Goals of OS:	
-> Maximum CPV utilization	V T
-> Less process startation	
-> Less process stantation -> Higher priority job escec	ution
Types of operating system!	
1) Single process operating so	extern [MS DOS 1981]
Commence has a such a commence	By the printing in the
estimate the sugar and on the state	on the following of
(P1) - (P2)	> Goes to Ilo
(?)	when it comes & executed
(3)	then Pz will start
(13)	atter wise no.
	one process at one time
	one process at one time
2> Batch Processing operating	one process at one time
	one process at one time
2) Batch Processing operating	one process at one time rejetur [ATLAS, 1950-1960]
2) Batch Processing operating (user 1) -> Tob 1) Sof	one process at one time rejetum [ATLAS, 1950-1960] Furore Batch 2 Batch 1 Toby
2) Batch Processing operating (user 1) -> Tob 1) Sof	one process at one time rejetem [ATLAS, 1950-1960]
2) Batch Processing operating (user 1) -> Tob 1) Sof	regetem [ATLAS, 1950-1960] Furne Batch 2 Batch 1 Job 7 Job 4
2) Batch Processing operating (user 1) -> [50b1] Sof	rejetem [ATLAS, 1950-1960] Furore Batch 2 Batch 1 Tob 7 Job 1 Job 2 Stack
2) Batch Processing operating (user 1) -> Tob 1) Sof	regetem [ATLAS, 1950-1960] Funce Batch 2 Batch 1 Job 5 Job 1 Job 2
2) Salch Processing operating (user 1) -> Tob 1 (user 2) -> Tob 2 (user n) Tobn	regetem [ATLAS, 1950-1960] Furre Batch 2 Batch 1 Job 5 Job 1 Job 2 Stack like funch
2) Batch Processing operating (user 1) -> [50b1] Sof	rejetem [ATLAS, 1950-1960] Fure Batch 2 Batch 1 Job 5 -> Job 1 Job 2 Stack Like funch
2) Balch Processing operating (user 1) -> Tob 1) Sof (user 2) -> Tob 2 oper (user n) Tobal Sorts H	regetem [ATLAS, 1950-1960] Furre Batch 2 Batch 1 Job 5 Job 1 Job 2 Stack like funch
2) Salch Processing operating (user 1) -> Tob 1 (user 2) -> Tob 2 (user n) Tobn	regetem [ATLAS, 1950-1960] Furre Batch 2 Batch 1 Job 5 Job 1 Job 2 Stack like funch

3) Kulti Programming Operating System [THE, Dijkstra 1960s]
His single CPO
· · · · · · · · · · · · · · · · · · ·
Goes Jo Jawill be executed (PD) Ready Queue Goes Journal of the second of the seco
goes /
J10 J2 will be executed
(PD)
the transfer of the second of
Context Switching is the process by which a CPU stafes exceeding one process or thread and starts executing another
stops executing one process or thread and starts
executing another.
It involves saving the data (content) of the currently running process & loading the saved state of the next process to run.
running process & loading the saved state of the next
process to run.
why context switching needed?
· To multitark & share CPV time along multiple processes
· To switch from user mode to kernel mode or vice-versa
· To multifack & share CPV time along multiple processes · To switch from user mode to kernel mode or vice-versa · To handle interrups & give propriety to wigent tasks.
Parage Control Blook is it to the
Process Contral Block is a data structure used by the
OS to store all the essential information about a process.
It acts as the identity cord of a process, allowing the OS to manage and track each process efficiency
te manage and track each process efficiency
(19) in Dept. 1 1 1 2
Why is PCB important?
Hi-11-6 A O SIL
Helps in conteset switching
supports process scheduling
Evalles process tracking, suspension and resumption.

Ex. When a process is paused, the OS saves its state in the PCB Later, when the process resemmes, theos retrieves info from the PCB & continues execution seamlessly.

17 Mullitasking Operating System [CTSS, MIT early 1960s] Single CPO

An OS that allows a computer to run two or more. programs simultaneously by managing CPV time and resources among them efficiently.

How it works?

- · The CPV executes part of one program, then content switches to another
- · By switching rapidly between process, it creates the illurion that multiple tasks are running at the same time

Types of con Kultitasking:

1) Porcentive Kultifasking

- · The OS decides how long each process runs · Ensures fairness & responsiveness
- 2) Cooperative Kultitasking

 - · Each process voluntarily xields control to others.

 · Zers reliable; one hang mis behaviour process whale

 system gets hang

Types of preemtive multitasking!

- 1) Robin Round (RR)
- · Each process get a fixed time slice

 · If it doesn't finish, it's presented and moved to the

 back of the queue

 · Fair & simple, but not ideal for high-priority tasks.
- 2) Priority based presentive scheduling
- · Each process is assigned a priority

 · The CPV always picks the highest-priority ready process

 · Can cause starvation of low-priority processes.
- 3) Shortest remaining time first (SRTA)
- · Extension of shartest job first, but presentive
- · Ha new process arrives with shorter remaining time than the current one, the current one is preented
- · Can be complex to implement due to need for accurate time predictions.
- 4) Kultilevel Querie Scheduling
 - · Processes are grouped ento different queue based · on priority or type · Each queue has its own scheduling algorithm.

 · Higher priority queues can present lower ones.
- 57 Multiple level Freedback Queue

 - · Avanced version of multi-level queue.
 · Allows processes to move blu queues based on age

and licharicour
· Presentive by design to prioritize interactive and
· Pouemtive by design to prioritize interactive and .
Types of Non-Preemtine
Types of Non-Preemtine
17 first come first serve (FCFS)
· Processes are executed in the order they arrive.
· Simple queue-based scheduling
· Processes are executed in the order thy arrive. · Simple queue-based scheduling · No prioritization, long jab delay short ones (convay effects)
27 Shortest job first
The process with the smallest burst time is relected next
· Obtimal in terms of average usaiting time
Optimal in terms of average waiting time Difficult to know exact burst time in advance.
To serve some some.
3> Priority Scheduling
* Each brocese is assigned Animit
· Each process is assigned a priority. · If two have the same priority, use fcfs · Starvation passible for low-priority tasks.
el-ent on land same periority, use tots
statution passible for low-priority tarks.
47 Kultilered Queul
Processes are permanently assignment to a specific
· lack queue has a scheduling algorithm (Like FIFE)
· Cach queue has a reheduling algorithm (the fifs) · No preserreption blu queues.

5) Kulti-processing operating system [windows NT]
· More than 1 CPV in a single computer. · Increases reliability, I CPV fails ather can work · Better throughput · Zesser process starvation
6) Distributed Operating System [LOCUS]
*OS manages many bunches of resources, >=1 cpvs, >=1 memory >=1 (rpvs cte Lowely connected ator autonomus, interconnected computer rocks Collection of independent, networked, communicating and physically separate computational rocks
CPU memory P3 P1 Memory CPU Memory
Memory Py P1 P2 P2 P4

[AT CS] 77 Real time operating system · Real time everar free, computations within tight time boundaries · Air traffic contral rystum, ROBOTS etc.