

1.2 SOFTWARE & SOFTWARE DEVELOPMENT · 1.2.1

Interrupts & scheduling

Interrupts and the **ISR**, and the five **scheduling algorithms**. Spec 1.2.1(c)(d).

01 Interrupts

What A signal to the CPU requesting attention.

Sources Hardware · software · timer.

Why Make multi-tasking possible.

02 Handling an interrupt

1**Pause**

Finish the current instruction; save state (registers + PC) to the stack.

2**Service**

Load and run the ISR for that interrupt.

3**Resume**

Restore state from the stack; continue. Priorities decide order.

03 Scheduler aims

Goals Max throughput · fairness · good response time · keep resources busy.

SJF vs

SRT SJF = smallest **total** time; SRT = smallest **remaining** time (pre-emptive).

Quantum The fixed time slice used by round robin.

04 Five scheduling algorithms

Round

robin Fixed time slice (quantum) each, in turn; unfinished → back of queue. Fair, good response.

FCFS Arrival order, no priorities. Simple; a long job blocks the rest.

Shortest job

first Smallest estimated **total** run time next.

Shortest remaining

time Smallest estimated **time left** next; can pre-empt.

Multi-level feedback

queues Several priority queues; jobs move between them; favour short / I/O-bound jobs.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Seven slips on interrupts and scheduling questions.

01

An interrupt **pauses** the current process; the OS saves its state, runs the ISR, then resumes it.

02

State is saved to the **stack** (registers + program counter) so the process resumes exactly where it stopped.

03

Interrupts have **priorities** — a higher one can interrupt a running ISR.

04

Describe round robin with its **time slice / quantum** — that is the key detail.

05

SJF = total time; **SRT** = remaining time and pre-emptive. Don't mix them up.

06

FCFS is simple but a long job **blocks** everything behind it — poor for interactive use.

07

For "which algorithm suits system X": match interactive/real-time needs to **round robin**, batch/overnight jobs to **SJF/SRT**, and justify with the scheduler's goals.