16 How an OS shares resources – Part 1

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English	**Write it 3 times** (English)		
Multi-tasking			
process			
blocked			

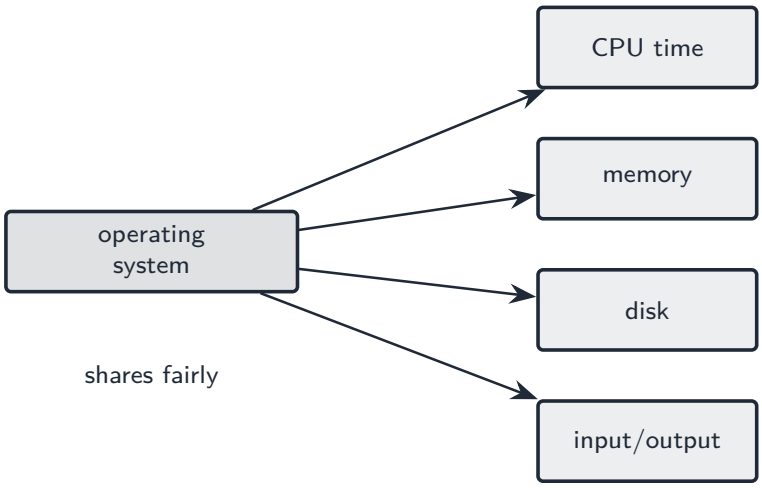
Recall

Many programs compete for one computer. The operating system shares its resources fairly and keeps things responsive.

New ideas

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

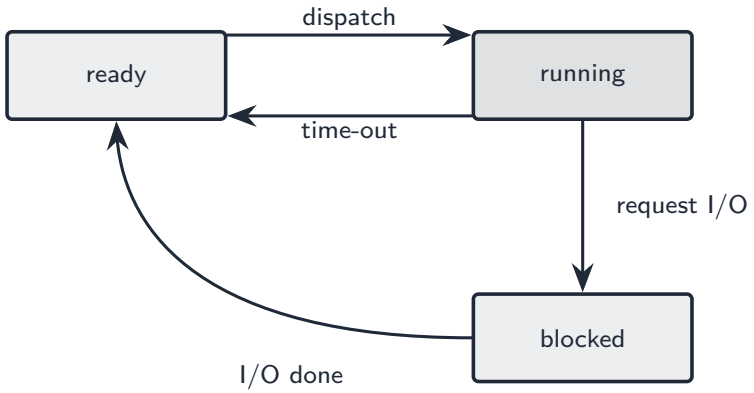
1 Sharing resources



The OS shares the CPU, memory, disk and I/O. **Multi-tasking** switches the CPU quickly between programs so several seem to run at once.

Worked example. You play music and type a document at the same time. On one CPU core only one can actually run at any instant, but the OS switches between them thousands of times a second, so both appear to run together.

2 Process states



A **process** is a program in execution. It is **ready** (waiting for the CPU), **running**, or **blocked** (waiting for I/O).

Worked example. A process asks to read a file (slow I/O), so it goes to *blocked* while it waits. When the data arrives it becomes *ready*; the scheduler later gives it the CPU, so it is *running* again.

Watch out: on a single core only *one* process truly runs at a time – multi-tasking

> is fast switching, not real simultaneity. And a *blocked* process is only waiting, not finished.

Practice

Try these in order.

16.1 Name two resources the OS shares. [2]

16.2 What does *multi-tasking* do? [1]

16.3 What is a *process*? [1]

16.4 What does it mean for a process to be *blocked*? [1]

16.5 A running process uses up its turn. Which state does it go to? [1]

16.6 On a single-core CPU, explain why two programs only *appear* to run at the same time. [2]

16.7 Challenge. A process is waiting for the user to type. Explain why the OS moves it to *blocked* rather than leaving it running, and what this does for the other programs. [2]