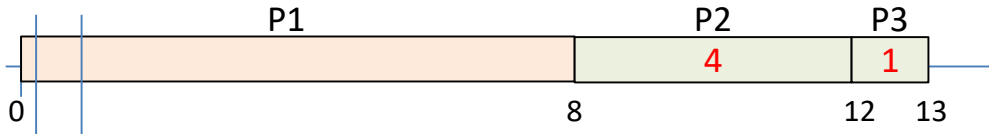
SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING
ET0023 OPERATING SYSTEMS

TUTORIAL 6 Process Scheduling – Suggested Solutions

Q1. Turnaroundtime = completion_time – arrival_time

(a)



0.4, P2 arrives

1.0, P3 arrives

P1 = end-time – arrival time
= 8 – 0
= 8
P2 = 12 – 0.4 = 11.6
P3 = 13 – 1 = 12
Average turnaround time = (8 + 11.6 + 12) / 3.0 = 31.6/3.0

(b) P1 runs
P3 runs
P4 runs
Average turnaroundtime = 9.53

(c) CPU is idle for 1 sec. assume arrival times are the same
Ready Queue now holds P1, P2, P3, use SJF
P3 runs
P2 runs
P1 runs
Average turnaroundtime = 6.86

Q2. (a) FCFS
Jobs start with P1 and move downwards to P5
Finishing times are 10, 16, 18, 22 and 30
Average turnaround time = 19.2 minutes

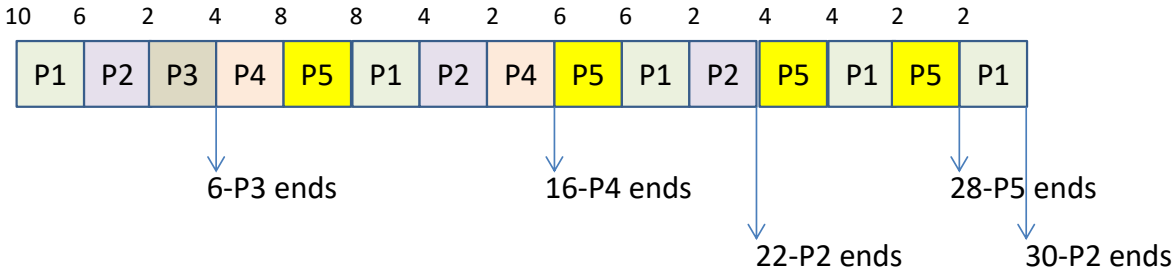
(b) SJF
Finishing times are 2, 6, 12, 20, 30
Average turnaround time = 14 minutes

(c) Priority Scheduling
Jobs run as P2, P5, P1, P3 and P4
Finishing times are 6, 14, 24, 26 and 30
Average turnaround time = 20 minutes

Q3. Round Robin scheduling

(a) If quantum is set to 10 minutes, then
Jobs run like a FCFS system
P1 runs to completion, P2 through P5
Finishing times are 10, 16, 18, 22 and 30
Average turnaround time = 19.2 minutes

(b) If quantum is set to 2 minutes, then



Average turnaround time = 20.4 minutes

Disadvantage: We must take into consideration of the context switching times. If the quantum is set too low, the context switch time must be taken into account as well.

Q4

a) FCFS and RR are quite simple to implement.
SJF – need to find out how long the job will run (estimation? Not exact)
SRT – can only be determine when the job runs, not possible to predict.

b) SRT and RR require timers to interrupt the CPU to context switch the next process

c) Pre-emptive algorithms require timers, hence answer is same as (b)

d) FCFS is very simple to implement (can even be run on batch systems), however, the yield is poor and with highly variable response and turnaround times.

RR has a better response time which can be improved with priority scheduling and the response/turnaround times are not dependent on the arrival order of processes as is the case with FCFS

Q5

Yes, the scheduler can have an empty queue.
Unless the scheduler can cope with the empty queue, the system may crash.
Some systems uses dummy process (which does nothing) to avoid the problem if the scheduler cannot cope.

In actual OSes, e.g. Windows/Linux there will always be some process running in the background.