DATE: / /
PAGE:

Coding Blocks - Non coding subjects Bootcamp

Webinar ① : 24th July '21 (Saturday)

What is operating system?

↳ It is a system software which helps a user to interact with hardware.

User ↔ [O.S.] ↔ [H.W.]

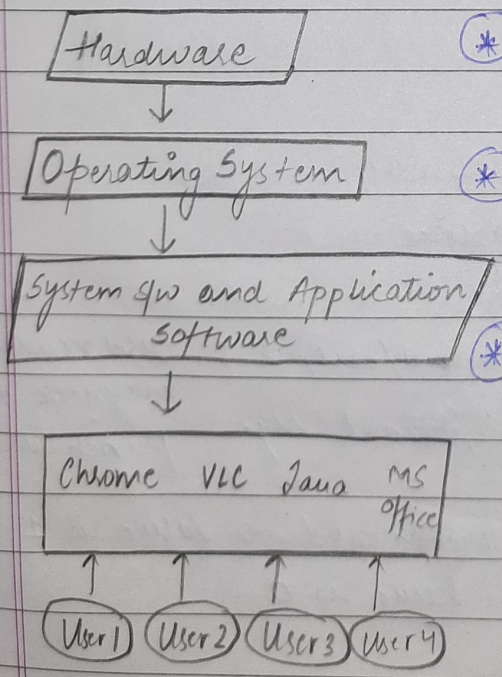
Can we work w/o an operating system?

Yes, but its gonna be very difficult cuz everything would need to be operated manually using machine level language knowledge.

What is need of an operating system?

① Making our job easier otherwise user has to do everything manually

② Manage Resources / CPU task managing



(*) First we require a laptop / computer (Hardware).

(*) Then we require an operating system such as windows / Linux

(*) Then one needs to install multiple applications and multiple users can execute applications.

Goal and functionality of OS

DATE: / /
PAGE:

Goals of OS
- Primary
  - User friendly
  - Convenience
- Secondary
  - efficiency

Operating System management system
- File management
- Memory management
- Network Security
- Process management

TYPES OF OPERATING SYSTEM

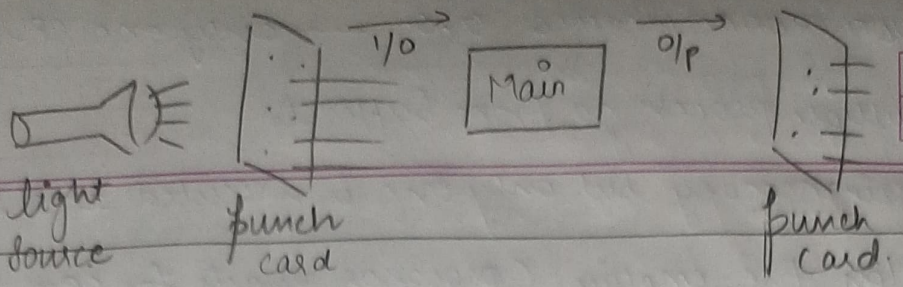
(i) Batch Operating System
(ii) Multiprogramming Operating System
(iii) Multitasking Operating System
(iv) Multiprocessing Operating System

MAIN FRAME COMPUTER

- very little memory
- Non interactive
- very slow.
- for taking input/output : Card reader, magnetic disc, punch card.
- Only binary format req.

* light passes through punch card ⇒ value is 1
otherwise ⇒ value is 0

- CPU has to wait a lot for I/O

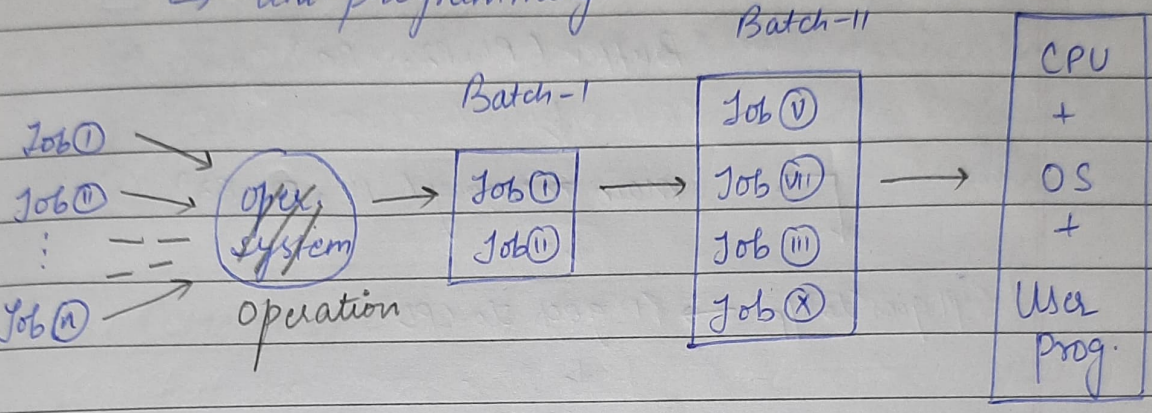


DATE: / /
PAGE:

(#) BATCH OPERATING SYSTEM

↳ Grouping similar jobs together

↳ Uni-programming



CONCEPT OF BUFFER IN O.S.

↳ Secondary Memory

↳ Fast but one program at a time

CASE - ①

User I/P → Main memory → CPU

↓

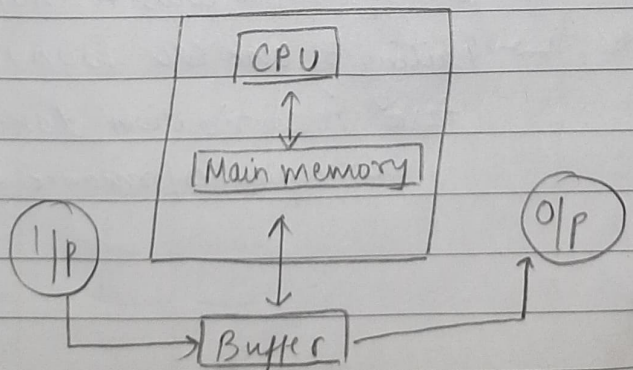
User O/P

CASE - ⑪

User I/P → Buffer → Main memory → CPU

↓

User O/P



MULTI PROGRAMMING O.S. → only one CPU

DATE: / /
PAGE:

↳ Multiple process go from buffer to main memory but only one is executed at a time

↳ Non pre-emptive (processor waits for CPU to get free)

Suppose we have a no. of processes P1, P2 .... Pn

Buffer (P1, P2 ... Pn)

↓

main memory (P1, P2, P3)

↓

P1 goes to I/p → P1 goes to CPU

↓

P1 goes for o/p, P2 goes to CPU

↓

and so on...

MULTI-TASKING O.S. → only one CPU

↳ Same as that of multiprogramming but pre-emptive

↳ Good response for each process.

↳ only one process executed at a time

MULTIPROCESSING O.S. → multiple CPU

↳ CPU(s) need to be managed

↳ if all are given equal importance → chaos.

↳ one master is chosen (randomly)

↳ Failing of one CPU won't be a problem as task redistribution takes place.

(Graceful Performance degradation)

CPU Scheduling

DATE: / /
PAGE:

* If a process gives max. time to → CPU → CPU bound process
                                  → I/O → I/O bound process

Terms related to CPU Scheduling

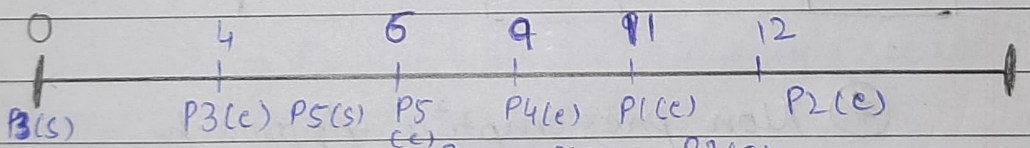
(i) Burst time : How long a process needs CPU for?
(ii) Arrival Time : Time at which a process came to M.M.
(iii) Exit time : Time at which a process leaves the M.M.
(iv) Turn around time : (Exit time − Arrival time)
(v) Waiting Time : (Turn around time − Burst time)
(vi) Response Time : First time a process was hit by CPU.

FIRST COME FIRST SERVE (Non-preemptive)

Process ID	Arrival	Burst
P1	3	2
P2	5	1
P3	0	4
P4	2	3
P5	1	2

Order of process execution
P3, P5, P4, P1, P2

Turn around time ?
Waiting time ?



At 0 sec P3 starts
At 4 sec P5, P4, P1 all are in system but P3 starts

P_id	AT	BT	ET	TAT	WT	RT
P1	3	2	11	12	10	10
P2	5	1	12	7	6	6
P3	0	4	4	4	0	0
P4	2	3	9	7	4	4
P5	1	2	6	5	3	3

Non pre-emptive (WT == RT)

What is Convoy Effect?

Some process might end up waiting forever due to long execution time of prev. processes.

What is Starvation?

↳ Biased processor (waiting due to partial CPU)

What is Deadlock?

↳ No process runs, they keep waiting for each other to execute.

SHORTEST JOB FIRST

↳ Pre-emptive ⇒ Shortest Remaining Time first

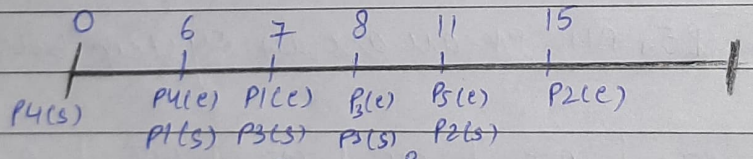
↳ Non pre-emptive

Process ID	AT	BT
P1	3	1
P2	1	4
P3	4	1
P4	0	6
P5	2	3

Calculate waiting time for ~~both~~ approaches.

non pre emptive

Non pre-emptive Approach



At 0 sec Only P4 available

At 6 sec All process are available

But Shortest job first ⇒ P1 and P3

⇒ P1 came first

and so on....

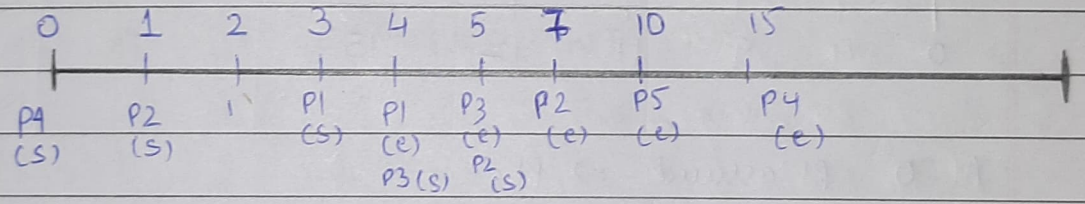
DATE: / /
PAGE:

P_ID	AT	BT	ET	TAT	WT	RT
P1	3	1	7	4	3	3
P2	1	4	15	14	10	10
P3	4	1	8	4	3	3
P4	0	6	6	6	0	0
P5	2	3	11	9	6	6

Pre-emptive Approach

Hit-AT

PID	AT	BT	ET	TAT	WT	RT
P1 x	3	x0	4	1	0	3-3=0
P2 x	1	4 3 2	7	6	2	1-1=0
P3 x	4	x0	5	1	0	4-4=0
P4	0	6 5	15	15	9	0-0=0
P5	2	3	10	8	5	7-2=5



At 0 Only P4 available (but only execute for 1 sec)

At 1 P2 comes into picture

BT(P4) > BT(P2) P2 starts

At 2 P5 comes into picture P2 continues P2 completes

BT(P5) == BT(P2)

and so on

* Gives min. waiting time
  ↳ Greedy Approach
* Used for efficiency comparison.

— END OF WEB-① —