Intro to Linux

1.4.2 - Process Management

CYBER.ORG
THE ACADEMIC INITIATIVE OF THE CYBER INNOVATION CENTER

Process Management

- Efficiently handle and control the execution of programs on a computer system
- Send signals to processes for termination, monitoring and listing processes, adjusting process priorities, recognizing different process states, and employing job control mechanisms for managing foreground and background tasks

Kill Signals

- Manage processes by signaling them to perform specific actions
- SIGTERM terminates processes, performing cleanup activities before exiting
- SIGKILL immediately terminates a process without allowing cleanup
- SIGHUP is associated with the hangup event on a terminal

Listing Processes and Open Files

- **`top`** is a dynamic, real-time process viewer providing continuously updated display of system processes
  - Provides overview of CPU usage, memory usage, running processes. etc
- **`ps`** (Process Status) is a command-line utility that displays information about processes running on the system
  - Provides static snapshot of current processes
- **`lsof`** (List Open Files) lists open files and processes that have them open
  - Provides information about files, sockets, and devices processes are currently using
- **`htop`** is an interactive process viewer
  - Provides a visually appealing representation of system processes

Setting Priorities

- **`nice`**
  - Launches a new process with a specified priority level
  - Adjusts CPU time compared to other processes
- **`renice`**
  - Changes the priority of a currently running process
  - Useful if users not to adjust processes based on a system's workload

Process States

- Zombie processes have completed their execution but still have entries in the process table
  - Entries are retained until the parent process retrieves the exit status of the terminated child process
- Sleeping processes lie in dormancy, waiting for a specified event to occur
  - While sleeping, no CPU resources are consumed
- Running processes actively execute instructions on the CPU
  - The active phase of a process, performing computations, responding to user input, and executing other instructions
- Stopped processes have been suspended, usually by signal
  - Don't use CPU resources but can be resumed later by sending an appropriate signal

Job Control

- Management of processes in the foreground and background, allowing users to interact with and control the execution of tasks
- **`bg`**
  - Puts a suspended or stopped process in the background
- **`fg`**
  - Brings a background process to the foreground, making it an active task in the terminal
- **`jobs`**
  - Lists the jobs currently running or stopped in the background
- Ctrl+Z suspends the currently running foreground process
- Ctrl+C terminates a running process in the foreground
- Ctrl+D sends an end-of-file signal

pgrep, pkill, and pidof

- **`pgrep`** (process grep)
  - Searches for processes based on name or other attributes
  - Prints the process IDs of matching processes
- **`pkill`** (process kill)
  - Sends a signal to processes based on their name
  - Sends the SIGTERM signal
- **`pidof`**
  - Obtains the process ID of a running program based on its name
  - Returns the PID of the first process found with the specified name