Exercise walk-through

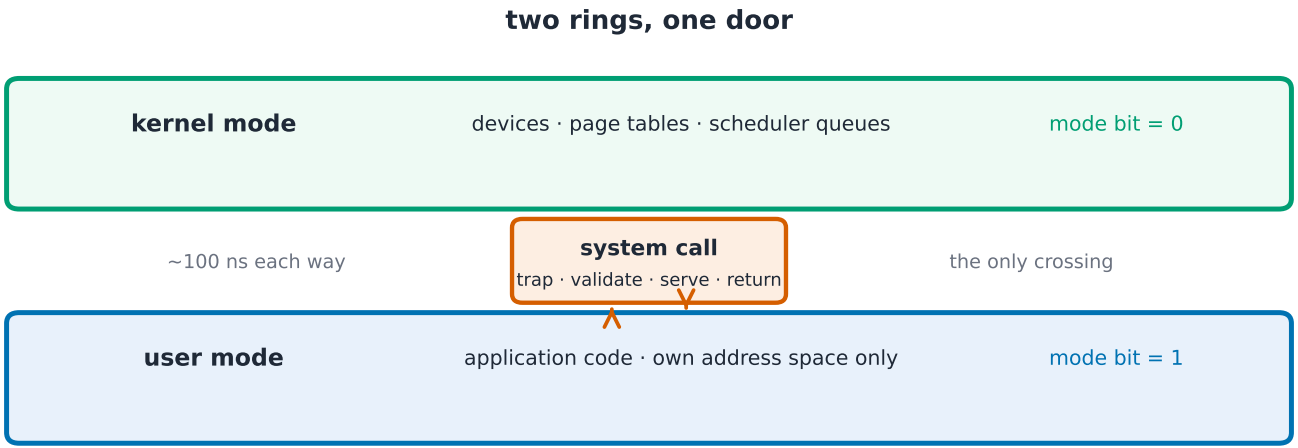
7.6
Protection and security

University Computer Science

Protection: two rings, one door

1. **Rings Kernel mode** touches devices and page tables; user mode crosses only by trap - two crossings, about 100 ns each.
2. **Batch** A million syscalls cost 0.2 s of crossings - the reason libraries buffer and batch.
3. **Deputy** The confused printer daemon obeys the user with its own authority - hand it capabilities, not file names.

Privilege boundary



least privilege: 9 permission bits per file; validate every crossing; pass capabilities, not names

> User code crosses to the kernel only through the syscall door.

Check yourself

- Waiting on disk I/O is the... state? (**Blocked**.)
- Course set average turnaround under **SJF**...? (7.75.)
- 16-bit VA, 256-byte pages: virtual pages...? (256.)
- The only user-to-kernel crossing is the...? (**System call**.)

Exercises 2.1·2.3

2.1 Name the instruction that crosses user to kernel, and the mechanism that picks the handler. [3]

2.2 Give the 9 permission bits as a triple of triples for rw-r——. [3]

2.3 Why is passing an open descriptor safer than passing a file name? [3]

Exercises 2.1·2.3

2.1 Name the instruction that crosses user to kernel, and the mechanism that picks the handler. [3]

2.1

- The trap instruction; a fixed vector table picks the handler ✓.

2.2 Give the 9 permission bits as a triple of triples for rw-r——. [3]

2.2

- Owner rw-, group r–, other —: (110)(100)(000) ✓.

2.3 Why is passing an open descriptor safer than passing a file name? [3]

2.3

- The descriptor is authority the caller already earned – the daemon cannot widen it; a name asks the deputy to use its own greater authority ✓.

Exercise 3.1

3.1 A program issues 5000 syscalls; each crossing costs 120 ns.

Compute the total crossing overhead. Compute how many syscalls a buffering library must batch into one to cut the overhead by a factor of 100. State what the kernel must validate at every crossing regardless of batching, naming two checks. [8]

Exercise 3.1

3.1 A program issues 5000 syscalls; each crossing costs 120 ns.

Compute the total crossing overhead. Compute how many syscalls a buffering library must batch into one to cut the overhead by a factor of 100. State what the kernel must validate at every crossing regardless of batching, naming two checks. [8]

3.1

- Total: $5000 \times 2 \times 120$ ns $= 1.2$ ms ✓.
- Batch factor 100: one syscall per 100 calls – 50 syscalls at 240 ns each $= 12$ µs ✓.
- Validate always: the caller's identity and permissions, and every argument – pointers and lengths – at the boundary ✓.

Exercise 3.2

3.2 A printer daemon runs with privilege; a user asks it to print /salary.db.

Explain the confused-deputy problem in this scenario, naming whose authority opens the file. State the capability fix and why the daemon can then refuse safely. Give the general principle the scenario illustrates, and one way the deputy could still misbehave even with capabilities. [7]

Exercise 3.2

3.2 A printer daemon runs with privilege; a user asks it to print /salary.db.

Explain the confused-deputy problem in this scenario, naming whose authority opens the file. State the capability fix and why the daemon can then refuse safely. Give the general principle the scenario illustrates, and one way the deputy could still misbehave even with capabilities. [7]

3.2

- The daemon opens /salary.db with ITS OWN privilege, not the user's – the deputy is confused about whose authority to use ✓.
- Fix: the user passes an open descriptor (capability); the daemon prints only what it was handed and can refuse names outright ✓.
- Principle: least privilege – and validate every crossing ✓.
- Even with capabilities, a buggy deputy can print the content to the wrong place or leak it – authority is bounded, behaviour is not ✓.