

## OS homework 3<sup>rd</sup>

### CPU 调度

1. Why is it important for the scheduler to distinguish I/O-bound programs from CPU-bound programs?
  
2. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P <sub>1</sub>	10	3
P <sub>2</sub>	1	1
P <sub>3</sub>	2	3
P <sub>4</sub>	1	4
P <sub>5</sub>	5	2

The processes are assumed to have arrived in the order P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>, all at time 0.

- a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).
- b. What is the turnaround time of each process for each of the scheduling algorithms in part a?
- c. What is the waiting time of each process for each of the scheduling algorithms in part a?
- d. Which of the algorithms in part a results in the minimum average waiting time (over all processes)?
  
3. Which of the following scheduling algorithms could result in starvation?
  - a. First-come, first-served
  - b. Shortest job first
  - c. Round robin
  - d. Priority

4. The traditional UNIX scheduler enforces an inverse relationship between priority numbers and priorities: The higher the number, the lower the priority. The scheduler recalculates process priorities once per second using the following function:

$$\text{Priority} = (\text{Recent CPU usage} / 2) + \text{Base}$$

where base = 60 and recent CPU usage refers to a value indicating how often a process has used the CPU since priorities were last recalculated. Assume that recent CPU usage for process P1 is 40, process P2 is 18, and process P3 is 10. What will be the new priorities for these three processes when priorities are recalculated? Based on this information, does the traditional UNIX scheduler raise or lower the relative priority of a CPU-bound process?