

## LEC-2: Types of OS



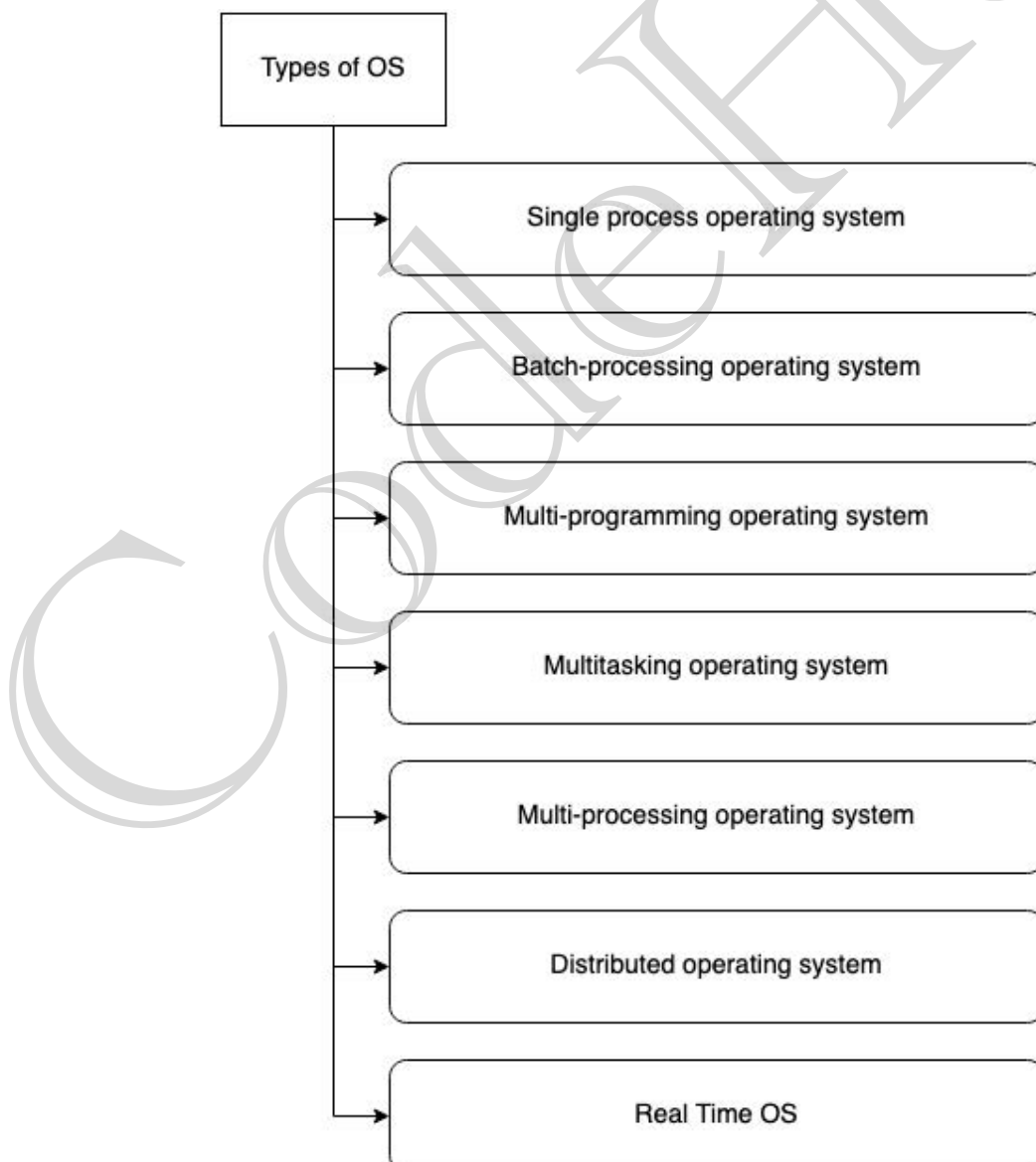
### OS goals –

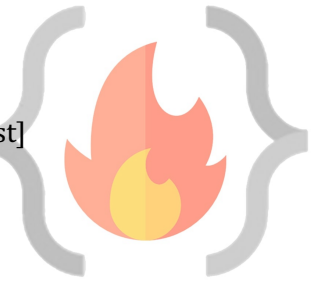
- Maximum CPU utilization
- Less process starvation
- Higher priority job execution

### Types of operating systems –

- Single process operating system
- Batch-processing operating system
- Multiprogramming operating system
- Multitasking operating system
- Multi-processing operating system
- Distributed system
- Real time OS

[MS DOS, 1981] *does one job at a time, hence Goal 2, 3 not achieved*  
[ATLAS, Manchester Univ., late 1950s – early 1960s]  
[THE, Dijkstra, early 1960s]  
[CTSS, MIT, early 1960s]  
[Windows NT]  
[LOCUS]  
[ATCS]





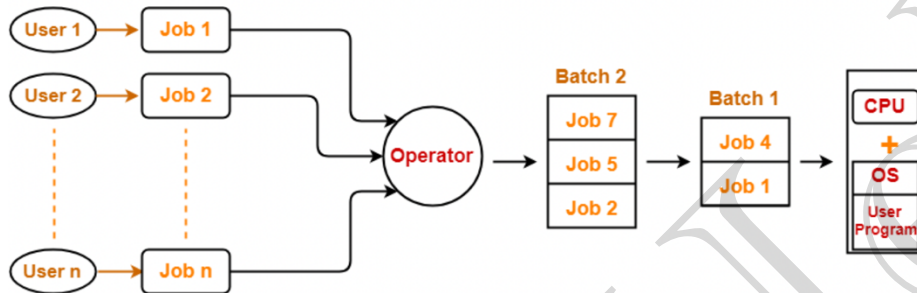
**Single process OS**, only 1 process executes at a time from the ready queue. [Oldest]

### Batch-processing OS,

1. Firstly, user prepares his job using punch cards.
2. Then, he submits the job to the computer operator.
3. Operator collects the jobs from different users and sort the jobs into batches with similar needs.
4. Then, operator submits the batches to the processor one by one.
5. All the jobs of one batch are executed together.

Goal 1, 2, 3 not achieved.

- Priorities cannot be set, if a job comes with some higher priority.
- May lead to starvation. (A batch may take more time to complete)
- CPU may become idle in case of I/O operations.



**Multiprogramming** increases CPU utilization by keeping multiple jobs (code and data) in the **memory** so that the CPU always has one to execute in case some job gets busy with I/O.

- Single CPU
- Context switching for processes.
- Switch happens when current process goes to wait state.
- CPU idle time reduced.

Meaning of Context Switching, If P1 is working, and P2 is in wait state. But now, P1 is stopped and P2 is brought into working state, then we bookmark the place where P1 is stopped by storing into PCB (Process Control Block) and P2 is taken up from PCB

**Multitasking** is a logical extension of multiprogramming.

- Single CPU
- Able to run more than one task simultaneously.
- Context switching and time sharing used.
- Increases responsiveness.
- CPU idle time is further reduced.

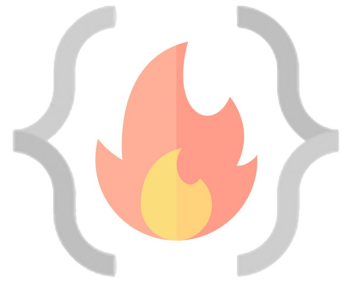
Time sharing means, after executing P1 for some time say (1000ms), then P2 starts. Again, after 1000ms, P2 stops and P3 starts.

Today's OSs are Multitasking OS

**Multi-processing OS**, more than 1 CPU in a single computer.

- Increases reliability, 1 CPU fails, other can work
- Better throughput.
- Lesser process starvation, (if 1 CPU is working on some process, other can be executed on other CPU.

**MOST  
IMPORTANT  
FOR  
INTERVIEW**



### **Distributed OS,** (Also called Loosely Coupled OS)

- OS manages many bunches of resources,  $\geq 1$  CPUs,  $\geq 1$  memory,  $\geq 1$  GPUs, etc
- **Loosely connected autonomous,** interconnected computer nodes.
- collection of independent, networked, communicating, and physically separate computational nodes.

### **RTOS**

- **Real time** error free, computations within tight-time boundaries.
- Air Traffic control system, ROBOTS etc.

Used in places, where we need 99.9% accuracy like that in ATC (Air Traffic Control), Industrial Applications (nuclear plants).

CodeHelp