ANDIJAN STATE PEDAGOGICAL INSTITUTE

Lecture materials · 2026–2027 academic year

SUBJECT

Information Systems and Digital Technologies

Lecture 10 · Operating System Functions and File Management

Part 1 of 2 · 2 hours

Lecturer: **Ismailov Alisher**

E-mail: alisherismailov1991@gmail.com

Contact: **99-499-73-64**

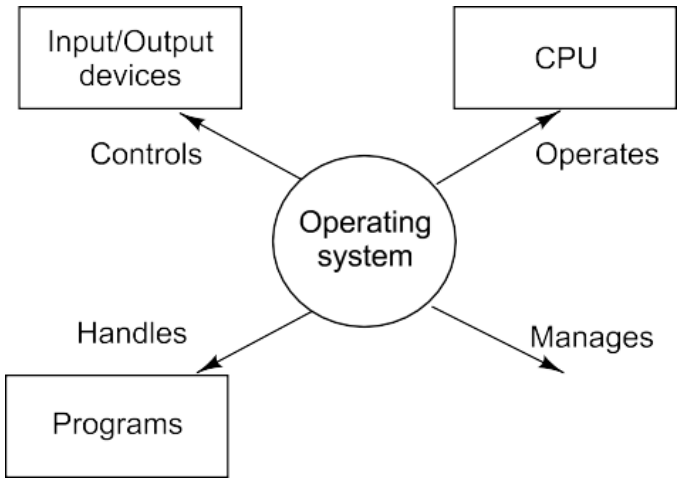
LECTURE 10 · 2 HOURS · PART 1/2

FUNDAMENTALS OF COMPUTERS · CHAPTER 11

Operating Systems

The master control program that makes the hardware useful

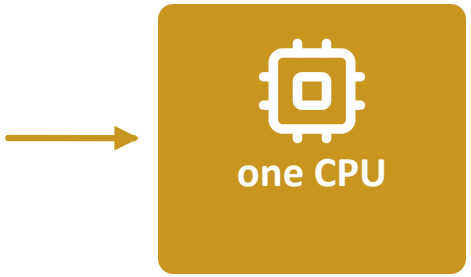
```
C:\> ver
process · memory · file · device · security
C:\> _
```



The roles of an operating system — textbook Fig. 11.1

Music, maps and chat — all at once. Who is in charge?

- Music player
- Maps
- Chat
- Printing a file



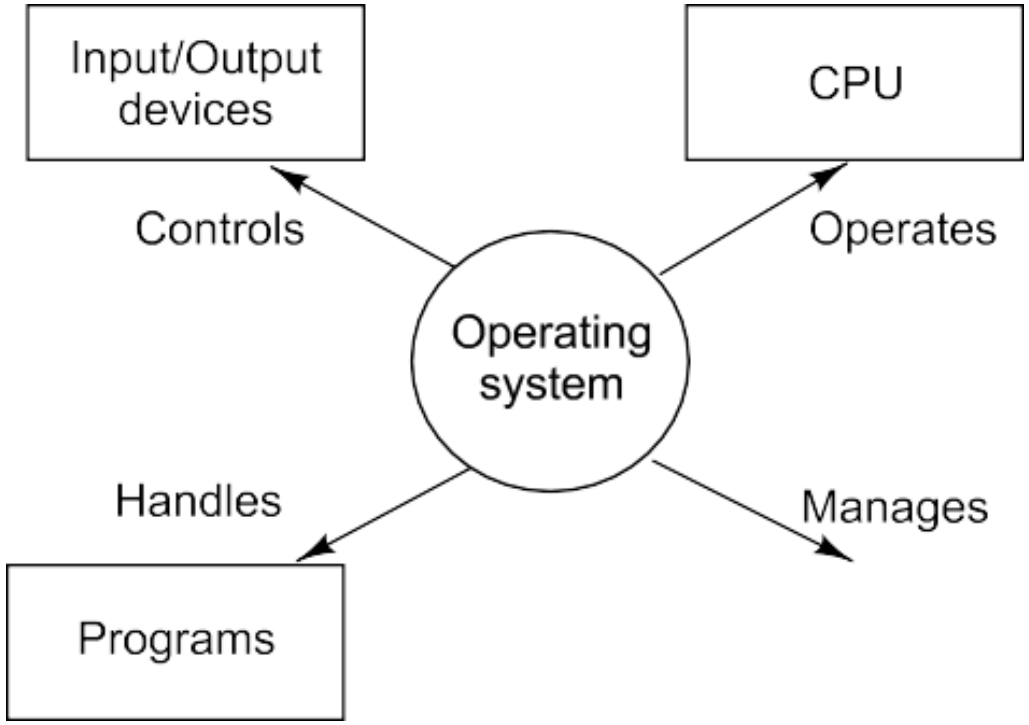
Think – Pair – Share (2 min)

1. Who decides which app uses the CPU next?
2. What happens if two apps want the printer at the same moment?
3. Name every operating system you have used.

By the end of this lesson, you can…

1. Explain what an operating system is and name its five main functions.
2. Describe the states of a process and how the OS schedules processes.
3. Identify deadlock, synchronisation and interprocess communication in examples.
6. List the services an OS gives to users and programs, and explain what a device driver does.

An operating system makes the hardware useful



The roles of an operating system — textbook Fig. 11.1

- Operates the CPU
- Controls input/output devices
- Handles application programs
- Manages storing and finding information on disks

The main part of system software — loaded first, before any other task.

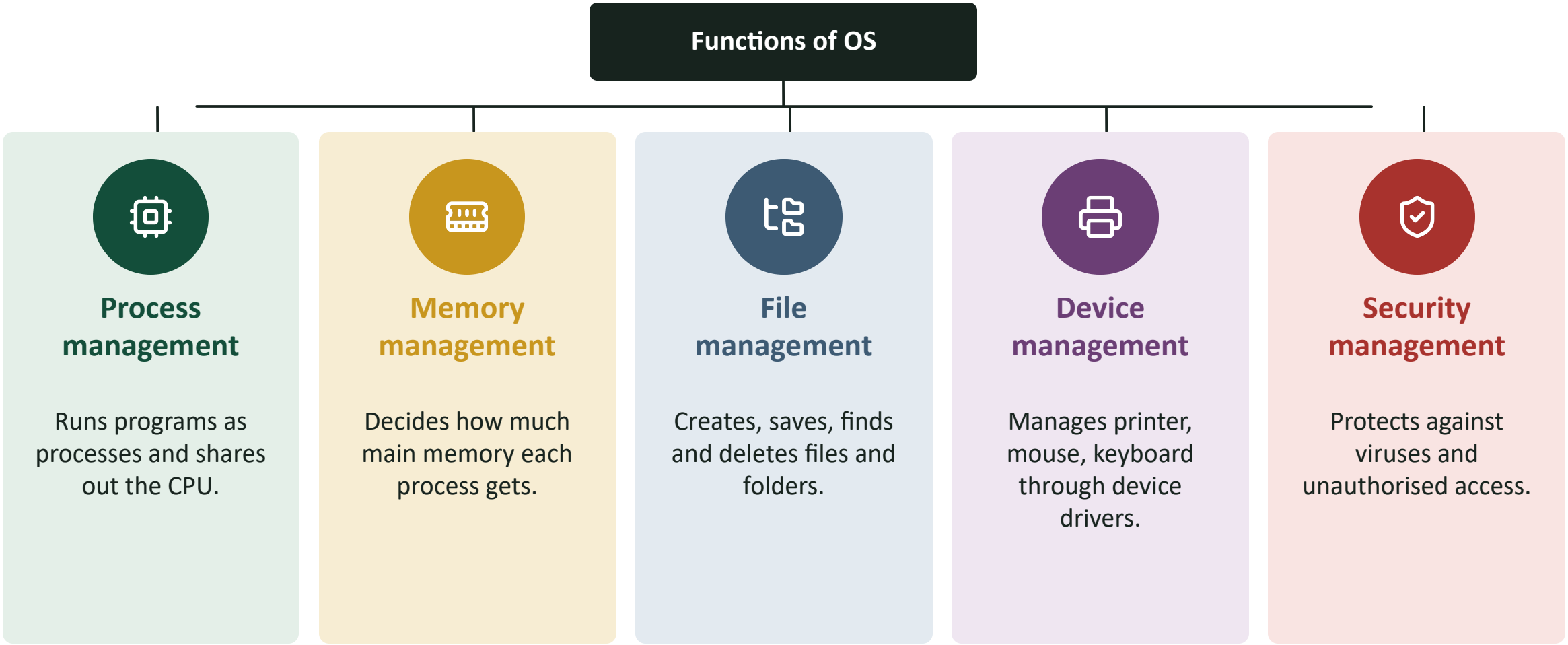
Seventy years of operating systems



1940s	1950s	1960s	1970s	1980s	1990s	2000s
No OS. Programs in assembly language talk directly to hardware.	Batch OS (BKY, CAL, Chios). One task at a time.	Multitasking: spooling and time-sharing. SABRE.	UNIX from AT&T Bell Labs: multi-user, with a kernel.	MS-DOS (Microsoft), HP-UX, Macintosh (Apple).	GUI era: Windows 95, 98, NT; FreeBSD; OS/2.	Windows XP, Server 2003, Mac OS X. Focus on security.

OPTIONAL MODERN NOTE Since this book: Windows 7, 8, 10 and 11, macOS, many Linux versions, and phone systems such as Android and iOS.

The OS is the resource manager: five jobs



Redrawn from textbook Fig. 11.2

CHECK FOR UNDERSTANDING

Which function is working?

You rename your essay and move it into a folder.

Your document is sent to the printer.

The computer asks for your password.

Word is loaded into RAM before it starts.

Three programs take turns on the CPU.

Choose from

- Process management
- Memory management
- File management
- Device management
- Security management

Services the OS provides

01

User interface

GUI, command line or touch screen.

02

Program execution

Load a program into memory, run it, end it.

03

I/O operations

Programs use devices only through the OS.

04

File-system manipulation

Create, read, write, delete, search files.

05

Communication

Processes exchange data, on one computer or a network.

06

Error detection

Find and handle hardware and software errors.

07

Resource allocation and accounting

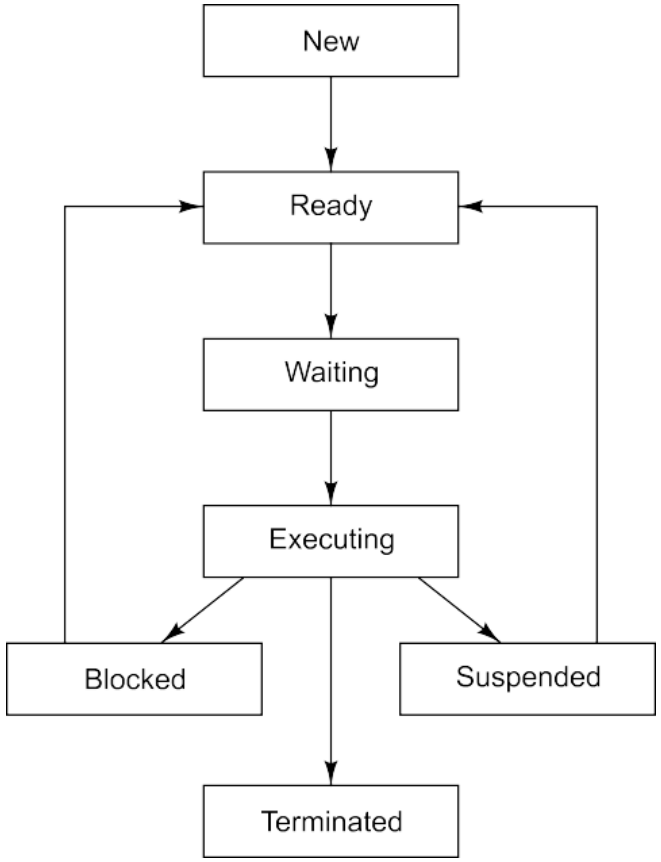
Share CPU, memory and devices; record who used what.

08

Protection and security

Control access; keep users and processes apart.

A process: a program that is running



Process states — textbook Fig. 11.3

`New`	the process is created
`Ready`	loaded into memory, ready to run
`Waiting`	waiting for CPU time and resources
`Executing`	being run by the CPU
`Blocked`	waiting for an event, e.g. I/O
`Suspended`	ready, but not placed in the ready queue
`Terminated`	finished; resources freed

Only one process uses the CPU at a time — the others wait in a queue.

Process Control Block: a process's ID card

Process id
Process state
Program counter
Register information
Scheduling information
Memory-related information
Accounting information
Status information related to I/O

Structure of a PCB — redrawn from textbook Fig. 11.4

Process id

a number that uniquely identifies the process

Program counter

address of the next instruction to run

Scheduling info

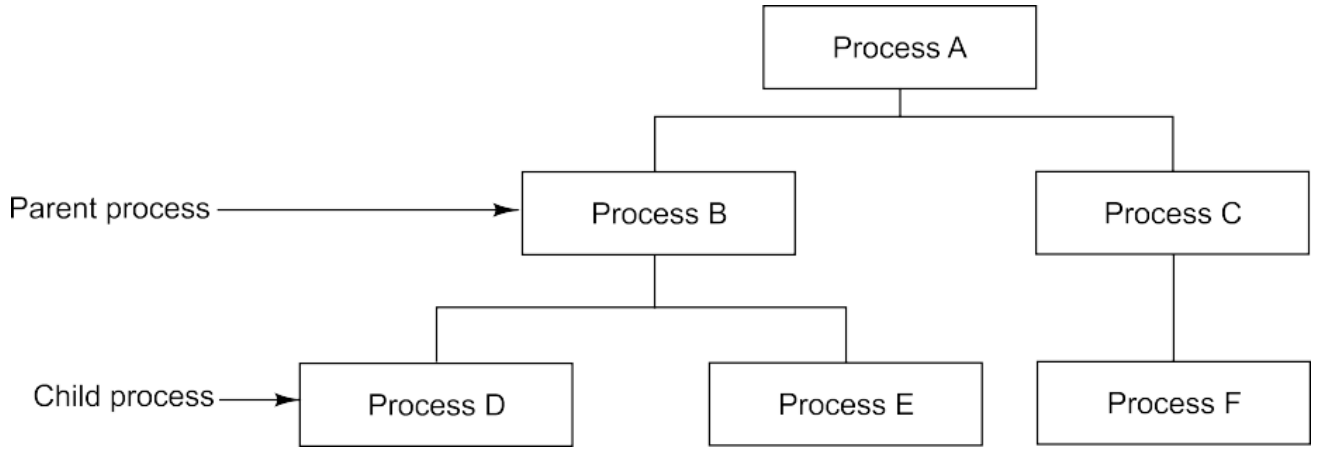
the process's priority

Accounting info

CPU use and execution time

Process table: the OS keeps every PCB in one table.

Creating and ending processes



Hierarchical structure of processes — textbook Fig. 11.5

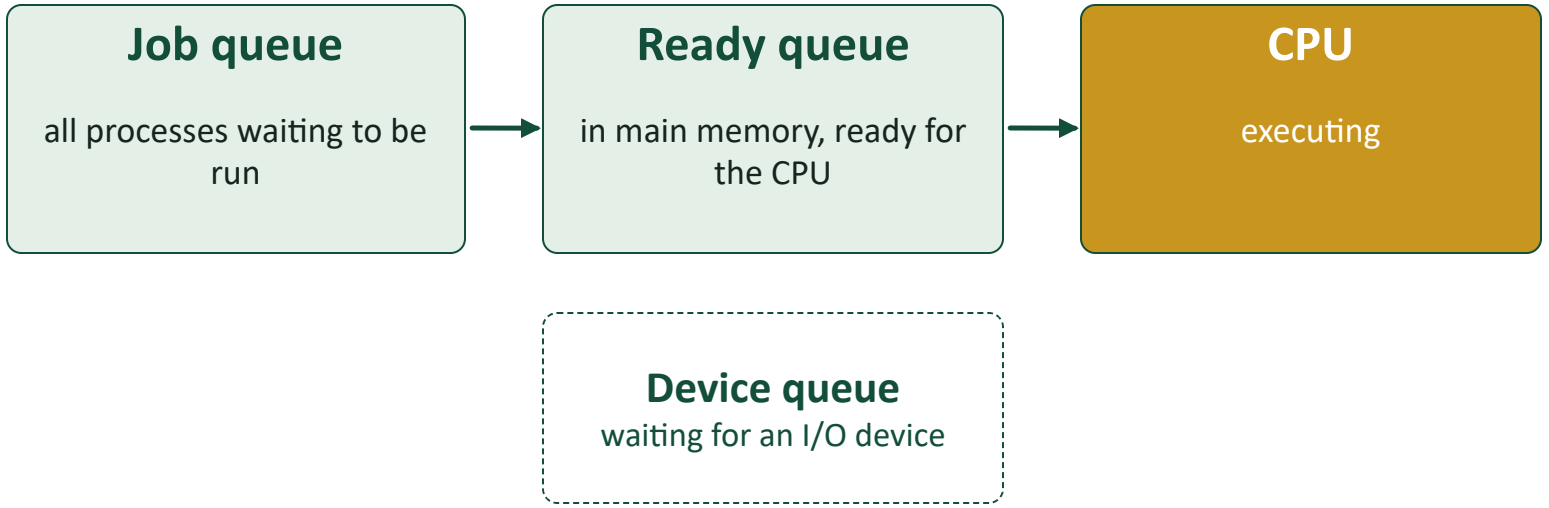
Process creation

- A parent process creates a child process with a system call such as fork( )
- New processes start when you log on, open MS Word or print

Process termination

Normal	Abnormal	Rule
all instructions ran successfully	the process stopped because of an error	if a parent ends, its children must end too

Scheduling: who gets the CPU next?



Two scheduling policies

Preemptive: a low-priority process must pause if a high-priority one is waiting.

Non-preemptive: the running process finishes before the next one starts.

Long-term scheduler

Chooses which processes go into the ready queue (main memory).

Mid-term scheduler

Swaps blocked or suspended processes out to secondary memory, and back in.

Short-term (CPU) scheduler

Chooses which ready process gets the CPU next.

Four scheduling algorithms

FCFS

First Come, First Served

The first process in the queue runs first.

SJF

Shortest Job First

The process needing the least CPU time runs first.

Priority

Priority scheduling

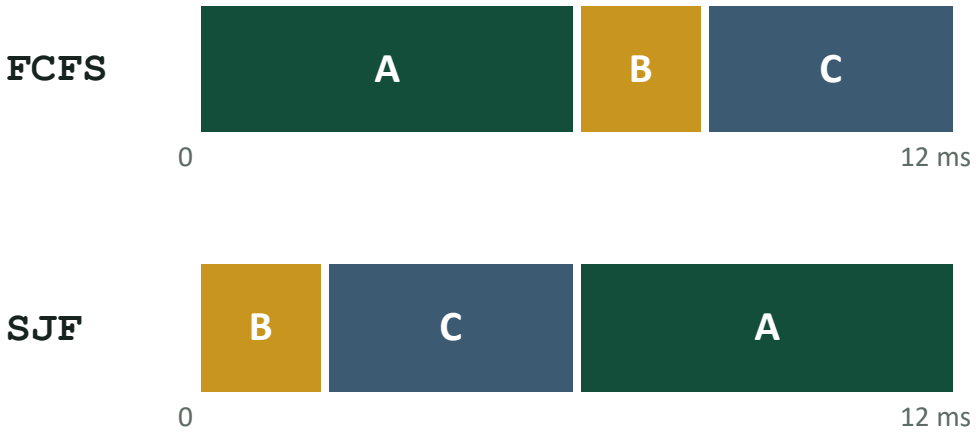
Highest priority first. Can be preemptive or not.

RR

Round Robin

Each process gets a time slice (10–100 ms), then waits for its next turn.

EXAMPLE: jobs arrive A (6 ms), B (2 ms), C (4 ms)



SJF lets the short jobs finish early, so on average processes wait less.

CHECK FOR UNDERSTANDING

Schedule it

Jobs in the queue (arrival order):

P	Q	R	S
5 ms	1 ms	3 ms	2 ms

1. In what order does FCFS run them?
2. In what order does SJF run them?
3. Which scheduler picks the next process for the CPU?

True or false?

4. In Round Robin, a time slice is usually 10–100 milliseconds.
5. In preemptive scheduling, a high-priority process can make a low-priority one pause.
6. A process in the device queue is waiting for the CPU.

Synchronisation: one at a time, please

The problem

Two processes change the same shared data at once. One change may be lost — the data becomes inconsistent.

Critical section

The code that changes shared data. Only ONE process may be inside it at a time.

THREE SOLUTIONS

Peterson’s solution

For two processes: while one runs its critical section, the other runs the rest of its code.

Locking

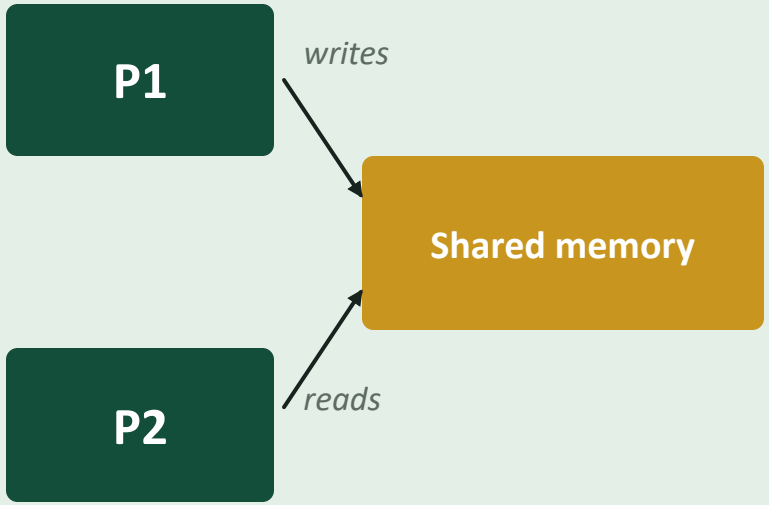
Take the lock before entering; release it when done.

Semaphore

An integer variable, changed by wait and signal operations, decides who may enter.

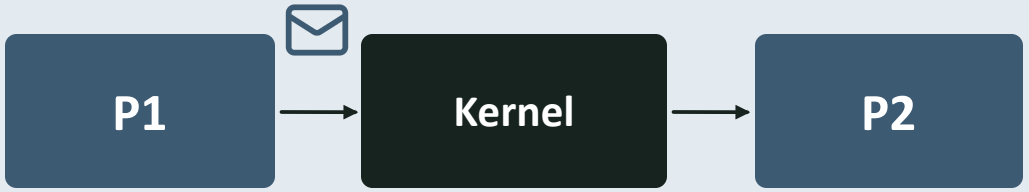
How processes talk to each other

Shared memory



P1 writes data into a shared part of memory; P2 reads it.

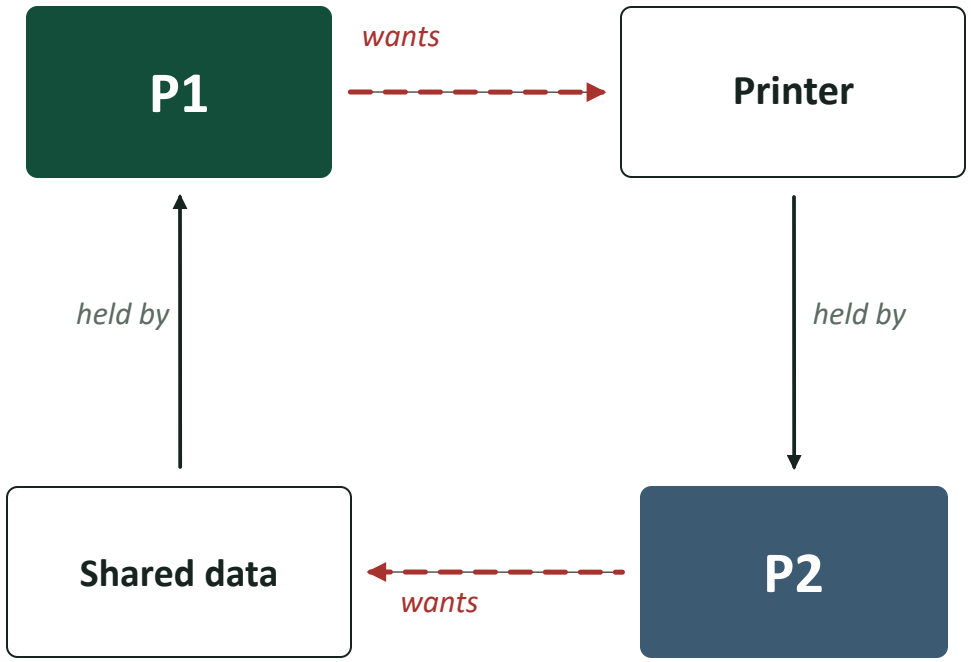
Message passing



P1 sends a message to the kernel, and the kernel passes it on to P2.

Redrawn from textbook Figs. 11.6 and 11.7

Deadlock: everyone waits for ever



Both wait for each other — nothing moves.

Why it happens

Mutual exclusion

resources can’t be shared

Hold and wait

hold one, wait for another

No preemption

can’t force a release

What the OS can do

Ignore it

assume it won’t happen

Detect & recover

preemption, rollback, or kill a process

Avoid it

only give a resource if no deadlock is possible

CHECK FOR UNDERSTANDING

Name that idea

A P1 writes a score into a part of memory; P2 reads it from there.

B Two processes each hold one resource and wait for the other's — for ever.

C Before changing a shared file, a process must take a lock.

D P1 sends data to the kernel, which forwards it to P2.

E The OS returns a process to an earlier saved state to break a deadlock.

F A data structure holds a process's id, state and program counter.

Review: can you answer these?

01 Why is the OS called the resource manager?

02 What is a process?

03 Name the seven process states.

04 What is a PCB, and where does the OS keep all of them?

05 How do SJF and Round Robin choose the next process?

06 What is a critical section?

07 How do shared memory and message passing differ?

08 Give the three conditions that can cause deadlock.

Exit ticket

Answer on a slip of paper before you leave.

1. Name three of the five functions of an operating system.

2. A process has been created and loaded into memory, but has not run yet. Which state is it in?

3. Jobs need 6, 2, 8 and 4 ms. In which order does SJF run them?