From last time

Explain why the time slice in pre-emptive process scheduling algorithms is normally significantly longer than the time needed for a context switch (2 marks) (define terms first) to minimise overhead

Why is a schedule giving lowest average turnaround time the same as that giving lowest average waiting time? (1 mark)

\[
\text{turnaround time} = \text{run time} + \text{wait time}
\]

Given a set of jobs with known processing time, all available to run, explain why repeatedly running the shortest job next gives the lowest average turnaround time. (3 marks)

\[
\text{runtime of 1st process} = \text{waiting time for all other processes}
\]

What is a CPU burst and an I/O burst? What is a CPU-bound and an I/O bound process? Why is it a good strategy in process scheduling to give higher priority to I/O bound processes? (4 marks) (definitions; I/O bound = smaller CPU burst)

As previous Q
Shortest Remaining Time First/Next (SRTF)

Preemptive (not time-sliced) version of SJF:

For each newly-ready process:
if CPU-burst < time to complete running process,
then context-switch & run the new process

Question: as previous example, but staggered arrival:

A 1 6
B 4
C 5 9
D

Shorter average waiting time than (non-preemptive) SJF
Example ctd.

2nd scenario: (repeatedly)
– 3 quanta for Q1, then 2 quanta for Q2, then 1 quantum for Q3
– each queue applies round-robin (time-slice = 1 quantum)

Question: A-F as before:

A (Q1)

B (Q1)

C (Q3)

D (Q3)

E (Q2)

F (Q1)
example ctd.

Three processes A, B, C (in the 3rd queue)
initial priority & quantum A=23, B=22, C=21

A: 15 CPU, 28 I/O, 44 CPU, 24 I/O, 21 CPU ...
B: 40 CPU, 22 I/O, 27 CPU ...
C: 3 CPU, 9 I/O, 51 CPU ...

recomputes: 40↑ 115↑ 183↑

Question: what triggers each recompute? what new quanta?
- A: \((23-15)/2+23=27\) B:22 C:\((21-3)/2+21=30\)
- A:23 B:22-3-15)/2+22=24 C:21
- A:23 B:22 C:21