" CPU Scheduling Algorithm "

* **Scheduling Criteria :**

1. CPU utilization
2. Throughput

} maximization

↑

optimization

↓

3. Turnaround time
4. Waiting time
5. Response time

} minimization

* **Preemptive Vs Non-Preemptive**

- Non-Preemptive ← مش أقدر أخرج العملية برا الـ CPU إلا لما تخلص أو تعمل Waiting بمزاجها.
- Preemptive ← بخرج العملية عادي قبل متخلص من الـ CPU

① Waiting Time (W.T)

الوقت اللي ال Process بتفضل مستناه لحد لما تشتغل

Start time - Arrival time

= 0 إلا لو اد اهولي في المسألة

② Turnaround Time (TAT)

الوقت الكلي من أول ما ال Process بتوصل لحد متخلص خالص

Complete time - Arrival time

execute time + Waiting time

③ Response Time (R.T)

أول مرة ظهرت فيه

First start time - Arrival time

1- First come First service (FCFS)

Process	Burst Time
P_1	24
P_2	3
P_3	3

Gantt chart

$W.T(P_1) = 0 - 0 = 0$

$(P_2) = 24 - 0 = 24$

$(P_3) = 27 - 0 = 27$

$RT(P_1) = 0 - 0 = 0$

$(P_2) = 24 - 0 = 0$

$(P_3) = 27 - 0 = 0$

$TAT(P_1) = 24 - 0 = 24$

$(P_2) = 27 - 0 = 27$

$(P_3) = 30 - 0 = 30$

2. Shortest Job First Scheduling (SJF) Algorithm:

↳ Non-preemptive & preemptive (Shortest remaining time first)

↳ CPU time (Burst time) ← بناخد الزمن الأقل فالأكثر

a) Non preemptive

Process (P)	CPU time (C.T)
P_1	3
P_2	5
P_3	2
P_4	4

Gantt chart

P_3	P_1	P_4	P_2

0 2 5 9 14

$W.T(P_1) = 2-0 = 2 = R.T(P_1)$

$W.T(P_2) = 9-0 = 9 = R.T(P_2)$

$W.T(P_3) = 0-0 = 0 = R.T(P_3)$

$W.T(P_4) = 5-0 = 5 = R.T(P_4)$

$T.T(P_1) = 5-0 = 5$

$T.T(P_2) = 14-0 = 14$

$T.T(P_3) = 2-0 = 2$

$T.T(P_4) = 9-0 = 9$

b) preemptive (Shortest remaining time first)

Process (P)	Arrival time (A.T)	CPU time
P_1	0	~~8~~ 7
P_2	1	~~4~~ 0
P_3	2	9
P_4	3	~~5~~ 0

Gantt chart

R	P_2	P_4	P_1	P_3

0 1 5 10 17 26

*لو مش وصلهم 0 بيأخد 1 مثلا هنعمل فترة من 0 لـ 1 idle وكذلك لو وقفوا

$W.t(P_1) = (0-0) + (10-1) = 9 \rightarrow R.T(P_1) = (0-0) = 0$

$W.t(P_2) = 1-1 = 0 = R.T(P_2)$

$W.t(P_3) = 17-2 = 15 = R.T(R_3)$

$W.t(P_4) = 5-3 = 2 = R.T(R_4)$

$T.T(P_1) = (1-0) + (17-1) = 1+16 = 17$

$T.T(P_2) = 5-1 = 4$

$T.T(P_3) = 26-9 = 17$

$T.T(P_4) = 10-3 = 7$

3- Priority Scheduling Algorithm :

- كل عملية مصاحب لها عدد صحيح int اسمه priority أعلى قيمة له = 1
  - Non - preemptive

Process	CPU time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

بتقارن بينها ← الـ Priority الأعلى بتاخد الرقم الأقل

- ينبدأ دايما بـ ا
- الأعلى فالأقل Priority

Gantt Chart

P_2	P_5	P_1	P_3	P_4

0 1 6 16 18 19

$W.T(P_1) = 6 - 0 = 6$

$W.T(P_2) = 0 - 0 = 0$

$W.T(P_3) = 16 - 0 = 16$

$W.T(P_4) = 18 - 0 = 18$

$W.T(P_5) = 1 - 0 = 1$

4- Round Robin Scheduling Algorithm :

- لازم يديني في المعطيات ← Quantum time (Q) ← كل العمليات ليها زمن محدد

خلصت العملية الزمن تخرج

Process	CPU time
P_1	~~3~~ 2 1 0
P_2	~~5~~ 4 3 2 1 0
P_3	~~2~~ 1 0
P_4	~~4~~ 3 2 1 0

$Q = 1$

$W.T(P_1) = (0-0) + (4-1) + (8-5) = 3 + 3 = 6$

$W.T(P_2) = (1-0) + (5-2) + (9-6) + (11-10) + (13-12)$

$= 1 + 3 + 3 + 1 + 1 = 9$

$W.T(P_3) = (2-0) + (6-3) = 2 + 3 = 5$

$W.T(P_4) = (3-0) + (7-4) + (10-8) + (12-11)$

$= 3 + 3 + 2 + 1 = 9$

Gantt Chart

P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_4	P_2	P_4	P_2

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

5- Multilevel Feedback Queue :

لـ ٣ طوابير (Queues) فوق بعض والعملية بتتخطط بينهم حسب حسب محتاجة وقت

وذا ايه : الطابور الأول (Q_0) : بشغال بنظام Round Robin (RR) ووقت 8ms

الطابور الثاني (Q_1) : بشغال برده Round Robin (RR) ووقت 16 ms

الطابور الثالث (Q_2) : بشغال بنظام FCFS (اللي يجي الأول ينفذ)

Process	Arrival time	Burst time
P_1	0	7
P_2	1	60
P_3	2	20
P_4	3	40

Gantt Chart

	0	7ms	15ms	23ms	31ms	43ms	59	67ms	83ms
Q_0	P_1	P_2	P_3	P_4					
Q_1			P_2	P_2	P_3	P_4			
Q_2					P_2	P_2	P_2	P_4	