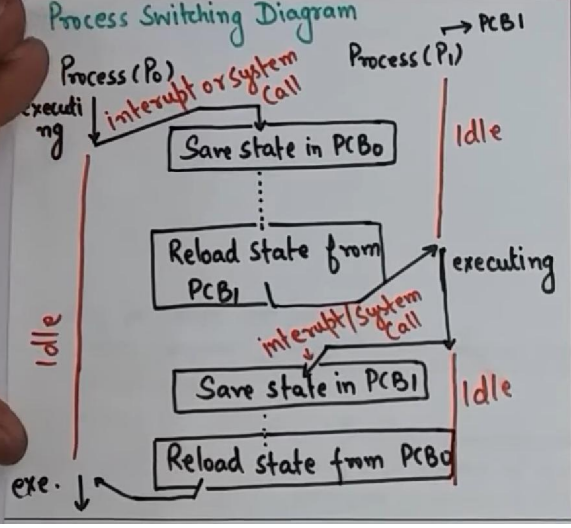
(Operating System - process - mgmt. - 3)

Operating System
Process Management
(Process Switching)

Switching from one process to another in a system requires saving state of the old process and loading saved state for new process. [Context Switching]

Process Switching Diagram



Process Scheduling: Used to switch CPU among processes.

Scheduling Queues

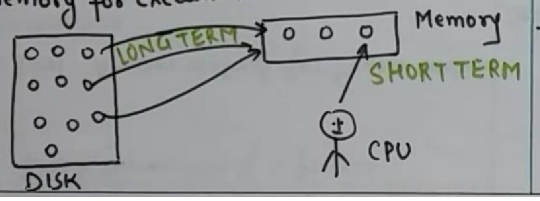
- Job Queue → Contains all the processes in the system.
- Ready Queue → Contains processes that are in main memory and ready to be executed.
- Device Queue → Contains list of processes waiting for a particular I/O device.

Are implement using linked list.

(Operating System - Process Mgmt. 3)

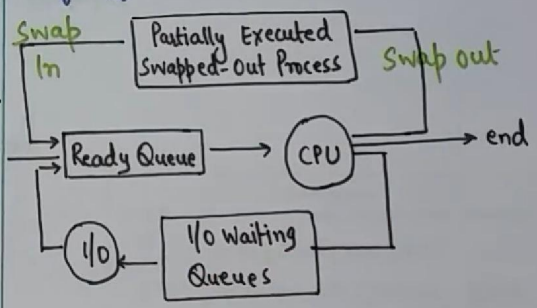
Schedulers: A process migrates among the various scheduling queues throughout it's lifetime. The O.S must select, for scheduling purpose, process from these queues. The selection process is carried out by the Scheduler.

- Long-term Scheduler (Job Scheduler): Selects processes from the pool and loads them into main memory for execution.
- Short-term Scheduler (CPU scheduler): Selects process from a pool that contains all processes that are waiting for execution.



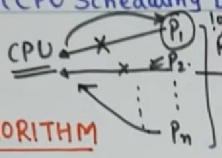
Medium Term Scheduler

Used to remove process from memory to reduce the degree of multiprogramming. Later, the process can be reintroduced into memory, and its execution can be continued where it left off.



(CPU Scheduling Lect. 1)

Operating System



CPU-SCHEDULING ALGORITHM

↳ used to select among the processes in memory that are ready to execute, and allocates the CPU to one of them.

Types of Algorithm

Pre-emptive
↳ Running to Ready
or
Waiting to Ready
CPU can be taken away from the Running Process.

Non-preemptive
↳ Running to waiting
or
Terminates
if once a process has been given CPU, then CPU can't be taken away till the Process terminates.

Scheduling Criteria

i) CPU utilization → Average fraction of time for which Processor or CPU is Busy. (0–100%)
ii) Throughput → no. of processes system can execute in a period of time. (t) ↳ Total amt. of work done. depends on Length of process
iii) Turnaround Time → Interval b/w Submission of process and Completion. P → 5 → 25 (20)
iv) Waiting Time → Average period of time a process spends waiting.
v) Response Time → Amount of time it takes when a req. was submitted Until the first response is produced.

(CPU Scheduling Lect. 2)

Scheduling Algorithms → non-preemptive.

i) First Come First Serve (FCFS) Scheduling

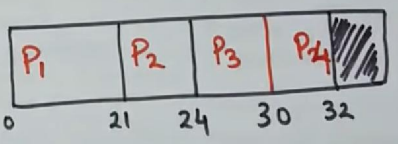
↳ i) Execution on First Come, First Serve basis.
ii) Easy implementation.
iii) High Average Waiting time.

Example. Calculate Average waiting time for the following:

Process	Burst Time (execution time)
P_1	21
P_2	3
P_3	6
P_4	2

Assume all Process Arrives at 0 time unit

Gantt chart



Average w.t = $\frac{w.t(P_1+P_2+P_3+P_4)}{4}$

w.t $P_1 = 0$
$P_2 = 21$
$P_3 = 24$
$P_4 = 30$

a.w.t = $\frac{0+21+24+30}{4}$
= 18.75 time Units.

(CPU Scheduling Lect. 3)

2.) Shortest-Job-First (SJF) Scheduling

↳ Can be implemented in both Pre-emptive and non-pre-emptive mode.

→ Minimum Waiting time.

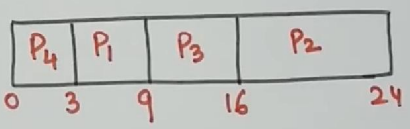
→ Pre-emptive SJF is also known as Shortest Remaining Time Algo.

Example of Non-Pre-emptive SJF

Calculate avg. waiting time.

Process	Burst time (in ms)
P_1	6 ✓
P_2	8 ✓
P_3	7 ✓
P_4	3 ✓

Gantt chart



w.t of $P_4 = 0$
w.t of $P_1 = 3$
w.t of $P_3 = 9$
w.t of $P_2 = 16$

Avg. waiting time $= \frac{0+3+9+16}{4} = \frac{28}{4} = 7$ ms.

(CPU Scheduling Lect. 3)

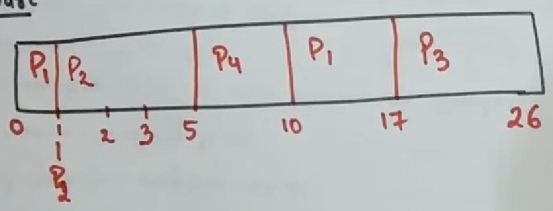
Example. of Pre-emptive SJF or SRT.

Ques) Calculate average waiting time

Process	Arrival time	Burst time (in ms)
P_1	0	8
P_2	1	4
P_3	2	9 (9)
P_4	3	5

w.t of $P_1 = 0+(10-1) = 9$
w.t of $P_2 = 0$
w.t of $P_3 = 17-2 = 15$
w.t of $P_4 = 5-3 = 2$

Gantt chart



(CPU Scheduling Lect. 5)

Round- Robin (RR) Scheduling Algorithm

i) Quantum is allocated to each process for execution. → Small amount of time. (2) Pre-emptive.

ii) Once a process is executed for a given period of time, it is preempted and other process executes for a given period of time. time Quantum.

Example. Calculate Average Waiting time for the following:-

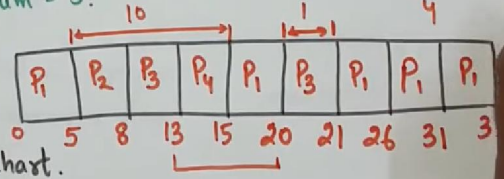
Process	Burst Time
→ P_1	21/16/11/6/1
P_2	3
→ P_3	6/1
P_4	2

Assume all process arrives at 0 and Quantum = 5.

w.t of $P_2 = 5$
w.t of $P_4 = 13$
w.t of $P_3 = 8+7 = 15$
w.t of $P_1 = 0+10+1 = 11$

Average w.t = $\frac{5+13+15+11}{4}$

$= \frac{44}{4} = 11$ m



Gantt chart.

(CPU Scheduling Lect. 4)

Priority Scheduling → Preemptive / non-preemptive.

i) Process with highest priority is executed first.

ii) Processes with same priority are executed in FCFS manner.

iii) Problem of Starvation may occur.

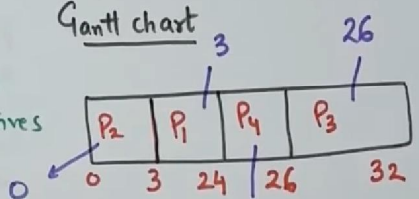
P_4 → Aging (increases priority of process)

Example. Calculate Average Waiting time for the following:-

Process	Burst Time	Priority
P_1	21	2
P_2	3	1
P_3	6	4
P_4	2	3

Assume all Process arrives at 0.

Gantt chart



24

Avg wt = $\frac{0+3+24+26}{4}$ = 13.

Easy Engineering Classes – Free YouTube Lectures

EEC Classes For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India EEC Classes

Important Numerical Questions of Operating System

Questions on Priority Based CPU Scheduling(Non-Preemptive)

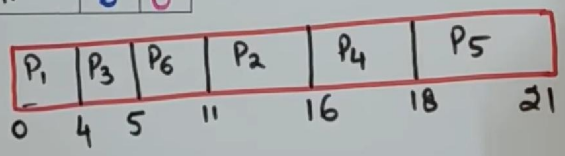
Ques1. Calculate Waiting Time and Turn Around Time

Process Number	Arrival Time	Burst Time	Priority	Completion Time	T.A.T	W.T
P1	0	4	4	4	4	0
P2	1	5	5	16	15	10
P3	2	1	7(H.P)	5	3	2
P4	3	2	2	18	15	13
P5	4	3	1(L.P)	21	17	14
P6	5	6	6	11	6	0

Average w.t = $\frac{0+10+2+13+14+0}{6}$

$= \frac{39}{6} \Rightarrow (6.5)$ Ans

Gantt chart :-



Easy Engineering Classes – Free YouTube Lectures

EEC Classes For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India EEC Classes

Important Numerical Questions of Operating System

Questions on Priority Based CPU Scheduling(Preemptive)

Ques1. Calculate Waiting Time and Turn Around Time

Process Number	Arrival Time	Burst Time	Priority	Completion Time	T.A.T	W.T
P1	1	4	4	18	17	13
P2	2	2	5	14	12	10
P3	2	3	7	10	8	5
P4	3	5	8	8	5	0
P5	3	1	5	15	12	11
P6	4	2	6	12	8	6

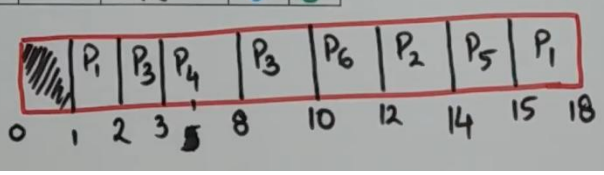
$P_1(3)$
$P_3(2)$

Average waiting time

$= \frac{13+10+5+0+11+6}{6} = \frac{45}{6}$

$= (7.5)$ Ans.

Gantt chart



Operating System Threads | Single, Multithreaded Process, benefits| Multithreading Models

(Operating System - Threads)

Operating System Threads

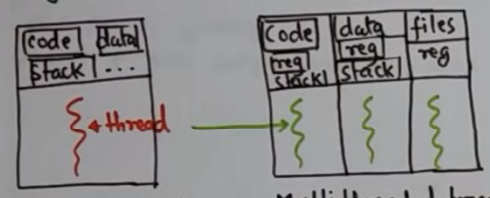
A thread is a basic unit of CPU utilization; it comprises a thread ID, program counter, a register set, and stack.

Process (Heavyweight), Threads (Lightweight)

Easy Engineering Classes – Free YouTube Coaching

For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India

Single and Multi-threaded Processes.



Single-threaded process

Multithreaded process → Can be used to run several tasks at a single point of time

Role of thread in RPC: → Remote procedure call.

threads can be used for interprocess communication.

Benefits of Multithreaded Programming:-

i) Responsiveness → Program can continue running even if some parts of its are blocked, ↑ responsiveness to user.

ii) Resource sharing → Allows an application to have several diff. threads of activity within same address space.

iii) Economy → It is economical to create threads ∵ of data and resource sharing.

iv) utilization of multiprocessor architecture → Threads can be made to run in parallel on diff. processors, thus ↑ the Concurrency.

8:33 / 11:43

(Operating System - Threads)

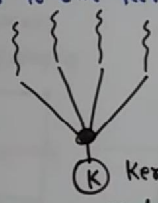
Operating System Multithreading Models.

Threads → user threads, Kernel threads } user level threads are supported above the Kernel and managed w/o Kernel support, whereas Kernel-threads are supported & managed directly by the O.S.

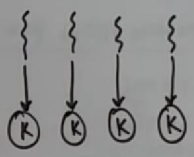
Easy Engineering Classes – Free YouTube Coaching

For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India

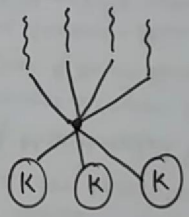
1.) Many-to-One Model: Maps many user-level threads to one Kernel thread.



2.) One-to-One Model: Maps each user thread to a Kernel thread.



3.) Many-to-Many Model: Multiplexes many user-level threads to a smaller or equal no. of Kernel threads.



(Process Synchronization Lecture.1)

Operating System - Process Synchronization

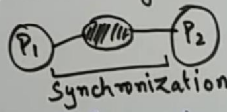
Easy Engineering Classes – Free YouTube Coaching

For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India

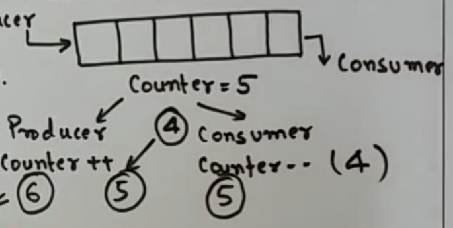
Cooperating Process - Is the process that can affect or be affected by other processes executing in the system.

P_1
P_2
P_n

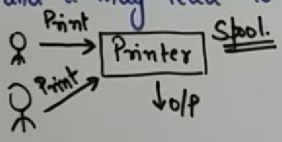
Share logical address space
or
Share data through file or message.



Producer- Consumer Problem



Problem arises in Concurrent Access to Shared data and it may lead to inconsistency.



RACE Condition

It is the situation where several processes access & manipulates some data concurrently, and the outcome of execution depends on the particular order in which access takes place.