January 28, 2016-1 — Notes

CSE 450 Operating Systems

Professor Guangzhi Qu

Monday & Wednesday 3:30 — 5:17 pm

Nicholas Land

Process State

- As a process executes, it changes *state*
  - **new** The process is being created
  - **ready** The process is waiting to run
  - **running** Instructions are being executed
  - **waiting** Process waiting for some event to occur
  - **terminatied** The process has finished execution
- **Job queue** — set of all processes in the system
- **Ready queue** — set of all processes residing in main memory, ready and waiting to be exectued
- **Device queues** — set of processes waiting for an I/O device
- **Scheduling** — Processes (PCBs) migrate among the various queues
- **Long-term scheduler** (or job scheduler) — selects processes should be brought into the ready queue
  - Long-term scheduler is invoced very infrequently
  - The long-term scheduler controls the *degree of multiprogramming*
- **Short-term scheduler** (or CPU scheduler) — selects which processes should be next and allocates the CPU
  - Short-term scheduler is invoced very frequently (milliseconds) $\Rightarrow$ (must be fast)
- Praent processes create children processes, which, in turn create other processes, forming a tree of processes
- Must construct new PCB
- Resource sharing strategies
  - Parant and children share all resources (I/O states, address space information)
  - Children share a subset of parent's resources
  - Parent and child share no resources (UNIX *exec*())
- Execution
  - Parant and child execute concurrently
  - Parent waits until the children terminate