Quiz 1 solutions

1) Round robin schedulers normally maintain a list of all runnable processes, with each process occurring exactly once in the list. What would happen if a process occurred twice in the list? Can you think of any reason for allowing this?

In this modified version of round robin scheduling, the process which is made to occur twice in the list would get more CPU time than other processes. We can use such scheduling whenever a process is to be given more priority than others.

2) Measurements of a certain system have shown that the average process runs for a time T before blocking on I/O. A process switch requires a time S, which is effectively wasted (overhead). For round robin scheduling with Quantum Q, give a formula for the CPU efficiency for each of the following.

a. Q=∞

b. Q>T

c. S<Q<T

d. Q=S

e. Q nearly 0

Assume there are n process CPU efficiency = $\frac{(effective\ cpu\ time)}{(total\ time)} * 100$

a) Q=∞

$$\frac{(\sum_{i=1}^{n} Ti)}{(\sum_{i=1}^{n} Ti + (n-1)*S)} = n*T/(n*T+(n-1)*S)$$

Each process takes T time on an average therefore n process will take n*T time.Since q=∞ each job would be processed in one round and there will be n-1 switchings.

b) Q>T

In this case also all the processes would be processed in one round and there would be n-1 switchings

$$n*T/(n*T+(n-1)*S)$$

c) S<Q<T

Here there will be Summation(ceil(Ti/Q)) switchings where i belongs to 1, 2,... n-1 if n were the number of processes.

$$\frac{n*T}{(n*T+n*ceil(T/Q)*S)}$$

d) Q=S

$$\frac{n*T}{(n*T+n*ceil(T/Q)*S)}$$

becomes

$$\frac{n*T}{(n*T+n*T)}=1/2$$

e)if Q nearly 0

$$\frac{n*T}{(n*T+n*ceil(T/Q)*S)}$$

Here the denominator will become large reducing the efficiency

3) Most round robin schedulers use a fixed size quantum. Give an argument in favor of a small quantum. Now give one in favor of a large quantum.

A small quantum reduces the response time for all processes, which is important for interactive processes. However, a long quantum reduces the overhead of process switching, which increases throughput and CPU utilization. A short quantum is useful for a general-purpose computer. A long quantum is useful for a batch system. You might have a system that uses a short quantum when there are jobs that need a quick response time, then lengthens the quantum when heavy computation needs to be done.

4) Five batch jobs A through E, arrive at a computer center at almost the same time.They have estimated running times of 10,6,2,4 and 8 minutes. Their (externally determined) priorities are 3,5,2,1 and 4 respectively with 5 being the highest priority. For each of the following scheduling algorithm, determine the mean process turnaround time. Ignore process switching overhead.

a. Round robin.

b. Priority Scheduling

c. First-come, first served (run in order 10, 6, 2, 4, 8)

d. Shortest job first.

For (a) assume that the system is multi-programmed and that each job gets its fair share of the CPU. For (b) through (d0 assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

Turnaround time is the amount of time that elapses between the job arriving and the job completing. Since we assume that all jobs arrive at time 0, the turnaround time will simply be the time that they complete.

(a) Round Robin:

Here, we assume that the time quantum of the scheduler is 1 second. The table below gives a break down of which jobs will be processed during each time quantum. A * indicates that the job completes during that quantum.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

A B C D E A B C* D E A B D E A B D* E A B E A B* E A E A E* A A*

Average turnaround = (8 + 17 + 23 + 28 + 30)/5 = 106/5 = 21.2 minutes

(b) Priority Scheduling:

1-6 7-14 15-24 25-26 27-30

B E A C D

Avg. turnaround = (6 + 14 + 24 + 26 + 30)/5 = 100/5 = 20 minutes

(c) FCFS

1-10 11-16 17-18 19-22 23-30

A B C D E

Avg. turnaround = (10 + 16 + 18 + 22 + 30)/5 = 96/5 = 19.2 minutes

(d) SJF

1-2 3-6 7-12 13-20 21-30

C D B E A

Avg. turnaround = (2 + 6 + 12 + 20 + 30)/5 = 70/5 = 14 minutes

5) Five jobs are waiting to be run. Their expected run times are 9,6,3,5 and X. In what order should they run to minimize average response time? (Your answer will depend on X)

`For five task t1..t5 with run times a,b,c,d and e respectively`

which are batch processing type

```
t1 response time =a
t2 response time =a+b
t3 response time =a+b+c
t4 response time =a+b+c+d
t5 response time =a+b+c+d+e
```

Avg response time is given as Response time $= \frac{(5a+4b+3c+2d+e)}{5}$

Shortest job first is the way to minimize average response time (or Shorted Remaining Time to Completion First).For different values of X, order of execution of jobs and avg response time are as follows

0 < X <= 3: X, 3, 5, 6, 9. Avg response time = $\frac{(5*x+4*3+3*5+2*6+9)}{5} = x + 48/5$

3 < X <= 5: 3, X , 5, 6, 9 . Avg response time = $\frac{(5*3+4*x+3*5+2*6+9)}{5} = (4*x+51)/5$

5 < X <= 6: 3, 5, X, 6, 9. Avg response time = $\frac{(5*3+4*5+3*x+2*6+9)}{5} = (3x+56)/5$

6 < X <= 9: 3, 5, 6, X, 9. Avg response time = $\frac{(5*3+4*5+3*6+2*x+9)}{5} = (2x+62)/5$

X> 9: 3, 5, 6, 9, X. Avg response time = $\frac{(5*3+4*5+3*6+2*9+x)}{5} = (71+x)/5$