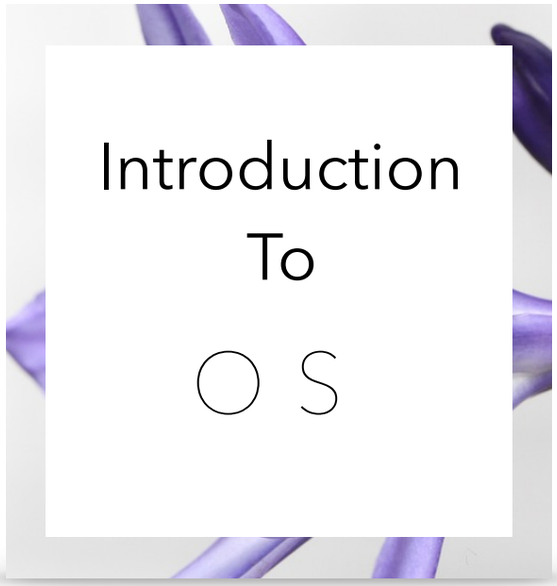


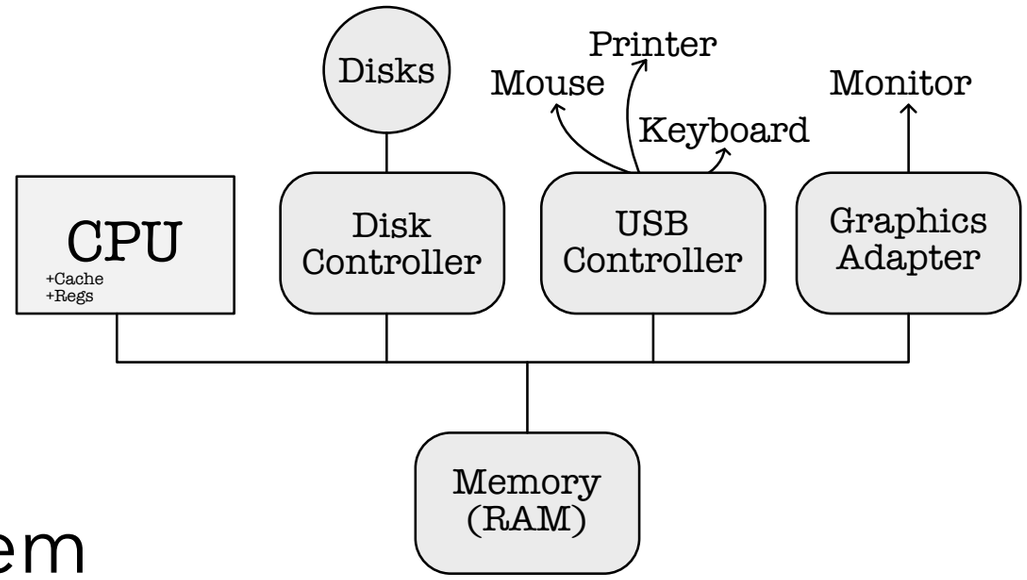
50.005 CSE

Natalie Agus
Information Systems Technology and Design
SUTD

WHAT IS AN OS?

One of the four parts of computer system:

- Hardware
- Operating System
- Programs
- User



DUAL MODE

User

Restricted, for security

Kernel

More privileges, access to hardwares like memory location

System call

WHAT IS AN OS **FOR**?

Functions of the OS:

- Resource allocator
- Control program executions

The heart of the OS is the **Kernel**

HOW DOES OS RUN?

1. Computer ON

Loads BIOS

2.BIOS Scans

Prep all attached devices to CPU in a state for OS

3. BIOS Loads MBR

Master Boot Record (disk address 0), contains info about disk partition as well as bootstrap code. The bootstrap code points to the bootloader

3. MBR Loads Bootloader

Bootloader is at bootsector (usually the start of the disk partition). The bootloader points to the entry codes of the OS. If there's multiple OS, then bootloader will ask user which OS to start

4. OS starts

OS bootstraps and starts

PURPOSES OF AN OS

Controls and coordinate use of hardware among various applications (programs) and users

1. Hardware & I/O control

WHAT ARE I/O DEVICES?

Input:
- Graphic tables
- Cameras
- Barcode reader
- Gamepad
- Joystick
- Mouse
- Keyboard
- Microphone
- Scanner
- Touchpads
- Pen input

Output:
- Monitor
- Printer
- Projector
- Speaker
- Headphones
- Monitors

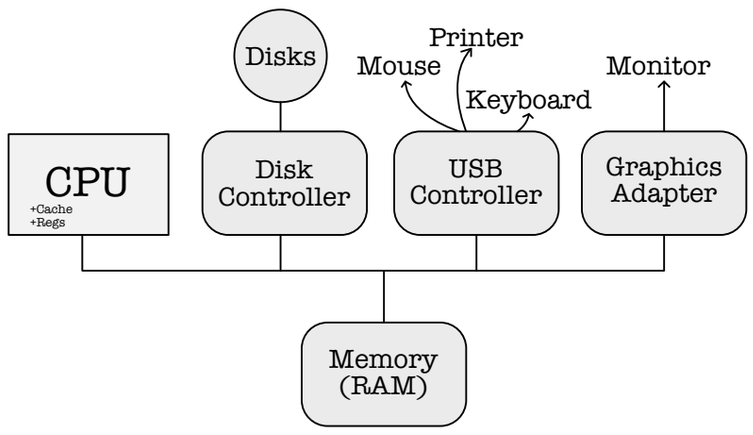
Both:
- Modems
- Network cards
- Touch screen
- FAX
- Sound card

HOW I/O DEVICES WORK

CPU, I/O device controllers, and I/O devices act **independently** of one another

Each I/O device has a **controller** (whats why you install *drivers*) that comes with a ***local buffer***

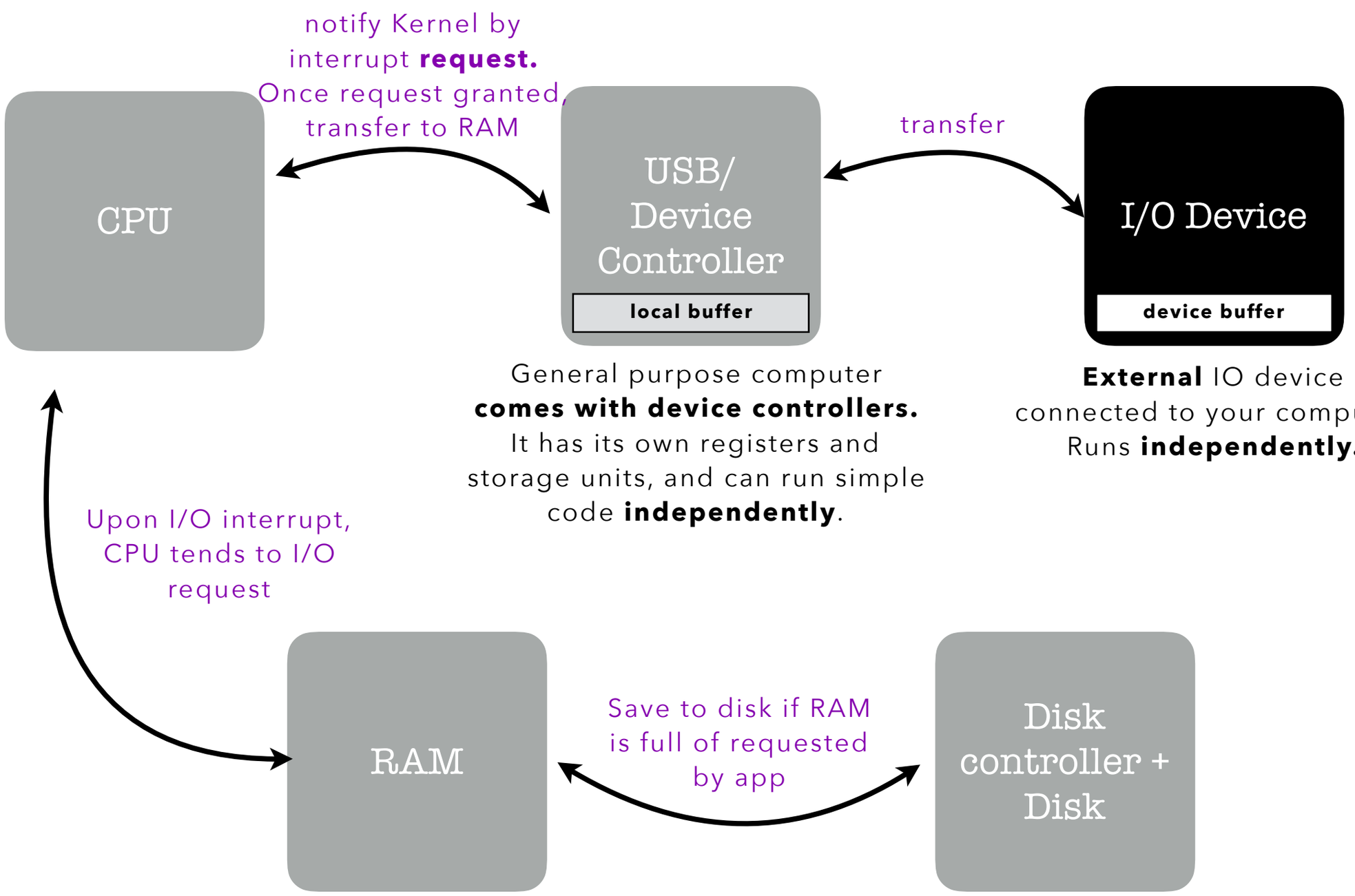
I/O devices and controllers can run code an starts activity on its own too



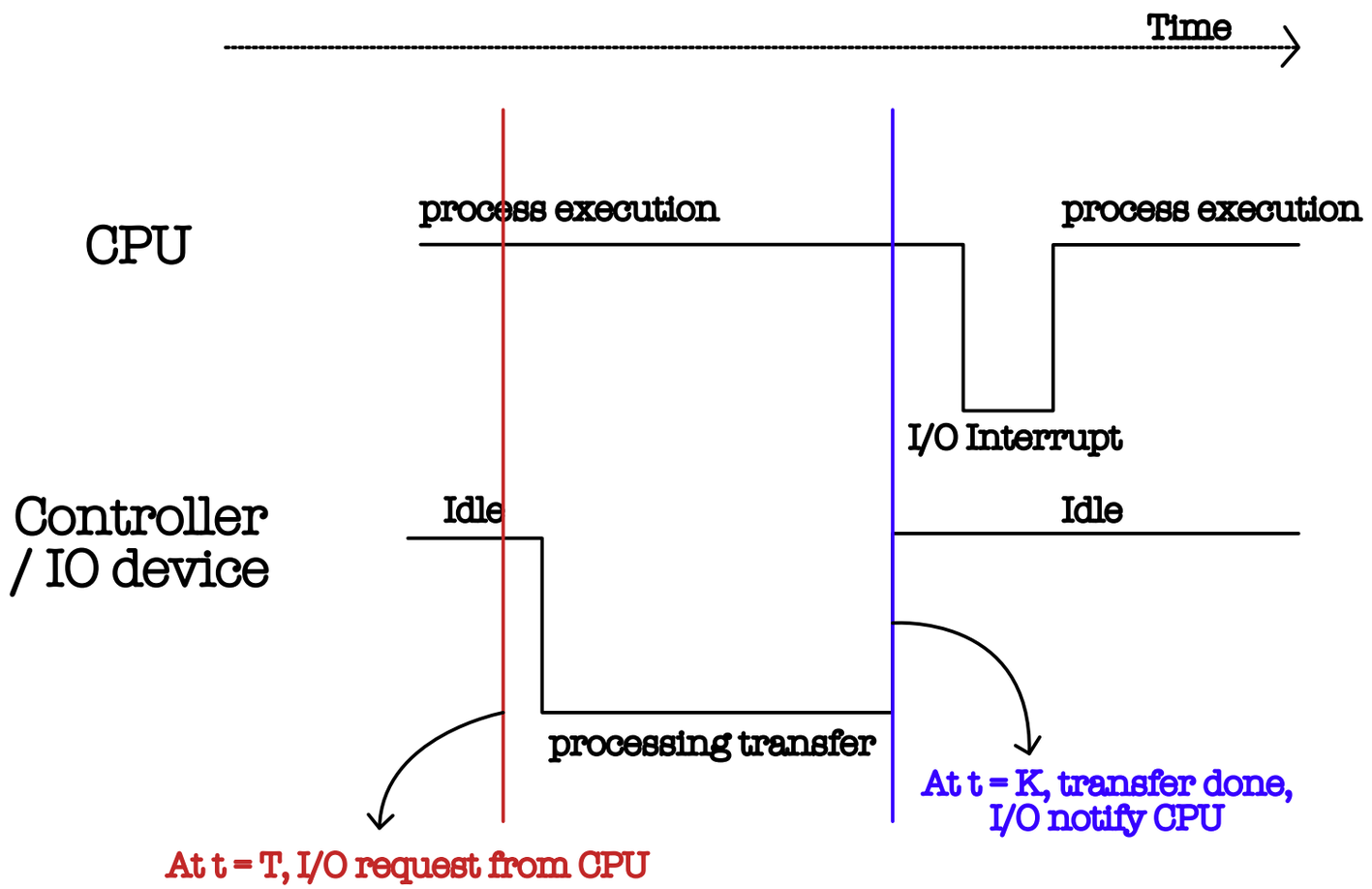
(A) CPU wants to move data to/from device controller buffer from/to memory (RAM)
(B) I/O happens when data is moved from/to device controller buffer to/from device

We need **coordination** to do step (A) and (B) above: with *interrupts*

HOW I/O DEVICES WORK

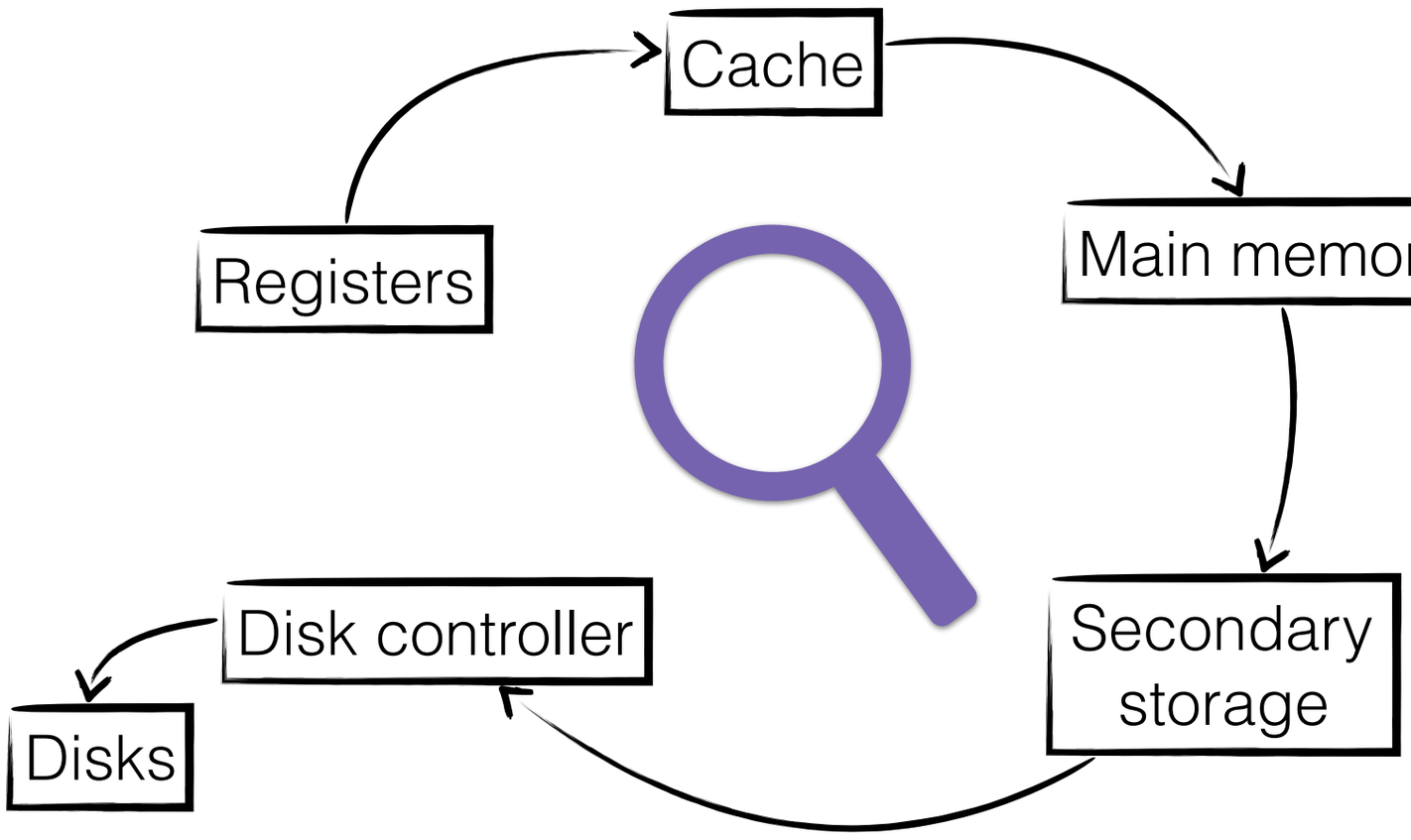


PURPOSES OF AN OS



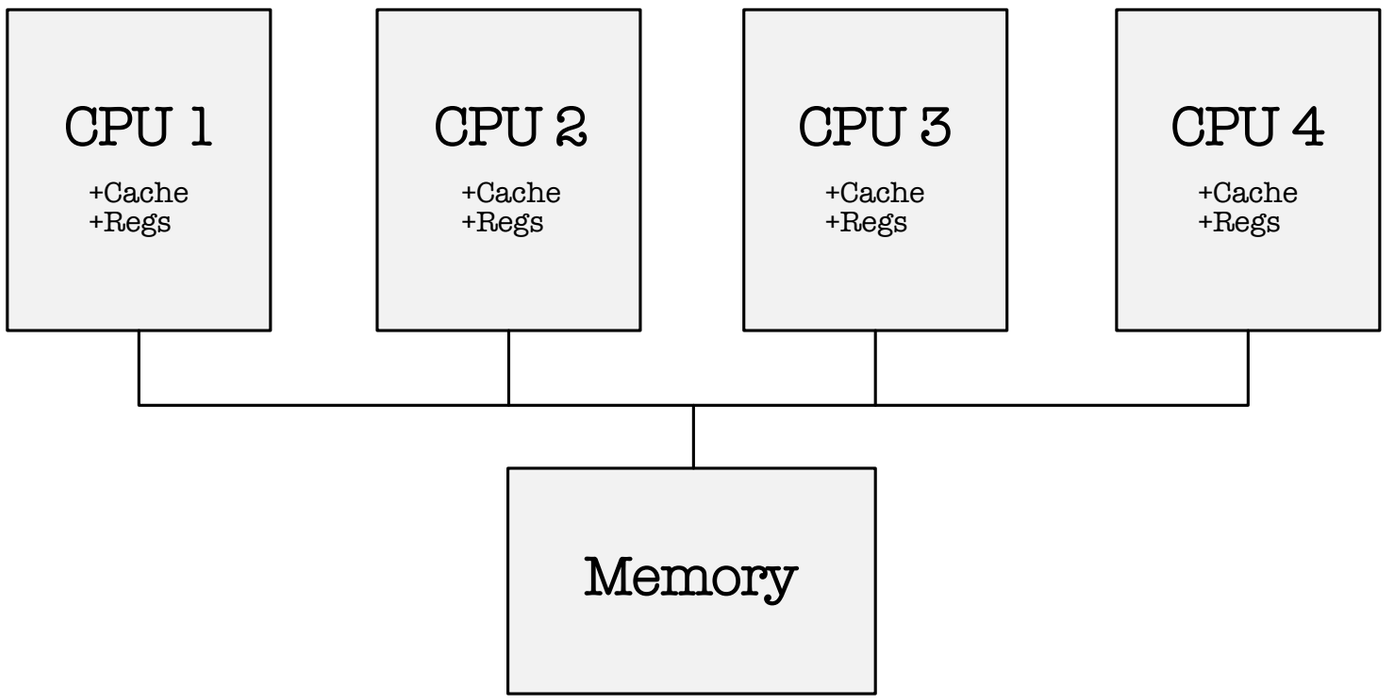
2. Handles interrupts: vectored and polling

PURPOSES OF AN OS



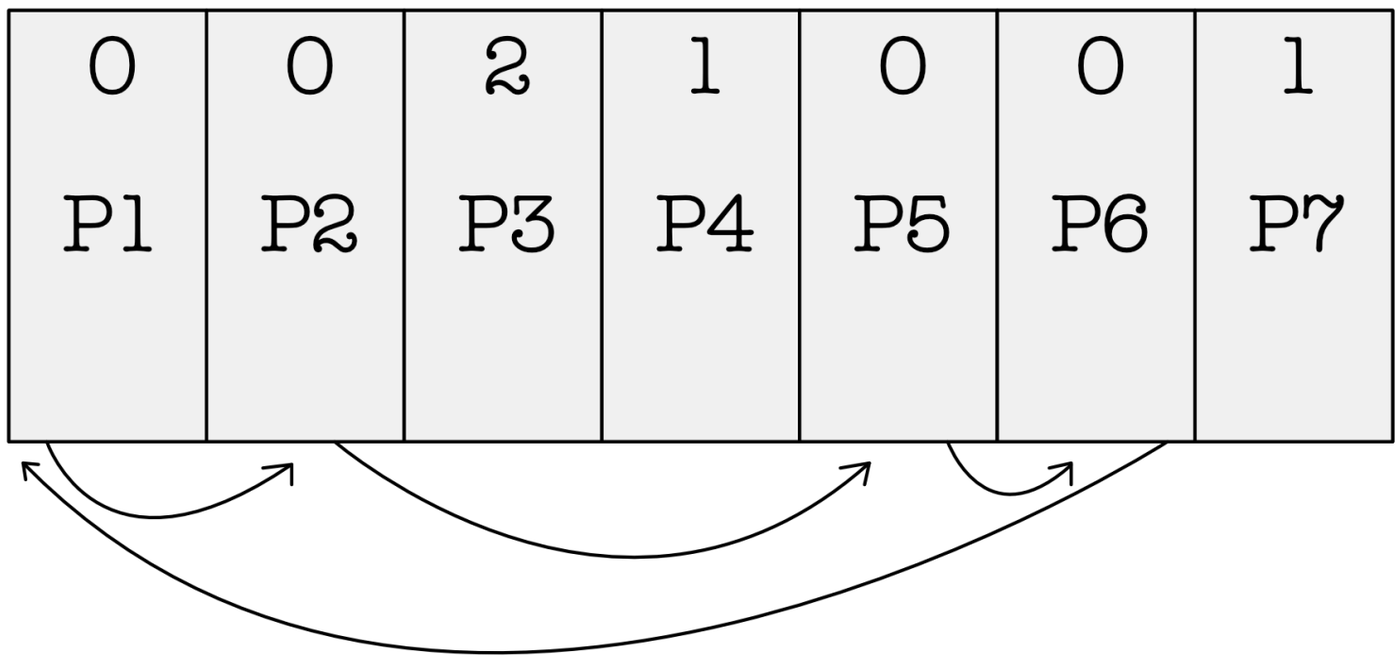
3. Managing the Storage Hierarchy

PURPOSES OF AN OS



4. Multiprogramming: process management (Scheduling & context switch)

WHY MULTIPROGRAMMING?



Context switch

MULTIPROGRAMMING

WHY

1. **Be efficient in organizing / scheduling jobs or data, since CPU can only execute 1 instruction per clock cycle**

2. **Allows timesharing: context switch so rapidly so that users still see it as interactive computing**

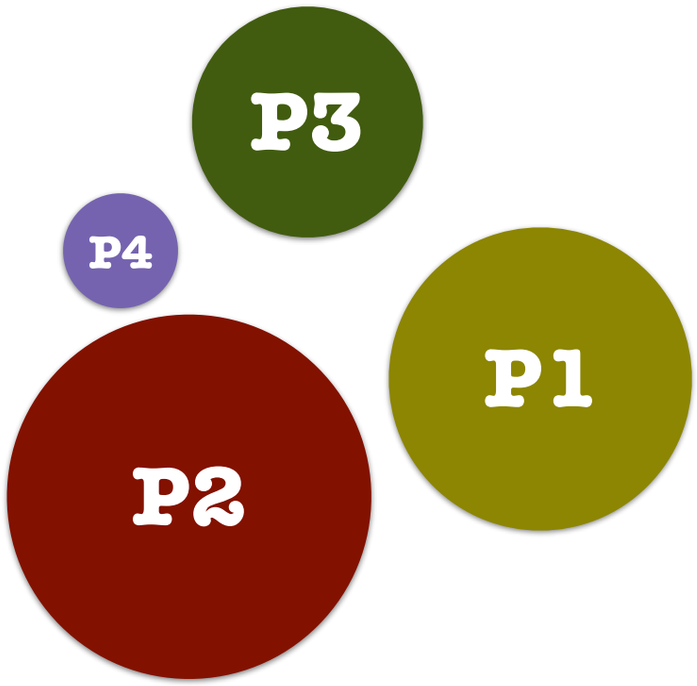
HOW

1. **Response time is fast enough**

2. **Always have at least 1 program active at any time**

3. **If RAM is full, swap with disk**

PROCESS MANAGEMENT



System has many processes, **multiplexed**

1. Create/delete user & kernel processes
2. Schedule & sync process comms
3. Manage threads
4. Deadlock handling
5. Garbage collector (free resources)

Kernel's process manager