**6.4** What advantage is there in having different time-quantum sizes at

different levels of a multilevel queueing system?

The priority of each queue is different. The higher the priority, the smaller the time-quantum size is, and the lower priority, the bigger the time-quantum size is. The purpose of this is to promote small jobs whose CPU burst time is short to be finished by CPU quickly. In order to facilitate the completion of those large jobs whose CPU burst time is long, the time-quantum size of last queue (the lowest priority queue) is generally large, even the last queue can use FCFS to schedule jobs.  
  
for example:  
Suppose that there are three jobs at the same time to reach the CPU. JOB1’ s burst time is 2, and JOB2’s burst time is 100, JOB3’s burst time is 1. If queues all have same time-quantum size(5), then the completion of JOB1 needs 2 and the completion of JOB3 needs 1, however, they all have 5 to operate which will waste some time. If the time-quantum size of first queue is 1, JOB1 can be done without wasting any time and resources. If the time-quantum size of second queue is 2, JOB3 can be done without wasting any time and resources.

**6.17** The following processes are being scheduled using a preemptive, round-robin

scheduling algorithm. 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 *Pidle* ). This

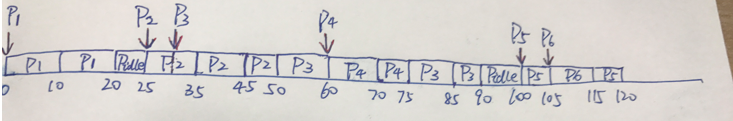
task has priority 0 and is scheduled whenever the system has no other

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?

P1:20-0=20

P2:50-25=25

P3:90-50=40

P4:75-60=15

P5:120-100=20

P6:115-105=10

c. What is the waiting time for each process?

P1:0

P2:0

P3:75-60+50-30=35

P4:0

P5:115-105=10

P6:0

d. What is the CPU utilization rate?  
105/120=87.5%

**6.31** Consider two processes, *P*1 and *P*2, where *p*1 = 50, *t*1 = 25, *p*2 = 75, and

*t*2 = 30.

a. Can these two processes be scheduled using rate-monotonic

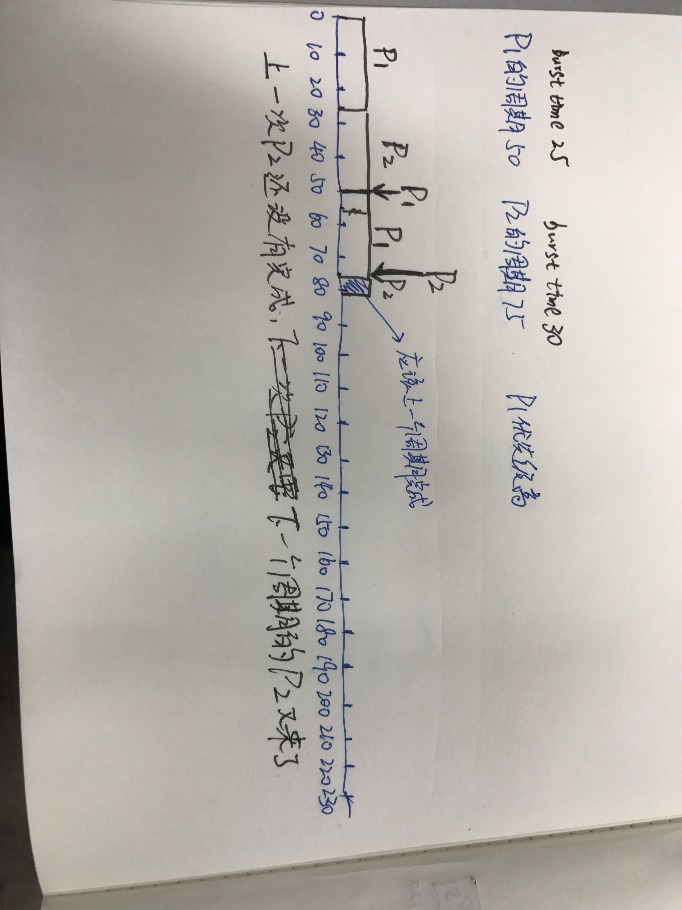
scheduling? Illustrate your answer using a Gantt chart such as

the ones in Figure 6.16–Figure 6.19.

b. Illustrate the scheduling of these two processes using earliest deadline-

first (EDF) scheduling.

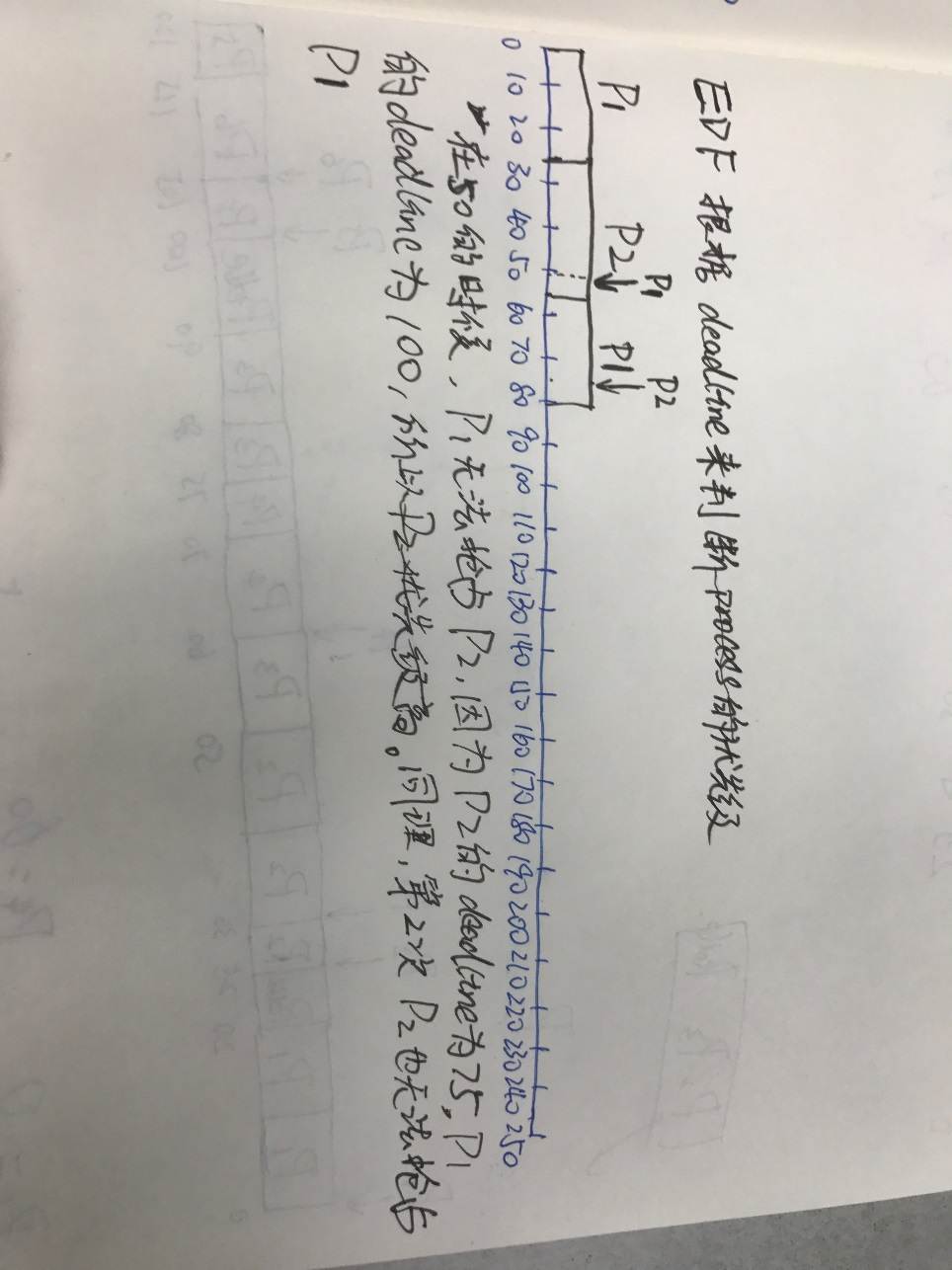
**a.**



These two processes can’t be scheduled using rate-monotonic

scheduling. Because P2 can’t be finished before next period.

**b**



These two processes can be scheduled using EDF scheduling. Because both P2 and P1 can be finished before next period.