Donia said

Ch.5 : CPU scheduling

Ch.5 → Cpu Scheduleing

* Cpu Scheduler → وظيفته :- تكون عند ال Stat rang وتعمل على ترتيب أولوية اعمالها وذهابها الى ال Cpu
Select the jop and allocates a cpu Core
↳ the Stat rang is based on the periorty on the taskes

* whan a process

1- Switches from running to waiting State → I/O or event wait, under the following four circumstances.

2- Swiches from running to ready state → interrupt

3- Swiches from waiting to ready → I/O or event Completion

4- Terminates.

→ nonpreemptive → Scheduling under 1 to 4. → ال أستطيع مقاطعة شغله

→ preemptive → 1- Consider access to shared data.
2- Consider preemptive while in Kernal mode.
3- Consider interrupts occuring during crcial OS activites.

حتى وهو يشتغل اقدر أوقف شغله

non preemptive
preemptive

* Scheduling Criteria →

- we have four Criteria :-

1- Cpu utilization
2- Throughput
3- Turnaround time
4- waiting time
5- Response time

- Cpu utilization :- خلي ال cpu يشتغل بكفاءة عالية جدا

- Throughput :- كم عدد ال process اللي بنا نفذها خلال ثانية واحدة

- Turnaround time :- excucute الوقت الذي يحتاجه ال process حتى يعمل
من يدخل لحد ما يطلع ←

Turnaround time = Completion - arrival time

- waiting time :- الوقت الذي ينتظره ال process داخل ال ready queue

waiting time = turnaround time - burst time

- Response time :- Submitted until the first response produced لحتى الوقت الذي نحتاجه

مثلاً الكيبورد لما نضغط على حرف ديطلع على الشاشة
هذا الوقت اسمه Response time

* Optimization Criteria →
  - Max cpu utilization
  - Max throughput
  - Min turnaround time
  - Min waiting time
  - Min response time

} جدولة معايير تحسين الخوارزمية.

- First-come, first-served (FCFS) scheduling :-

ex1 :-

S.N → 12

هون برتب حسب الـprocess

Process	Burst time
P_1	24
P_2	3
P_3	3

P_1	P_2	P_3
0 – 24	24 – 27	27 – 30

waiting time → $P_1 = 0$, $P_2 = 24$, $P_3 = 27$

averge waiting time : $(0+24+27) \div 3 = 17$

مجموعهم ÷ عددهم

ex2 :- S-N-13

P_2	P_3	P_1
0 – 3	3 – 6	6 – 30

waiting time → $P_2 = 0$, $P_3 = 3$, $P_1 = 6$

averge waiting time = $(0+3+6) \div 3$
$= 3$

Convoy effect :- short process behind long process
↓
وقت التأخير

• Consider one cpu-bound and many I/O bounded process.

* Shortest-Job-First (SJF) scheduling →

averge minumim waiting time ← مثالية جداً لكنها تخيلية
ممكن يسأل المستخدم متى بده يستخدم.

→ example :-

4 process → P_1 6 ← Burset time
P_2 8
P_3 7
P_4 3

هون بيرتب حسب ال Burset time ←

P_4	P_1	P_3	P_2
0 – 3	3 – 9	9 – 16	16 – 24

0 3 9 16 24

* average waiting time → $(0+3+16+9)/4 = 7$

- Determining Length of next cpu Burst →

هل بناخذ ال next cpu Burst ؟

$P_2 = 10\,ms$
cpu → 2 m/s
|
1/0
|
cpu → 8 ms

حسب التوقعات ←

$\tau_{n-1} = a\,t_n + (1-a)\,\tau_n$

(τ_{n-1} ↓) اللي انا بتوقعها — ال cpu

(τ_n ↓) القيمة المتوقعة

$\alpha,\ 0 \leq \alpha \leq 1$
↳ constant → $\frac{1}{2}$ = غالبا بتساوي

يعني في ال primitive انا قادر اوقفه كيف يعني ←

P_1 1ms , P_4 → 3ms

P_4 3ms ↑ بدينافي ↙

هادا الحكي بنحكيلو ←

زي P_1 يحتاج 1 اذا بوقف
على P_4 لانو P_1 بحتاج وقت
اقل

{ Shortest-remening-time-first

example of exponential averaging →

when $\alpha = 0$ →

$$\tau_{n+1} = \alpha T_n + [1-\alpha] \cdot \tau_n$$

zero

$$= 0 T_n + [1-0]\tau_n$$

then → $\tau_{n+1} = \tau_n$

when $\alpha = 1$ →

$$\tau_{n+1} = \alpha T_n + [1-\alpha]\tau_n$$

zero

$$= 1 \times T_n + [1-1]\tau_n$$

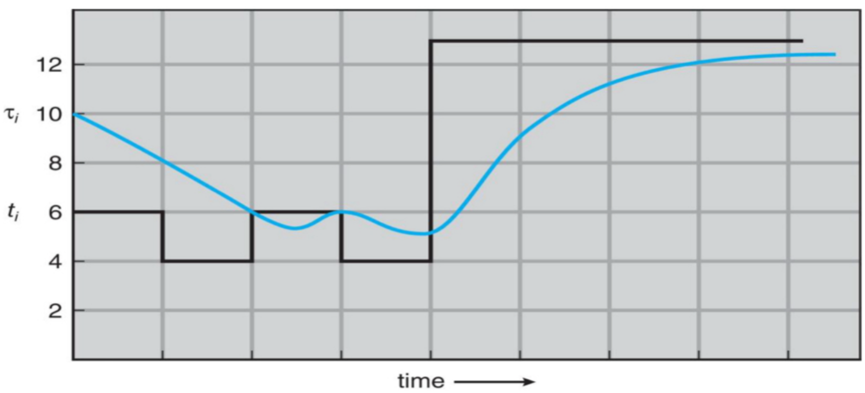
then → $\tau_{n+1} = T_n$

α and $1-\alpha$ → are less than or equal to 1.

3ms
↓
3
2
1

Prediction of the Length of the Next CPU Burst

$\alpha = \frac{1}{2}$



t_n ← CPU burst (t_i)		6	4	6	4	13	13	13	...
τ_n ← "guess" (τ_i)	10	8	6	6	5	9	11	12	...

find the $\tau_{n+1} = \alpha t_n + [1-\alpha]\tau_n$

$= \frac{1}{2} \times 6 + [1 - \frac{1}{2}] \, 10$

$= 3 + 5$

$\tau_{n+1} = 8$

---

example 8- shortest-remaining-time-first

Process	Arrival time	Burst time
P_1	0 → بيشتغل ١	~~8~~ 7
~~P_2~~	~~1~~	~~4~~ 1 ~~3~~ ~~2~~ (جاهزة 1+4=5)
P_3	← 2	9
~~P_4~~	~~3~~	~~5~~

P_1	P_2	P_4	P_1	P_3

0   1 (2, 3)   5   10   17   26

3

$2 - 1 = 1$

average time $= [(10-1) + (1-1) + (5-3) + (17-2)]/4$

$= 6.5$

P_1 كم انتظر من ا الى 10

P_2 متى وصل ومتى بدأ

*Priority Scheduling

الأولوية

ex

P_1	1	→	①
P_2	3	→	③
P_3	2	→	②

} → حسب الأولوية

(Smallest integer = higest proirity)

يعني رقم 1 هو أولى أهمية أو أولوية

ملاحظة مهمة :-

Largest integer = higest priority ← ممكن حسب السؤال - يجيب

بجيب العكس 5 اهم من 4

this type is →

- preempitive
- nonprempitive

stravation → Low priority process may never excute .

Aging → as time progresses increase the priority of the process.

example :-

Example of Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

P_2	P_5	P_1	P_3	P_4
0–1	1–6	6–16	16–18	18–19

- Average waiting time = 8.2

P_2	P_5	P_1	P_3	P_4

0 1 2 6 16 18 19

18 + 1 = 19

1+5 = 6 / 10+6 = 16 16+2 = 18

averge waiting time = (6+1+16+18) / 5

= 8.2

4S : 26

Round Robin (RR)

for example :

P_1	P_2	P_3
~~10~~ ~~8~~ 6	~~5~~ 3	~~5~~ 3

q = 2ms

↓

كل واحد فيهم الprocess رح يشتغل 2ms وبعدين الي بعده

time quantm = q

10 < q < 100 milisec

P_1	P_2	P_3	P_1	P_2	P_3	P_1	P_1

2 4 6 8 10 12 14

عم خلص التايم تاعو فبصير العملية preempted « مقاطعة »

العلاقة بين الprocess ← queue

* اذا كان عندي n process « يعني مش محدد عددهم » كل واحد منهم يأخذ $\frac{1}{3}$ الوقت تاع ال cpu.

[عندي q وعندي n] مستحيل ال process يوخذ وقت اكثر من (n-1)q time

example : n = 3, q = 2 → (n-1)q = (3-1) x 2
= 2x2
= 4ms

• Timer interrupts every quantum to schedule next process.

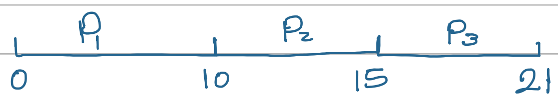
if the q is very long ⇝ ex :- q = 100 ⇝ it well work on the princible FIFO

⟶ first in First out

* example :-

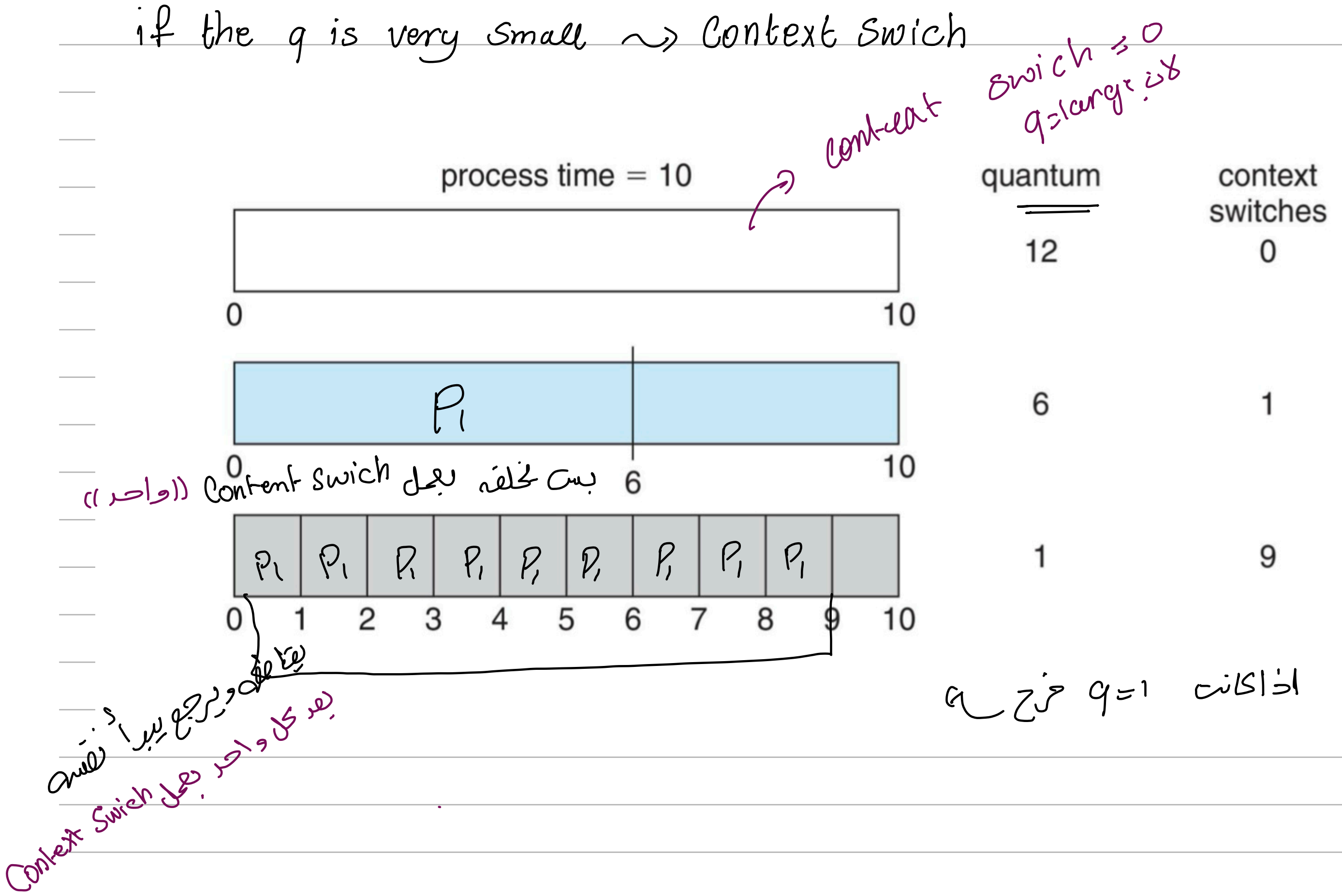
	P_1	P_2	P_3
	10	5	6

q = 10



هاي المشكلة كاني ما عملت اشي لما اعتبرت q كبيرة

if the q is very small ⇝ Context Swich



اذا كانت q=1 خرج q

* example of RR with Time Quantum = 4

P_1 ⟶ 24

P_2 ⟶ 3

P_3 ⟶ 3

P1	P2	P3	P1	P1	P1	P1
0–4	4–7	7–10	10–14	14–16	16–20	20–24

Conext swich

- turnaroundtime :- الوقت التي أدخل فيه مع وقت الانتظار ووقت انهاء الشغل لحد ما تطلع

notes :- 80% of cpu bursts should be shorter than q

multi level quen

- **Multi Level queue scheduling →**

System process → أي برنامج له علاقة بالنظام أضيفه هنا

interactive process ← interactive الـ process إلي الورقة ب الـ

في كل process حسب شغله أضيفه في بيور مستقل

* how it works :-

① كل process عندي داخل بيور معين بيترتب بطريقة معينة

② كل برنامج بالجلس بيشتغل دائماً داخل النظام تاعو

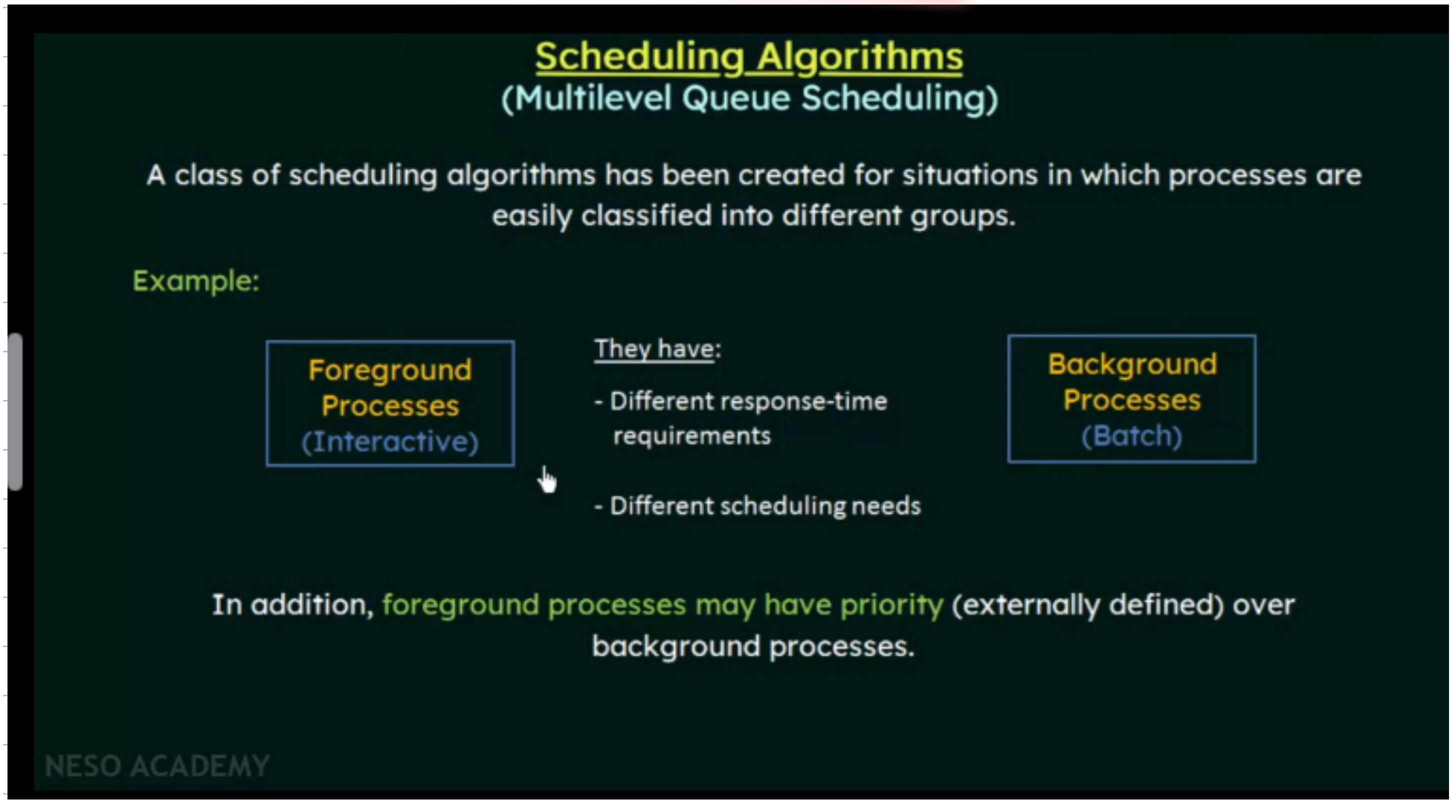
③ كل queue بنشغلها حسب الـ Time slice لـ cpu time الـ وموزع تعدى الوقت المسموح.

④ مرتب من الـ highest priority لـ lowest priority.

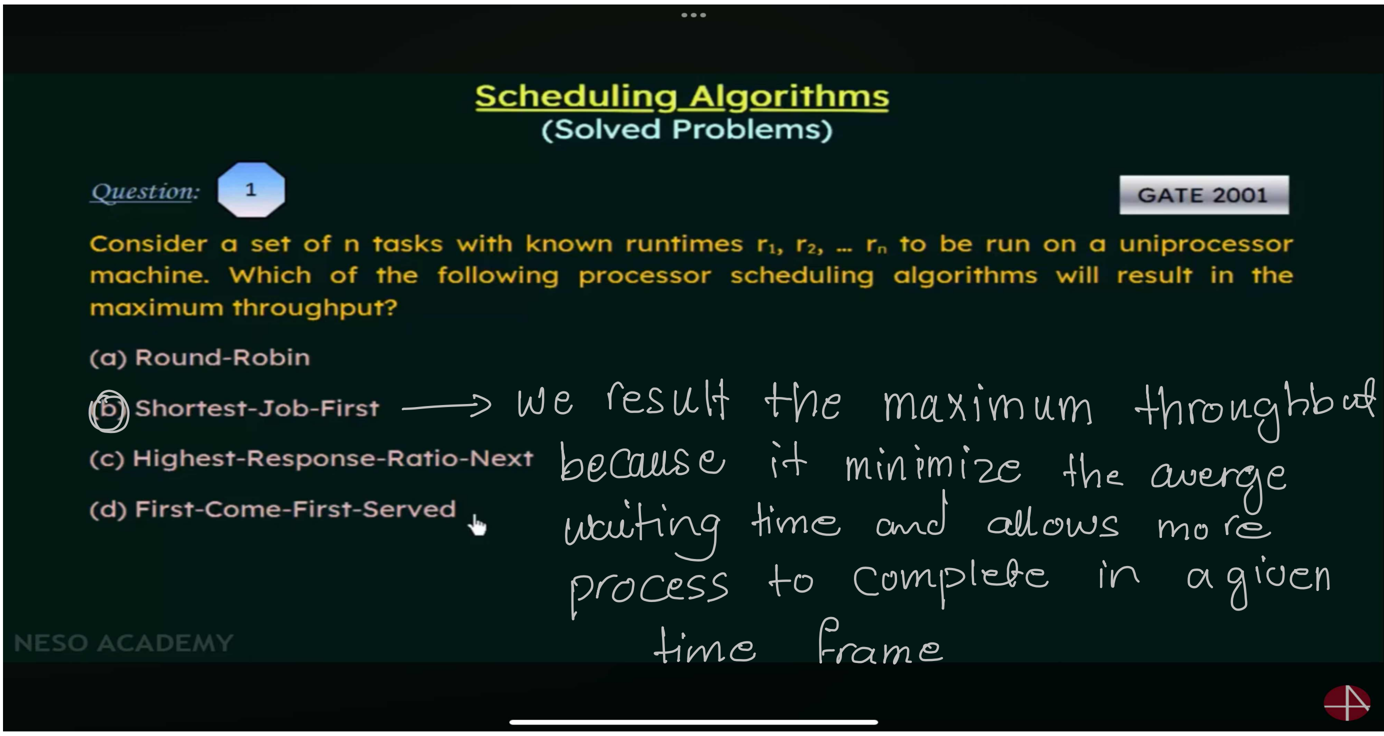
Multi feedback Queue →

Scheduling Algorithms
(multi-level Queue Scheduling)

A class of scheduling algorithms has been created for situations in which process are easily classified into different groups.



-Questions



(b) Shortest-Job-First → we result the maximum throughput because it minimize the averge waiting time and allows more process to complete in a given time frame

Question: 2

GATE 2002

Which of the following scheduling algorithms is non-preemptive?

(a) Round-Robin preemptive

(b) First In First Out → non → only

(c) Multilevel Queue Scheduling preemptive or non

(d) Multilevel Queue Scheduling with Feedback preemptive and non

NESO ACADEMY

Question: 3

GATE 2010

Which of the following statements are true?

I. Shortest remaining time first scheduling may cause starvation

II. Preemptive scheduling may cause starvation

III. Round Robin is better than FCFS in terms of response time

(a) I only

(b) I and III only

(c) II and III only

(d) I, II and III

NESO ACADEMY

Question: 4

GATE 2012

Consider the 3 processes, P1, P2 and P3 shown in the table.

Process ID	Arrival Time	Time Units Required
P1	0	5
P2	1	7
P3	3	4

The completion order of the 3 processes under the policies FCFS and RR2 (round robin scheduling with CPU quantum of 2 time units) are:

(a) FCFS: P1, P2, P3
RR2: P1, P2, P3

(b) FCFS: P1, P3, P2
RR2: P1, P3, P2

(c) FCFS: P1, P2, P3
RR2: P1, P3, P2

(d) FCFS: P1, P3, P2
RR2: P1, P2, P3

q = 2

FCFS P1 P2 P3

RR2

P1
0 2

P1 → ~~5~~ ~~3~~ 1 5-2=3

P2 → ~~7~~ ~~5~~ 3

P3 → ~~4~~ 2

P1	P2	P3	P1	P2	P3	P2
0–2	2–4	4–6	6–8	8–10	10–12	12–14

Question: 5

GATE 2013

A scheduling algorithm assigns priority proportional to the waiting time of a process. Every process starts with priority zero (the lowest priority). The scheduler re-evaluates the process priorities every T time units and decides the next process to schedule. Which one of the following is TRUE if the processes have no I/O operations and all arrive at time zero?

$P_0 \rightarrow L$

(a) This algorithm is equivalent to the first-come-first-serve algorithm.

(b) This algorithm is equivalent to the round-robin algorithm.

(c) This algorithm is equivalent to the shortest-job-first algorithm.

(d) This algorithm is equivalent to the shortest-remaining-time-first algorithm.

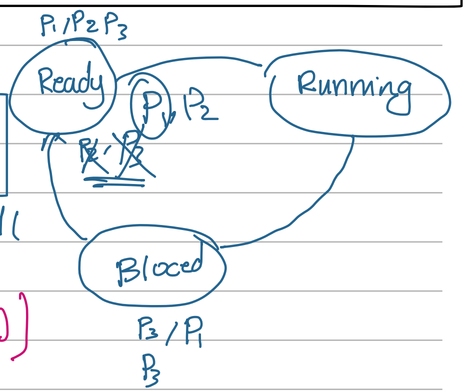
NESO ACADEMY

example : three process, $P_1 P_2 P_3$ find avg. WT, TT, & RT if system follows Sjf scheduling →

process	process time	AT	Cpu Burst(1)	I/O Burst	Cpu time
P_1	20	0	6	10	4
P_2	30	0	9	15	6
P_3	10	0	3	5	2

P_3	P_1	P_3	P_2	P_1	—	P_2

0 3 8 9 11 20 24 35 41



14 6 26

- waiting time → $((24-9)+(35-20)+(9-9))$

$P_3 \rightarrow 8$

$P_1 \rightarrow 19$

$P_2 \rightarrow 35$

P1 P2 P3

- T-time → 24 + 41 + 11

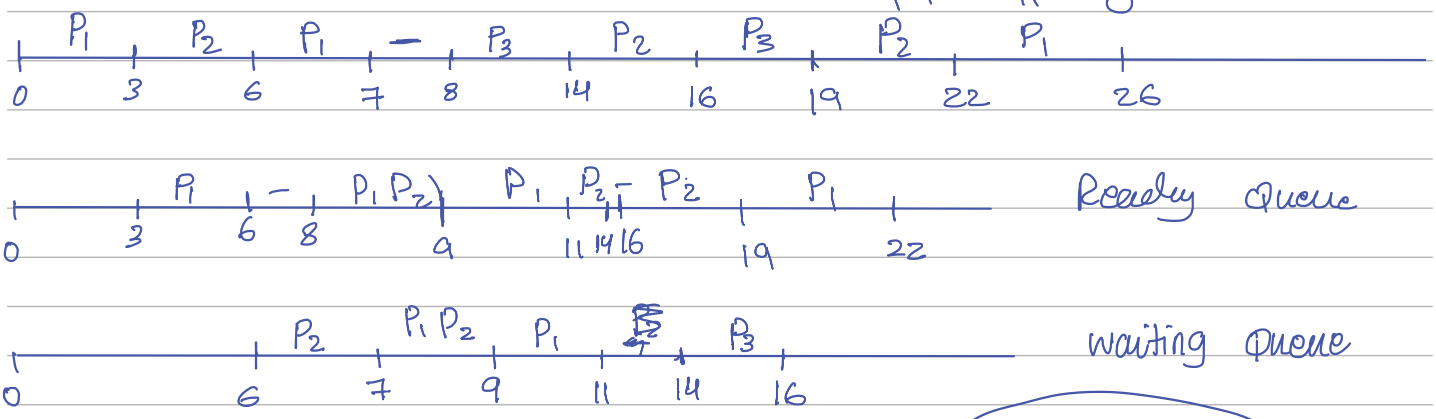
R-time → 3, 11, 0

A computer system has 3 processes. Each process initially has a CPU burst, then an I/O burst, and then another CPU burst, as shown in this table:

Process	Priority	Arrival Time	1st CPU Burst	I/O Burst	2nd CPU Burst
P1	2	0	~~4~~ 1	4	4
P2	1	3	~~3~~	3	~~5~~ 3
P3	0	8	6	2	3

preemptive priority

CT	TT	WT
26	26	14
22	19	8
19	11	0



$P_1 \rightarrow (6-3) + (22-11) = 14$

$P_2 = (14-9) + (19-16) = 8$

$P_3 = 0$

AV. WT = 7.33

A computer system has 3 processes. Each process initially has a CPU burst, then an I/O burst, and then another CPU burst, as shown in this table:

Process	Priority	Arrival Time	1st CPU Burst	I/O Burst	2nd CPU Burst
P1	2	0	4	3	4
P2	1	3	3	3	5
P3	0	8	6	2	3

what is the average waiting time given SRTF scheduling?

A computer system has 3 processes. Each process initially has a CPU burst, then an I/O burst, and then another CPU burst, as shown in this table:

Process	Priority	Arrival Time	1st CPU Burst	I/O Burst	2nd CPU Burst
~~P1~~	2	0	4	3	4
P2	1	3	3	3	5
P3	0	8	6	2	3

what is the average waiting time given SJF scheduling?

CT	TT	WT
11	11	0
16	13	2
27	19	8

P1 | P2 | P1 | P2 | P3 | — | P3
0, 4, 7, 11, 16, 22, 24, 27

P2 | P2 | P3 | — | P3
0, 3, 4, 10, 11, 16, 22, 24

Ready queue

P1 | P2 | — | P3
0, 4, 7, 10, 22, 24

waiting queue

WT = $P_1 = 0$

$P_2 = (4-3) + (11-10) = 2$

$P_3 = (16-8) = 8$

avg WT = 3.33

A computer system has 3 processes. Each process initially has a CPU burst, then an I/O burst, and then another CPU burst, as shown in this table:

Process	Priority	Arrival Time	1st CPU Burst	I/O Burst	2nd CPU Burst
P1	2	0	4	3	4
P2	1	3	3	3	5
P3	0	8	6	2	3

what is the average waiting time given Round Robin scheduling with q=2 ?

CT	TT	
13	13	2
22	19	8
26	18	7

5.66

P1 | P1 | P2 | P2 | P1 | P3 | P1 | P2 | P3 | P2 | P3 | P2 | — | P3
0, 2, 4, 6, 7, 9, 11, 13, 15, 17, 19, 21, 22, 23, 26

P1 | — | P2 | — | P3 | — | P2 | P3 | P2 | — | P2 | P3
0, 2, 4, 6, 7, 8, 9, 10, 13, 15, 17, 19, 21, 23

P1 | P2 | — | P3
0, 4, 7, 10, 21, 23

waiting

avg WT = $P_1 = (4-2) = 2$

$P_2 = (1+3+2+2) = 8$

$P_3 = 1+4+2 = 7$

$\frac{2+8+7}{3} = \frac{17}{3} = 5.67$

7-6=1
17-15=2
21-19=2

P_3 = 9-8=1
15-11=4
23-21=2