

16 Scheduling and context switch – Part 2

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English	Write it 3 times (English)
scheduler	_____
Round robin	_____
time slice	_____
process control	_____
block	_____

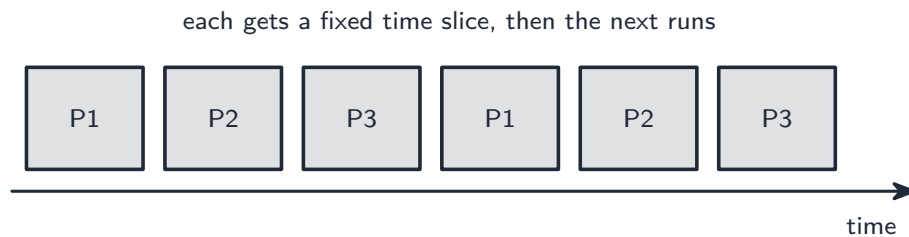
Recall

When many processes are ready, the OS must choose which runs next, and switch between them cleanly.

New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

■ 1 Round-robin scheduling



The **scheduler** picks which ready process runs. **Round robin** gives each a fixed **time slice**, then sends it to the back of the queue.

Worked example. Three processes share a 10 ms slice each: P1 runs for 10 ms, then is paused and P2 runs 10 ms, then P3, then back to P1 – round and round, so all make steady progress.

■ 2 The context switch



To switch processes, the OS saves the running one's state in its **process control block** (PCB) and loads the next one's.

Worked example. To switch from A to B, the OS saves A's registers and program counter into A's PCB, then loads B's saved values from B's PCB – so B carries on exactly where it left off.

Watch out: a context switch does no useful program work itself – it is pure overhead. Too short a time slice means *many* switches and lots of wasted time; too long a slice hurts responsiveness.

Practice

Try these in order.

16.1 What does the *scheduler* decide? [1]

16.2 In *round-robin*, what does each process get? [1]

16.3 After its time slice ends, where does a process go? [1]

16.4 What does the OS save in a *context switch*? [1]

16.5 What does *PCB* stand for? [1]

16.6 Explain why a context switch is “wasted” time, even though it is necessary. [2]

16.7 Challenge. Explain the trade-off in choosing the length of the round-robin time slice – very short versus very long. [2]
