# Homework1


## gzl0034 Gaoxiang Li


**1. [30 points] Discuss how the following pairs of scheduling criteria conflict in certain settings.**

**1.1 [15 points] CPU utilization and response time**

**1.2 [15 points] Average turnaround time and maximum waiting time**

CPU utilization can be increased by reduce the context switch. But if we reduce the number of context switch, response time will increase because the process may wait for a long time to be executed.

If we want to reduce the average turnaround time we need to use shortest first policy. But this policy will obviously increase the maximum waiting time especially for the longest process.

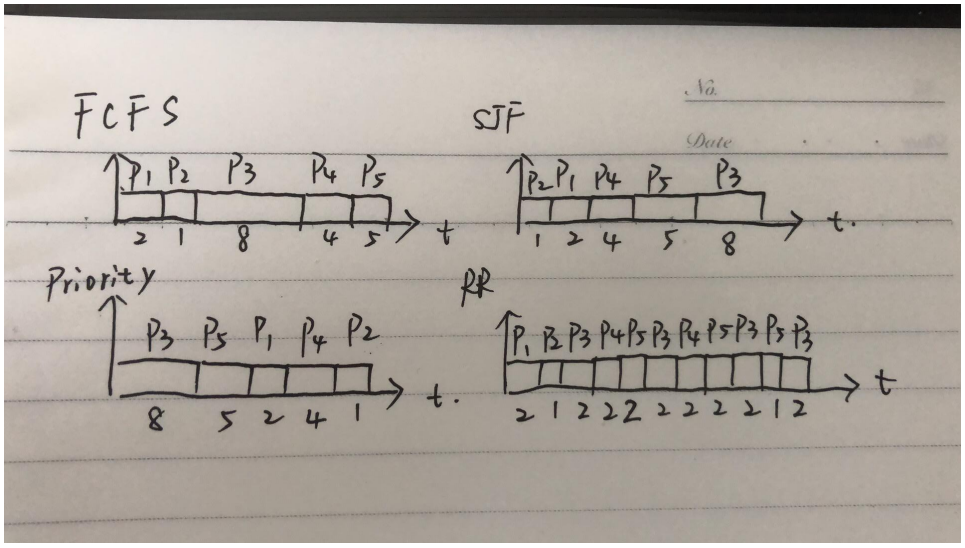
**2. [40 points] Consider the following set of processes, with the length of the CPU burst time given in milliseconds. The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.**

**2.1 [10 points] Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).**

**2.2 [10 points] What is the turnaround time of each process for each of the scheduling algorithms in Question 2.1?**

**2.3 [10 points] What is the waiting time of each process for each of these scheduling algorithms?**

**2.4 [10 points] Which of the algorithms results in the minimum average waiting time (over all processes)?**




**2.2**Turnaround Time:

FCFS: P1:2 P2:3 P3:11 P4:15 P5:20

SJF: P2:1 P1:3 P4:7 P5:12 P3:20

Priority: P3:8 p5:13 p1:15 p4:19 p2:20

RR: P1:2 P2:3 P3:20 P4:13 P5:18

**2.3**Waiting Time:
FCFS: P1:0 P2:2 P3:3 P4:11 P5:15
SJF: P2:0 P1:1 P4:3 P5:7 P3:12
Priority: P3:0 P5:8 P1:13 P4:15 P2:19
RR: P1:0 P2:2 P3:3 P4:5 P5:7

**2.4**RR result in minimum average waiting time.

**3. [15 points] Which of the following scheduling algorithms could result in starvation? Why? (1) First-come, first-served (2) Shortest job first (3) Round robin (4) Priority**

SJF: Some long process may starve if some shorter processes are added continually.
Priority:Some low priority process may starve if some high priority processes are added continually.

**4. [15 points] Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate a; when it is running, its priority changes at a rate b. All processes are given a priority of 0 when they enter the ready queue. The parameters a and b can be set to give many different scheduling algorithms. What is the algorithm that results from b > a > 0? Please justify your answer.**

FCFS
(1) For b>a>0, the priority of running process will always be bigger than the other process in the ready queue. Because the changing rate “a” of running process > the changing rate “b” of waiting process. So if one process is running, there is no process can preempt. The running process will running till it is over.
(2) For the changing rate b and all processes are give a priority of 0 when they enter the ready queue. If one process comes earlier than another, the priority of the early process will always be bigger than the later one.
In all, the scheduling algorithm is FCFS.