Exercise walk-through

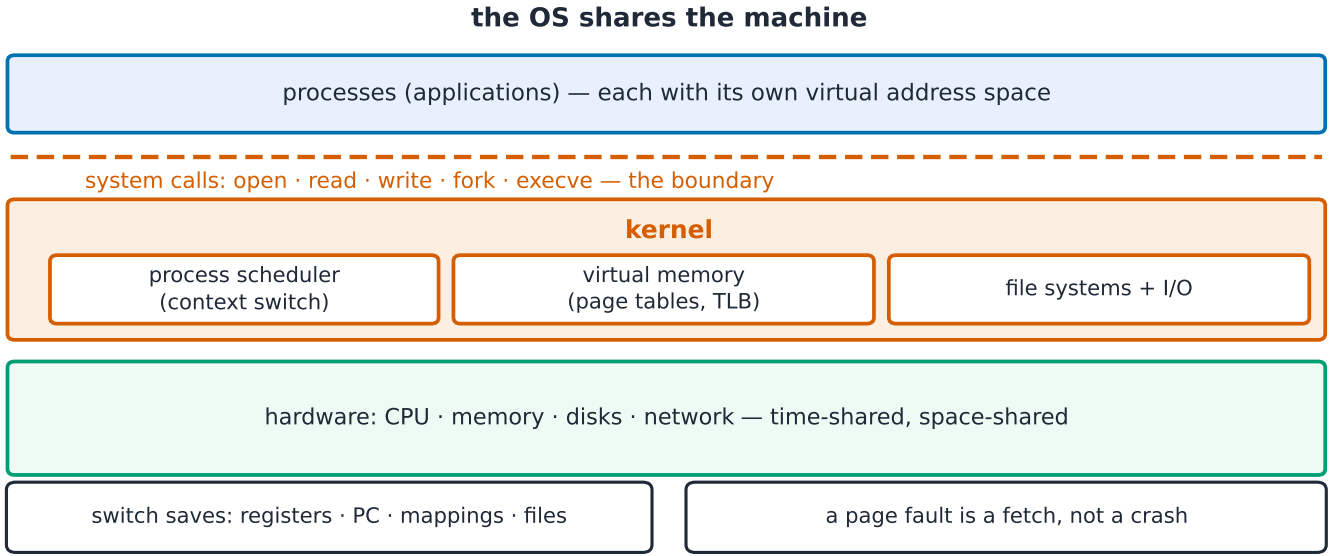
3.5
The operating-system interface

University Computer Science

The operating-system interface: services, not hardware

1. **Process** Program + state; a **context switch** saves registers, PC, memory mappings, open-file table — exactly what the next instruction needs.
2. **Virtual memory** Page tables + TLB give each **process** its own addresses: isolation, over-commit, fixed link addresses.
3. **Calls** open · read · write · fork · execve cross into the kernel; a **page fault** is demand paging working.

The OS layer diagram



System calls cross the boundary; the kernel shares CPU, memory and devices.

Exercises 2.1·2.3

2.1 Name the four items a context switch saves. [3]

2.2 What translates virtual to physical addresses? [3]

2.3 Name three system calls for files. [3]

Exercises 2.1·2.3

2.1 Name the four items a context switch saves. [3]

2.1

- Registers, PC, memory mappings, open-file table ✓.

2.2 What translates virtual to physical addresses? [3]

2.2

- The page tables (with the TLB as their cache) ✓.

2.3 Name three system calls for files. [3]

2.3

- open, read, write (close also) ✓.

Exercise 3.1

3.1 Two processes A and B timeshare one CPU; A faults on a page.

Explain the sequence of events from the fault to A resuming. State what of A is preserved during the wait and what the CPU does meanwhile. Name the states A passes through, in order. [8]

Exercise 3.1

3.1 Two processes A and B timeshare one CPU; A faults on a page.

Explain the sequence of events from the fault to A resuming. State what of A is preserved during the wait and what the CPU does meanwhile. Name the states A passes through, in order. [8]

3.1

- A faults → blocked; scheduler runs B; disk completes → A ready; scheduler resumes A with its saved state ✓.
- Preserved: registers, PC, mappings, files – the CPU cache is shared hardware and is merely cold ✓.
- The CPU runs B during A's wait – a page fault creates useful I/O concurrency ✓.
- States: running → blocked → ready → running ✓.

Exercise 3.2

3.2 Why virtual memory serves three needs at once.

Name the three needs virtual memory serves at once. For each, state the mechanism that provides it and what would break without it. [7]

Exercise 3.2

3.2 Why virtual memory serves three needs at once.

Name the three needs virtual memory serves at once. For each, state the mechanism that provides it and what would break without it. [7]

3.2

- Isolation gone: any bug in one process could read or corrupt another's memory ✓.
- Illusion gone: total program size capped by physical RAM – big working sets fail to load ✓.
- Convenience gone: every load would need relocation against the machine's actual physical layout ✓.