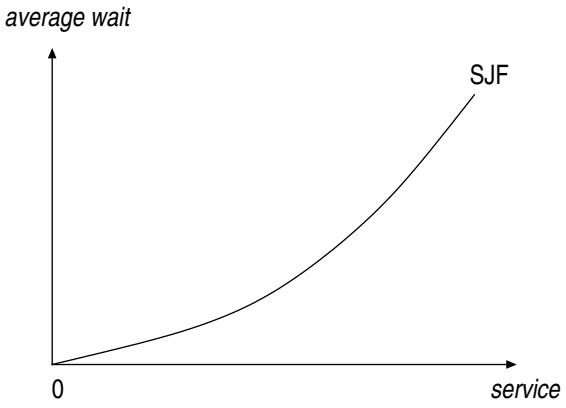
4 problems. 40 points. 30 minutes *Closed book. Closed notes. No electronic device.* *Write your name above.*

1. [6 points] An OS has 1 cpu, 2 io devices (io1, io2), pre-emptive cpu scheduling, and no multi-threaded processes. A process is terminated only by itself. The possible states of a process are given below. Draw the possible transitions (and omit the impossible ones).

new ready running io1 wait io2 wait terminated

2. [6 points] A collection of cpu-bound processes are scheduled on a cpu. The curve in the graph below shows the average wait vs service for SJF (shortest-job first, non-preemptive) scheduling.
(Recall: the *service* of a process is the total cpu time it requires; the *wait* of a process is the total time it spends in the ready queue; the *average wait* for service s is the average wait of all processes with service s.)

Draw on the same graph the expected curve for FIFO (instead of SJF). Repeat for SJF-preemptive. Repeat for RR (round robin). (So your answer is three curves on the same graph.)



3. [12 points] A multi-cpu shared-memory machine has a swap instruction (and no other "read-modify-write" instructions). Specifically, `swap(x,y)` atomically exchanges the contents of register `x` and memory location `y`.

Implement a (weak or strong) spin lock using the swap instruction. Specifically, give code chunks (at a level of detail as in the os-process slides) for

- lock definition
- lock `acq()`
- lock `rel()`

4. [16 points] You are given a multi-cpu machine with spin locks. Give an *efficient* implementation for a lock whose acquired durations can be long (e.g., seconds or minutes). Specifically, give code chunks (at a level of detail as in the os-process slides) for

- lock definition
- lock `acq()`
- lock `rel()`