processes

- A process represents a running program
- The OS needs to keep track of internal data structures for each process, such as:
  - address space
  - status (sleeping, stopped, etc.)
  - execution priority
  - resource utilization
  - owner of the process
- To create new process the `fork` system call is used which copies an existing process for the new process; the process that calls `fork` is called the parent process.

Identifiers For Processes

- PID: unique process ID number
- PPID: the process ID of the parent process
- UID: the user ID of the user that created the process
- EUID: the effective user ID of the process; determines permissions
- GID: the group ID of the process
- EGID: the effective group ID of the process

Signals

- Signals are a asynchronous process requests used for various purposes, such as:
  - a means of inter-process communication
  - sent from a terminal to terminate, interrupt, or suspend a process
  - sent by the OS when an infraction occurs, e.g. divide by zero
- The `kill` command can be used to send a signal to a process

Process Monitoring

- The `ps` command can be used to monitor processes
- The `ps aux` command will show information about all the processes running including:
  - the user who created the process
  - the PID of the process
  - cpu and memory utilization
- The `top` command is an interactive process monitor that updates information in real time.

The /proc Filesystem

- /proc is a pseudo-filesystem that contains various information about the system state.
- Process specific information is in /proc/<PID> and contains information such as:
    - cmd – the command the process is executing
    - cmdline – complete command line of the program
    - environ – environment variables
    - stat – general process state information
    - statm – memory usage