Recap

→ What is OS?
→ Operation
→ Interrupts ⇒ "saving state"

Computer System Architecture:

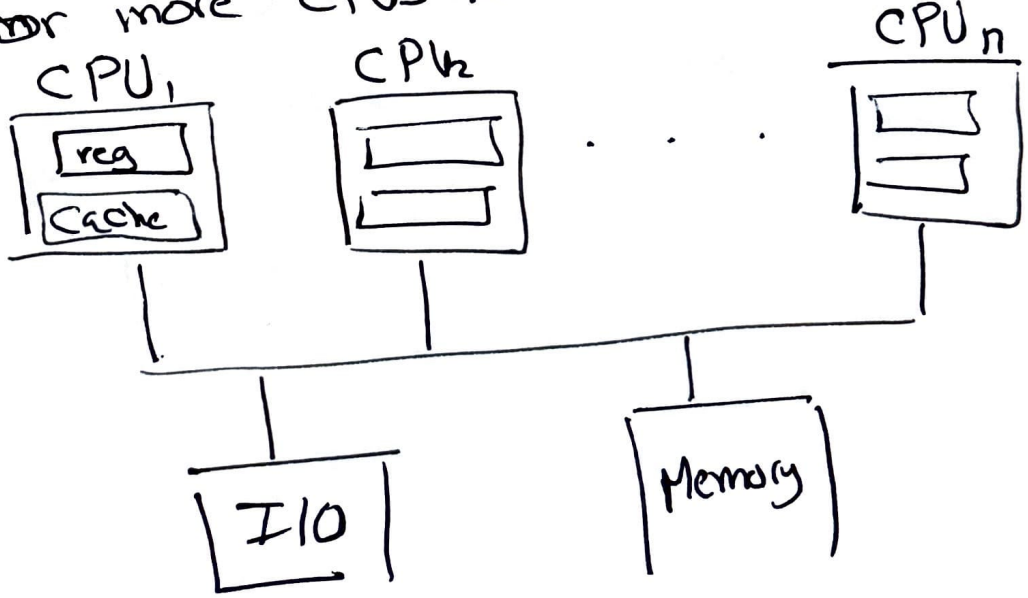
[1] Single processor System

1 CPU that executes general-purpose instructions.

Other processors may exist to help the CPU.
Donot necessarily execute user programs.

[2] Multiprocessor Systems

Two or more CPUs that execute instructions.



Why?

[1] Increase speed and throughput
more work can be done.

With N processors ⇒ speedup is less than N.

overhead in keeping correct work (sharing data).

[2] less cost than N- individual system.

share resources

[3] Increased Reliability:
Failure of one CPU doesn't halt the whole system.

Asymmetric

Each processor is responsible for Certain tasks (e.g OS).

Symmetric.

all CPU are peers.

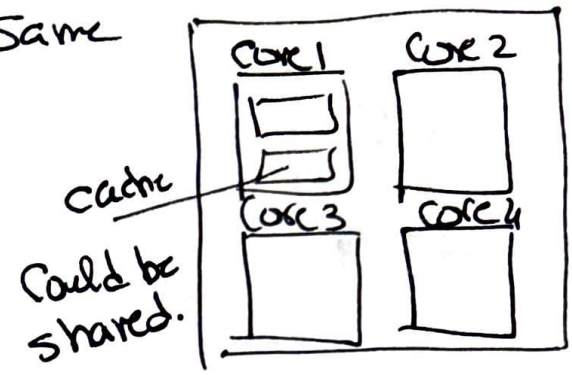
Multicore System

Place multiple cores on the same chip.

→ Communication between cores is faster

→ space

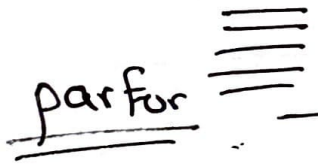
→ less power.



[3] Clustered System

Two or more individual systems connected via a fast network (Local Area Network).

HPC High Performance Computing Cluster.

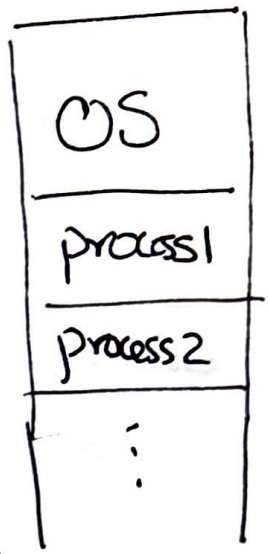


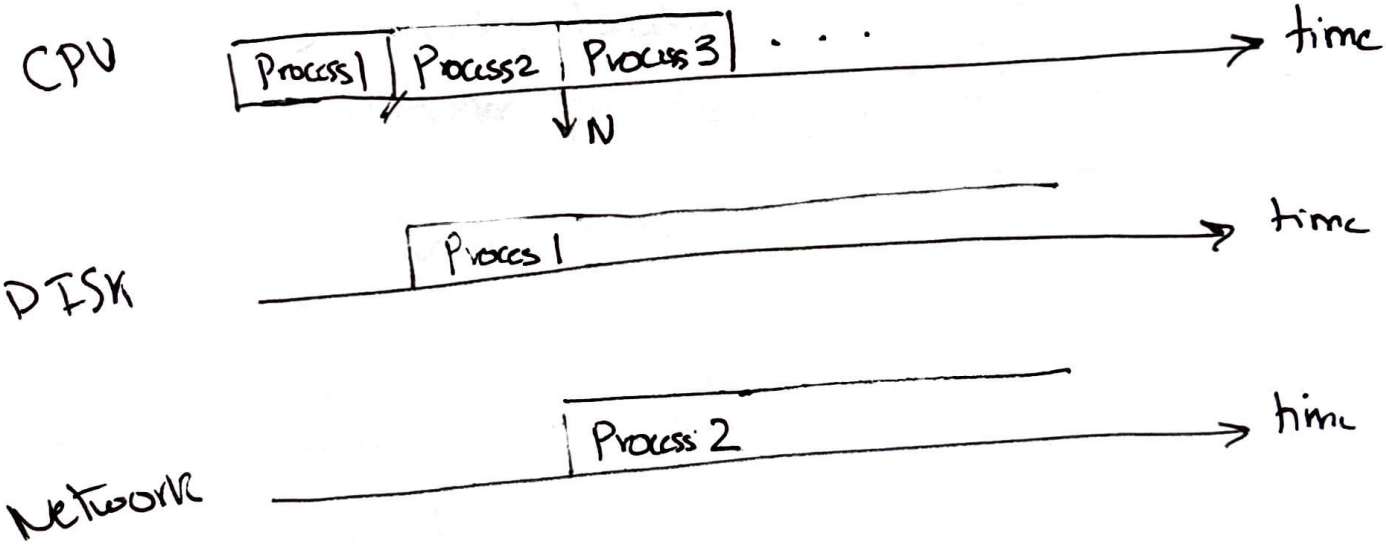
Operating System Structure

1 Multiprogramming.

A single process cannot keep all the resources busy all the time

allow more than 1 process to be in memory and switch executing between them.





2 Time Sharing

switch between processes more frequently

CPU Schedulers give each process a time slice to run before being interrupted.

Operating System Operation

OS are interrupt driven.

Since the OS shares resources with User programs protection is very important.

⇒ an error in a user program, it should not impact the OS nor other programs.

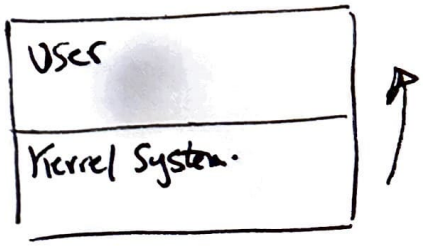
→ Infinite loop.
→ getting stuck.
→ reading location in memory that belongs to another process. (by default this should not be allowed).
→ buffer overflow.

[1] Dual mode of operation

Done with 1-bit
get set to 1 when OS executing User code.

User mode bit = 1
↓
~~Be~~ User mode bit=0

Privileged instructions that can only be executed in Kernel mode.



b Timers

Must ensure the OS maintains control over the CPU.

Timer is implemented with a counter and a clock. Counter gets decremented until it hits 0 and an interrupt occurs.

OS executes the ISR.

What functionalities the OS need to provide? [7]

Process management

Scheduling

Create / Terminate / Suspend processes.

process Communication

process Sychronization

I/O management

Supporting wide variety of devices.

read / write / aquire / release - - - -

Device drivers.

↳ DISK.

File System

protection / permissions

Indexing.

Create / delete / move / copy / - - - -

read / write

Memory Management

allocating memory / deallocating for process

tracking