Homework 5

Name: Hanzhi Liu (刘涵之) No: 519021910102

Practice Exercise: 5.4, 5.5, 5.10, 5.18, 5.20

5.4

Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, 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 larger priority number implies a higher priority), and RR (quantum = 2).

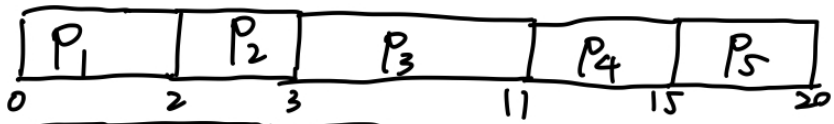
b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

c. What is the waiting time of each process for each of these scheduling algorithms?

d. Which of the algorithms results in the minimum average waiting time (over all processes)?

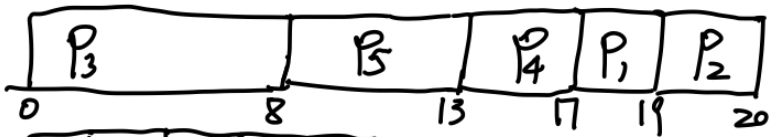
a.

FCFS

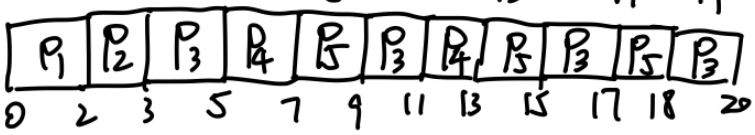


SJF

非抢占式优先



RR



b. Turnaround

Process	FCFS	SJF	非抢占式优先	RR
P_1	2	3	19	2
P_2	3	1	20	3
P_3	11	20	8	20
P_4	15	7	17	13
P_5	20	12	13	18

c. Waiting

Process	FCFS	SJF	非抢占式优先	RR
P_1	0	1	17	0
P_2	2	0	19	2
P_3	3	12	0	12
P_4	11	3	13	9
P_5	15	7	8	13
Avg	6.2	4.6	11.4	7.2

d. SJF有最短平均等待时间

5.5

The following processes are being scheduled using a preemptive, round-robin scheduling algorithm.

Process	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an **idle task** (which consumes no CPU resources and is identified as P_{idle}). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a

time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

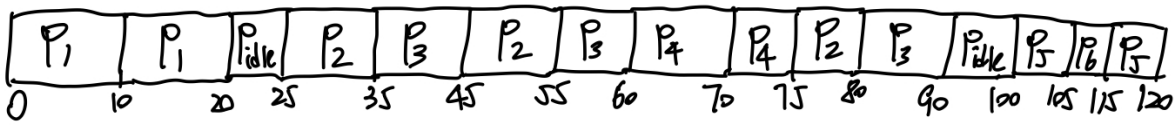
a. Show the scheduling order of the processes using a Gantt chart.

b. What is the turnaround time for each process?

c. What is the waiting time for each process?

d. What is the CPU utilization rate?

a. 搶式, RR



b.

	P_1	P_2	P_3	P_4	P_5	P_6
Turnaround	20	55	60	15	20	10
c. waiting	0	30	40	0	10	0

d. utilization rate $= \frac{120-(5+10)}{120} = 87.5\%$

5.10

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 P_1 is 40, for process P_2 is 18, and for process P_3 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?

- new priorities:
  - $P_1 = \frac{40}{2} + 60 = 80$
  - $P_2 = \frac{18}{2} + 60 = 69$
  - $P_3 = \frac{10}{2} + 60 = 65$
- cpu will lower the relative priority of CPU-bound processes

5.18

The following processes are being scheduled using a preemptive, priority-based, round-robin scheduling algorithm.

Process	Priority	Burst	Arrival
P_1	8	15	0
P_2	3	20	0
P_3	4	20	20
P_4	4	20	25
P_5	5	5	45
P_6	5	15	55

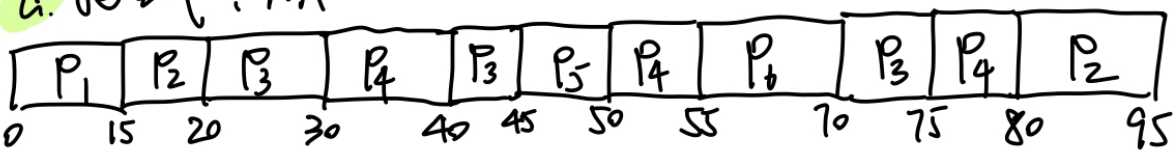
Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. The scheduler will execute the highest-priority process. For processes with the same priority, a round-robin scheduler will be used with a time quantum of 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

a. Show the scheduling order of the processes using a Gantt chart.

b. What is the turnaround time for each process?

c. What is the waiting time for each process?

a. 抢式，RR



b.

	P_1	P_2	P_3	P_4	P_5	P_6
turnaround	15	95	55	55	5	15

c.

	P_1	P_2	P_3	P_4	P_5	P_6
waiting	0	75	35	35	0	0

5.20 Which of the following scheduling algorithms could result in starvation?

a. First-come, first-served

b. Shortest job first

c. Round robin

d. Priority

SJF和优先级调度会导致进程饥饿（不断有剩余时间短的进程进入；不断有优先级高的进程进入）