** cpu – central processing unit  
   o Calculation module of hardware  
   o Interprets instructions and performs operations  
   o Controls processes  

** ls – list  
The ls command loops through names given by the kernel and displays them to the terminal.

<table>
<thead>
<tr>
<th>Process</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>A program or instance of a program that is running</td>
<td>An executable file that when run produces a certain result</td>
</tr>
</tbody>
</table>

There can be multiple instances of the ls program running as processes.

A process gets loaded into the CPU.

When you type a command you are communicating with the bash shell.

Bash looks for the program to see where its located. When found it issues a system call and tells the kernel by fork(). Parent process id doesn't change. The child process gets a new id. The child process sets up a system call exec(). On the disk there is a program that is /bin/ls and begins executing. While the ls program runs, bash gives up control of the terminal to ls. When finished ls sends the exit() system call. Parent process waits for its child to die with wait().

**System Library Function Calls:**
exec()      fork()      exit()      wait()
**Current Working Directory (CWD)** – the location where files are looked for.

`emacs` `fstab`
- When this command is run in the `/etc` directory, then this command will open the `fstab` file
- If you run this in a different directory that does not have `fstab` in it then `emacs` will open with a blank screen to edit a new `fstab` file

The current working directory gets passed to the child processes from the parent.

`cd` – change directory

Each process has a process id.

**Process Id - PID**
**Parent Process Id - PPID**
**User Id – UID**
**Group ID – GID**

**Terminal** – keystrokes, display

**SHHD** – secure shell daemon

**Getty/login** – one is parent of shell

**Daemon** – server process that runs in the background.

Chmod 775 ./foo
`ls -l ./foo`

```
-rwxr-xr-x 7    read/write/execute applies to the owner of file
r-x 5    Group can only read and execute
r-x 5    World can read/execute
```

The owner permissions supersede the group permissions in the above example.

Asynchronous communication mechanism – signals are sent to processes
  - Asynchronous – one way, not in order

One signal acts as a software interrupt; It tells the process to stop and pay attention

```
-rws    setuid will be permissions of user
setuid root    - common security problem
```
Signals:
- Interrupt – tells processes to stop
- Hang-up – like old modems, it is a signal that alerts that there is a disconnect
- Kill – destroys process without waiting for a response

Commands:
```
ps – lists running processes on a system
kill – sends a signal to kill a process
   kill –HUP 72
top – displays processes plus more info and automatically will refresh
```

HOMEWORK: Find and read GNU Manifesto to discuss on Friday.