1999

Operating Systems (1a): Tripos Questions 1998/99

1 Paper 1

Question 1 (10 Marks)

An operating system uses a single queue round-robin scheduling algorithm for all processes. You are told that a *quantum* of three time units is used.

What can you infer about the scheduling algorithm? [1 mark]

Why is this sort of algorithm suitable for a multi-user operating system? [1 mark]

The following processes are to be scheduled by the operating system.

Process	Creation Time	Required Computing Time
P_1	0	9
P_2	1	4
P_3	7	2

None of the processes ever blocks. New processes are added to the tail of the queue and do not disrupt the currently running process. Assuming context switches are instantaneous, determine the *response time* for each process. [6 marks]

Give one advantage and one disadvantage of using a small quantum. [2 marks]

Answers to Question 1

Inference

You can infer that the scheduling algorithm is *preemptive*. It makes no sense to talk about a quantum for non-preemptive scheduling.

Multi-User OS

A preemptive scheduling algorithm is suitable for a multi-user OS because it does not require that processes co-operate (as a non-preemptive one does).

Determining Response Times

The point of this question is to test the students' understanding of preemptive scheduling. The first 4 marks should be awarded for something resembling the table.

The schedule would proceed as follows, where R_i is the amount of time for which process P_i must yet be run.

Time	Current	Queue	R_1	R_2	R_3
0	P_1	P_1	9	—	—
1	P_1	P_1, P_2	8	4	—
2	P_1	P_1, P_2	7	4	—
3	P_2	P_2, P_1	6	4	—
4	P_2	P_2, P_1	6	3	—
5	P_2	P_2, P_1	6	2	—
6	P_1	P_1, P_2	6	1	—
7	P_1	P_1, P_2, P_3	5	1	2
8	P_1	P_1, P_2, P_3	4	1	2
9	P_2	P_2, P_3, P_1	3	1	2
10	P_3	P_3, P_1	3	0	2
11	P_3	P_3, P_1	3	0	1
12	P_1	P_1	3	0	0
13	P_1	P_1	2	0	0
14	P_1	P_1	1	0	0
15	—	—	0	0	0

A further two marks should go for the final computation; out-by-one errors (due to misunderstanding the nature of discrete time) probably deserve at least one mark.

P_1 starts at 0, finishes by 15: hence its response time is 15.

P_2 starts at 1, finishes by 10: hence its response time is 9.

P_3 starts at 7, finishes by 12: hence response time is 5.

Small Quantum

One mark for an advantage; one mark for a disadvantage.

Having a small quantum is an advantage in that it can reduce the average response time, particularly for short (e.g. I/O bound) runs. Processes do not have to wait as long until they get their chance to run (in general will need to wait $(n-1) \times q$ time units, where there are n processes in the run queue, and the quantum is q time units).

The down side is that context switches are *not* free. The smaller the quantum, the more frequent the context switches, and hence the larger the overall scheduling overhead.