**Fork**

fork() creates a new process by duplicating the calling process.

On success, the PID of the child process is returned in the parent, and 0 is returned in the child. On failure, -1 is returned in the parent, no child process is created, and errno is set appropriately.

Errors: **EAGAIN** fork() cannot allocate sufficient memory to copy the parent's page tables and allocate a task structure for the child.

**Stat**

These functions return information about a file. No permissions are required on the file itself

* **stat()** stats the file pointed to by path and fills in buf.
* **lstat()** is identical to stat(), except that if path is a symbolic link, then the link itself is stat-ed, not the file that it refers to.
* **fstat()** is identical to stat(), except that the file to be stat-ed is specified by the file descriptor fd.

**Kill**

The kill() system call can be used to send any signal to any process group or process.

Errors: **EPERM** The process does not have permission to send the signal to any of the target processes.

**Trap**

To switch between user mode and kernel mode. It acts like a function call but it jumps onto a specific address in the kernel space and switches into the kernel mode