



Example of SJF

<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
P_1	0	1
P_2	4	4
P_3	6	3
P_4	5	5

- SJF Gantt Chart

- Average waiting time =

- CPU utilization



Process Scheduling

Wait = (arrival) زمن الوصول - زمن الدخول

TAT = (arrival) زمن الوصول - زمن الخروج

CPU Utilization = $\frac{\text{Total waiting}}{\text{Total Time}} \times 100\%$

1) FCFS

من يأتي أولاً يدخل أولاً

إذا كان يوجد Arrival نبدأ بأقل زمن له إذا لم يكن موجود نلتزم بالترتيب حسب السؤال.

2) SJF

أقل عملية تدخل أولاً كاملة

3) SRTF

نأخذ الـ arrival الأقل في البداية وننقله

إلى الـ arrival الجديد ثم نقارنه (Burst) الأقل منهم يكتمل إذا دخل الـ arrival جديد نقوم بالمقارنة بين الثلاث وهكذا ولا ننسى عند الوصول إلى الـ arrival الآخر ان ننظر المدة المتبقية من (Burst) للعملية الحالية.

4) RR

نأخذ الـ arrival الأقل في البداية وندخله

حسب الـ Quantum المحدد بالسؤال

حسب الـ Quantum المحدد بالسؤال ثم ننظر هل يوجد عملية دخلت إذا تم الدخول ندخلها

حسب الـ Quantum وهكذا

5) Priority (Preemptive)

- إذا لم يكن هناك (arrival) نبدأ بـ (Priority) الأقل

ثم ندخل العملية كاملة وننتقل إلى الـ (Priority) الأعلى وهكذا

إذا كان هناك (arrival) نبدأ بـ الـ (arrival) الأقل

نقوم بتنفيذ الـ arrival الجديد إذا وجد ثم نقارنه (Priority) إذا كان الجديد

أقل ندخل العملية إلى الـ arrival الجديد وهكذا

6) Priority (Non-preemptive)

نقوم بالقاعدة السابقة ولكن نضطر إلى الـ (Priority)

الأعلى

Deadlocks-

- 1) إذا كانت الدائرة لا تحتوي على دائرة ^{مغلقة} مباشرة العوارض \rightarrow No deadlock
- 2) إذا كان هناك دائرة ^{مغلقة} \rightarrow إذا كان هناك نقطة واحدة من كل مربع يعني deadlock
- 3) إذا كان هناك أكثر نقطة تقع بالضعف هل يمكن أن نحلها
- عملية لو دُعا آخرون إذا تم يعني No deadlock

Banker's Algorithm

NEED \rightarrow

نستخرج من طريق المعادلة التالية

$$Need = Max - Allocation$$

Work \rightarrow

هو نفسه (available) ونستخرج بالمعادلة

$$Work = Total\ Number - Total\ Allocation \text{ (العدد)}$$

\rightarrow نختار الطريق إذا كان safe بتمتد
الشرط التالي

$$NEED \leq Work$$

1) إذا كان الشرط صحيح ننفذ المعادلة التالية

$$Work = \overset{\text{القيمة}}{Work} + Allocation \rightarrow \text{قيمة جديدة لها}$$

وهكذا إلى تنفيذ جميع العمليات

2) إذا كان الشرط خطأ لا ننفذ العملية وننتقل

إلى العملية الأخرى عند انتهاء كل العمليات نعد إليها

* لا تنسى تكتب المسار الصحيح بالنهاية



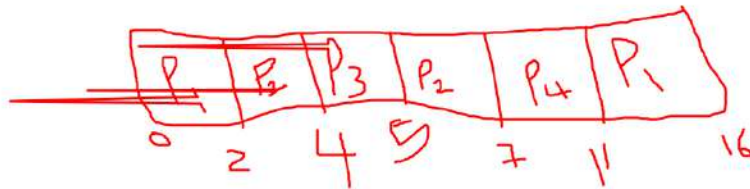
Preemptive SJN

Preemptive Shortest Job First

- Example:

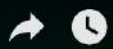
Process:	p1	p2	p3	p4
Arrival time:	0	2	4	5
Burst time:	7	4	1	4

- Schedule: ?



Process ID	Arrival Time	Burst Time
P1	0	12
P2	2	4
P3	3	6
P4	8	5

The average waiting time (in milliseconds) of the processes is _____.



Shortest Job First Scheduling (Solved)

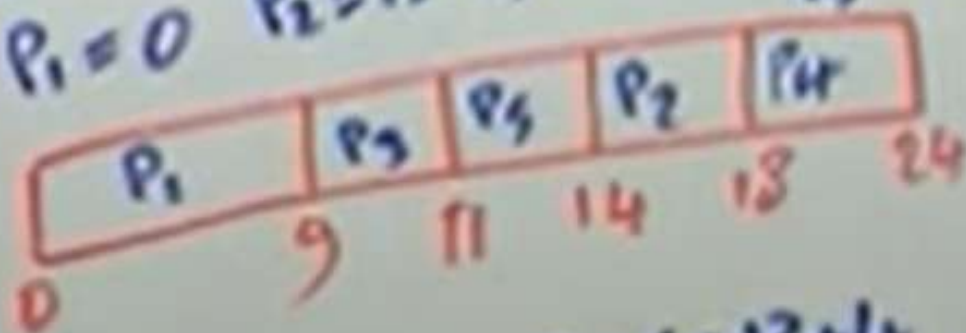
Process ID	Arrival Time	Burst Time
P1	0	10
P2	3	6
P3	7	1
P4	8	3



Ex 2: Priority (non-preemptive)

Process	Burst time	Arrival time	Priority
P ₁	9	0	3 ✓
P ₂	4	2	4 ✓
P ₃	2	3	1 ✓
P ₄	6	5	5 ✓
P ₅	3	7	2 ✓

$P_1 = 0$ $P_2 = 12$ $P_3 = 6$ $P_4 = 13$
 $P_5 = 4$



$$A.W.T = \frac{0 + 12 + 6 + 13 + 4}{5} = 7$$



Example of SJF

<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
P_1	0	1
P_2	4	4
P_3	6	3
P_4	5	5

- SJF Gantt Chart

- Average waiting time =

- CPU utilization



Case Study

Time quantum is 5ms

Process Name	CPU burst time (ms)	Priority
A	22	4
B	18	2
C	9	1
D	10	3
E	4	5

Burst time and priority for different processes

milliseconds:

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

Given processes with their Burst time

Round Robin Example:

Process	Duration	Order	Arrival Time
P1	3	1	0
P2	4	2	0
P3	3	3	0

Suppose time quantum is 1 unit.

Round Robin Example:

Process	Duration	Order	Arrival Time
P1	3	1	0
P2	4	2	0
P3	3	3	0

Suppose time quantum is 1 unit.

P1	P2	P3	P1	P2	P3	P1	P2	P3	P2
0									10

P1 waiting time : 4

The average waiting time(AWT) : $(4+6+6)/3=5.33$

P2 waiting time: 6

P3 waiting time: 6