الباب الثالث

جدولة العمليات

المجدول : Short-term scheduler

تمارين المحاضرة

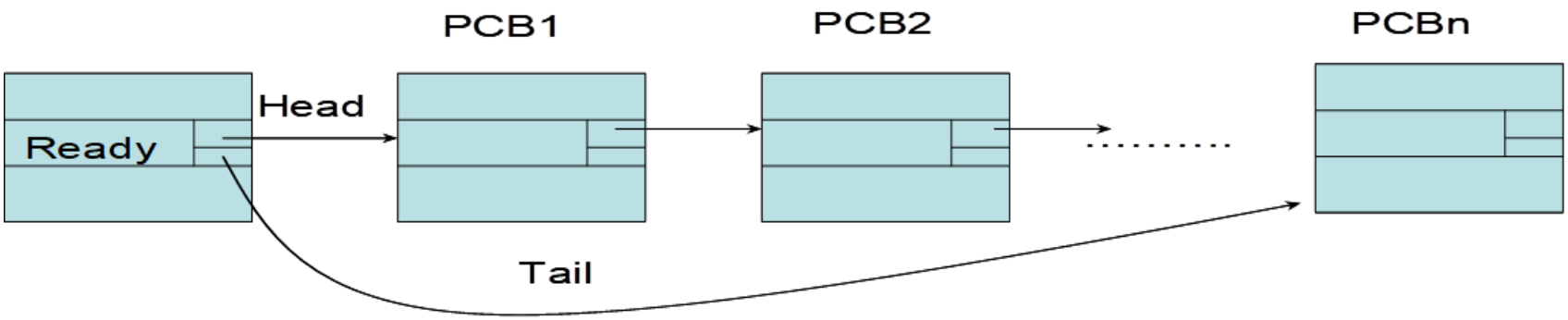
خوارزميات الجدولة (Scheduling Algorithms)

1- خوارزمية الرتل او الطابور (First-Come, First-Served FCFS)

في هذه الخوارزمية يتم تخصيص المعالج لأول عملية موجودة بالطابور اي بمعنى حسب وصولها لقائمة الجاهزية (Ready state) – الاول فالأول. تصنف هذه الجدولة بالـ **Nonpreemptive scheduling**.

حيث يتم تخزين تركيبة العملية (PCB) في قائمة الجاهزية على هيئة سلسلة (Linked List) بحيث تخزن العملية القادمة في ذيل السلسلة (Tail) والعملية الاولى على رأس القائمة (Head) كما هو موضح بالشكل الاتي:

متوسط زمن الانتظار عادة طويل.



First-Come, First-Served FCFS خوارزمية الرتل او الطابور

مثال : المعلومات الاتية تمثل مجموعة من العمليات **وصلوا بالترتيب** التالي ويتطلبوا زمن تنفيذ (CPU-Burst) موضح بالجدول الاتي

Process	Burst Time
P1	24ms
P2	3ms
P3	3ms

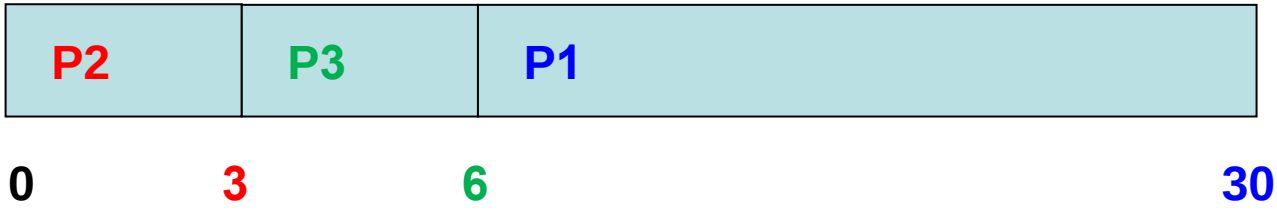
اذا تم وصولهم حسب التسلسل الاتي **P1, P2, P3** فان

- متوسط زمن الانتظار = (27+24+0) / 3 = **ms17**
- متوسط الزمن الكلي (**Turnaround time**) = (24+27+30) / 3 = **ms27**

خوارزمية الرتل او الطابور First-Come, First-Served FCFS

واذا كان الوصول حسب الترتيب P1 ,P3 ,P2 فان

Process	Burst Time
P2	3 ms
P3	3 ms
P1	24 ms



- متوسط زمن الانتظار= (6+3+0) / 3 = 3 ms،
- ومتوسط الزمن الكلي = (3+6+30) / 3 =13ms

* نلاحظ ان زمن الانتظار يتغير بحسب ترتيب وصول العمليات وخاصة عندما يكون زمن التنفيذ متباعد بين مختلف العمليات.

2- جدولة الشغل الاقصر اولا (Shortest-Job-First (SJF) Scheduling)

تنقسم الخوارزمية إلى نوعين هما :

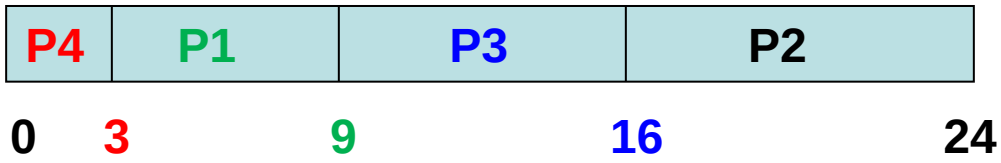
- **غير قابلة للتوقف Non-preemptive** إذا بدأت عملية ما التنفيذ داخل المعالج لن تتوقف اي عدم انتقال المعالج لأي عملية اخرى الا اذا انتهت العملية الحالية، بحيث أن **العملية التي تحتاج إلى زمن اقل يتم تنفيذها أولا.** هذه الجدولة تعطي زمن انتظار افضل مقارنة بجدولة الرتل او الطابور، أيضاً **هذه الجدولة تختار الشغل الاقصر المراد تقديم الخدمة له بشكل يضمن أن هذا الشغل سوف ينتهي و يترك النظام في اسرع وقت ممكن**، مما يؤدي إلى تخفيض عدد العمليات المنتظرة و بالتالي تخفيض زمن الانتظار العام.

جدولة الشغل الاقصر اولا (Shortest-Job-First (SJF) Scheduling)

مثال (**غير قابلة للتوقف**) : مجموعة من العمليات بزمن سلسلة المعالج (CPU-Burst) مقدر بالملي ثانية كالاتي:

Process	Burst Time
P1	6ms
P2	8ms
P3	7ms
P4	3ms

باستخدام **جدولة الشغل الاقصر اولا** (**SJF**) فان العمليات تخزن حسب الشكل الاتي:



وعليه يكون :

- **متوسط زمن الانتظار** = (0+3+9+16) / 4 = **7ms.**
- **ومتوسط الزمن الكلي** = (0+3+9+16+24) / 4 = **13ms**

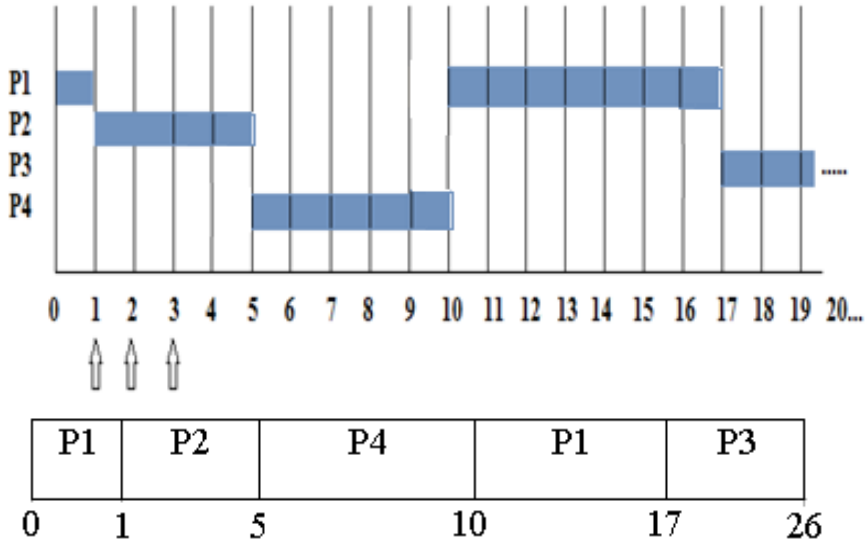
جدولة الشغل الاقصر اولا (Shortest-Job-First (SJF) Scheduling

تنقسم الخوارزمية إلى نوعين هما :

- **قابلة للتوقف Preemptive** إذا وصلت عملية جديدة إلى صف الانتظار وكان زمنها أقصر من الزمن المتبقي للعملية التي يتم تنفذها حاليا بالمعالج سيقوم المجدول بإخراج الأولى وإدخال التي وصلت حديثا.

مثال (قابلة للتوقف) : اذا كان وصول العمليات الاتية في زمن مختلف ويحتاجوا الى زمن تنفيذ كما هو موضح بالجدول الاتي:

العملية	زمن الوصول	الزمن
P1	0	8
P2	1	4
P3	2	9
P4	3	5



الحل

سيكون ترتيب العمليات داخل المعالج كما يلي:

زمن انتظار العملية = زمن الانتظار الكلي – زمن الوصول وعليه فان :

زمن انتظار العملية P1 = 10 -1 = 9

زمن انتظار العملية P2 = 1 - 1 = 0

زمن انتظار العملية P3 = 17 -2 = 15

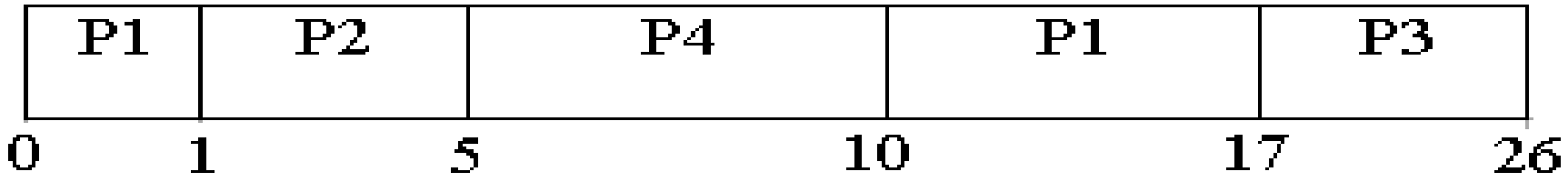
زمن انتظار العملية P4 = 5 - 3 = 2

- **ملاحظة: عند تساوي الزمن للعمليات تستخدم خوارزمية الرتل (FCFS).**

متوسط زمن الانتظار = (9+0+15+2) / 4 = **6.5 م/ث**

متوسط الزمن المستغرق (الكلي) = [(0-17)+(1-5)+(2-26)+(3-10)]/4

= [17+4+24+7]/4 = 52/4 = **13م/ث**



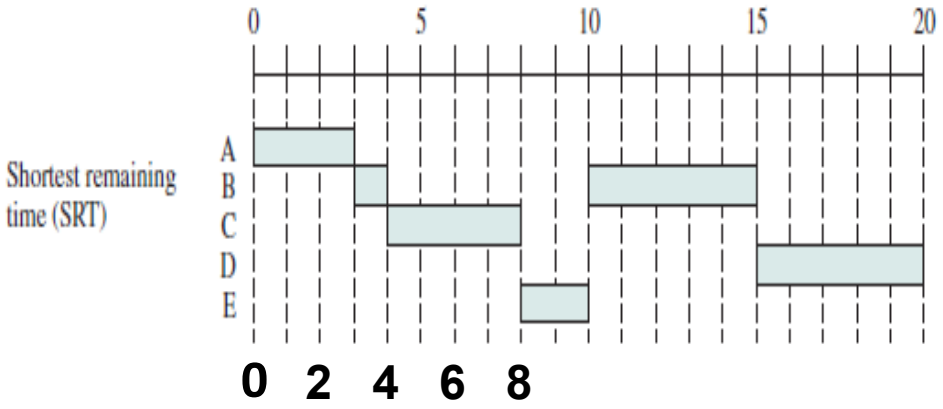
Process	Arrival Time	CPU Burst	Start Time	Waiting Time = Start Time - Arrival Time	Finish Time	Turnaround time= Finish Time - Arrival Time
P1	0	8	**10**	10-1= **9**	**17**	17-0= **17**
P2	1	4	**1**	1-1= **0**	**5**	5-1= **4**
P3	2	9	**17**	17-2= **15**	**26**	26-2= **24**
P4	3	5	**5**	5-3= **2**	**10**	10-3= **7**
Wt	**9+0+15+2/4 = 6.5**				**Tt**	**17+4+24+7/4 = 13**

3- جدولة الزمن المتبقي الاقصر (Shortest Remaining Time – SRT)

وهي **نسخة الاخلاء القسري من جدولة الشغل الاقصر اولا** وذلك بالاعتماد على تسجيل الزمن المنقضي بعد كل عملية تنفيذ، وهي ايضا تفترض **معرفة زمن التنفيذ عند انشاء العملية.**

حيث يقارن الزمن المتبقي للعملية **X** بالزمن المتبقي للعملية **Y قيد التنفيذ** – فاذا كان الزمن المتبقي للعملية **X** اقل من الزمن المتبقي للعملية **Y قيد التنفيذ** فانه يتم استبدال العملية **قيد التنفيذ** بالعملية **X** أي الاقل زمنا متبقيا، **لان هذه الجدولة ذات اسلوب الاخلاء القسري.**

مثال – اذا تم وصول العمليات **A, B, C, D, E** في زمن مختلف ويحتاجوا الى **زمن تنفيذ** كما موضح بالجدول التالي :

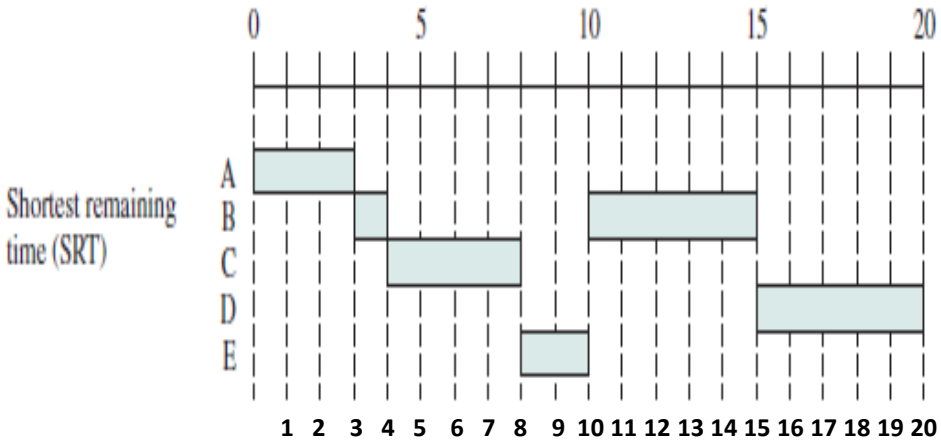


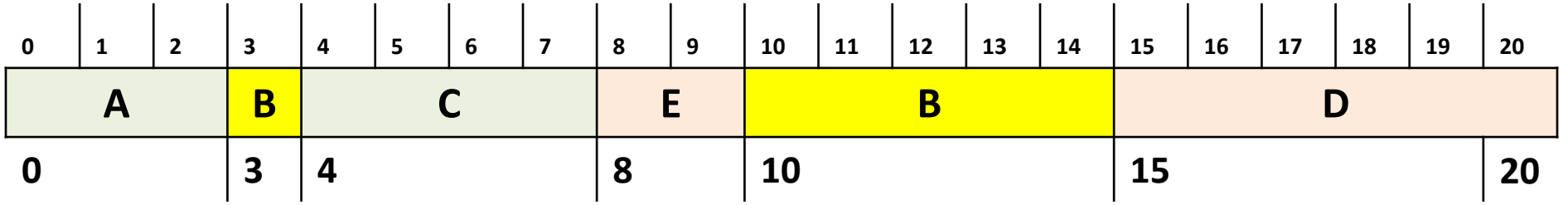
العمليات	زمن الوصول	زمن التنفيذ
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

- هنا العملية **A** يتم تنفذها الى النهاية لان زمن وصول العمليات الاخرى كان متأخرا (انظر الجدول السابق).
- وصلت العملية **B** عند **الثانية 2** الا انها تنتظر لحين اتمام العملية **A** لأنه عند وصول العملية **B** تكون العملية **A** هي الاقصر زمنا متبقيا.
- في **الثانية الرابعة** تصل العملية **C** وهي الاقصر زمنا في هذه اللحظة تعطى الفرصة للتنفيذ لأنها الاقصر زمنا متبقيا ويبقى الزمن المتبقي من العملية **B** **5 ثوان**.
- تستمر العملية **C** لغاية **الثانية 6** حيث تصل العملية **D**، وحيث ان الزمن المتبقي من العملية **C** هو **2 ثانية** اقل من زمن العملية **D**، لذلك تستمر العملية **C** في التنفيذ.
- عند **الثانية 8** تكون العملية **C** قد اتمت مهمتها في اللحظة التي وصلت فيها العملية **E**. وحيث ان العملية **E** هي الاقل زمنا فبالتالي يتم تنفيذها وينتقل المعالج للعملية **B**، ثم العملية **D**.

يمكن حساب متوسط زمن الانتظار حسب الجدول التالي :

العمليات	زمن الانتظار
A	0
B	6+3-2=7
C	4-4=0
D	15-6=9
E	8-8=0
متوسط زمن الانتظار	16/5=3.2ms





Process	Arrival Time	CPU Burst	Start Time	Waiting Time = Start Time - Arrival Time	Finish Time	Turnaround time= Finish Time - Arrival Time
A	0	3	0	0-0= **0**	3	3-0= **3**
B	2	6	10	10-3= **7**	15	15-2= **13**
C	4	4	4	4-4= **0**	8	8-4= **4**
D	6	5	15	15-6= **9**	20	20-6= **14**
E	8	2	8	8-8= **0**	10	10-8= **2**
Wt	0+7+0+9+0/5 = 3.2				Tt	3+13+4+14+2/5 = 7.2

4- جدولة الاسبقية (Priority Scheduling)

تعتبر جدولة **الشغل الاقصر اولا** (**SJF**) نوع خاص من **جدولة الاسبقية** حيث يمكن اعطاء العملية الاسبقية بنسبة (1/t) حيث t تمثل زمن التنفيذ لعملية ما.

في جدولة الاسبقية يتم **اعطاء كل عملية اسبقية** من **0-N**.

يمكن اعتبار **الترتيب التصاعدي** يمثل اسبقية العمليات من **الاعلى الى الادنى**.

وبالمثل فانه يمكن اعتبار **الترتيب التنازلي** يمثل اسبقية العمليات من **الادنى الي الاعلى**.
بمعنى انها تعتمد على نوع النظام المستخدم.

في هذا الباب فان الاسبقية تعطى على اساس **الترتيب التصاعدى** حيث يمثل **الرقم 0** العملية ذات **الاسبقية الاعلى** **والرقم 7** العملية ذات **الاسبقية الادنى**.

يمكن ان تكون جدولة الاسبقية **قابلة للتوقف او غير قابلة للتوقف**.

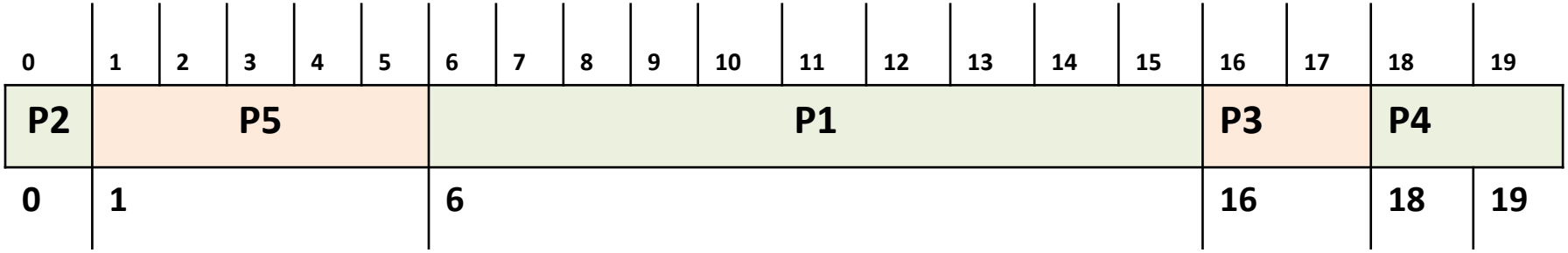
جدولة الاسبقية (Priority Scheduling)

مثال : لدينا 5 عمليات وصلت عند **الزمن 0** بالترتيب **P1, P2, P5 ...** بطول زمني (CPU-burst) مقاس بالملي ثانية بأسبقية محددة وفقا للجدول الاتي (**غير قابلة للتوقف**).

Process	Burst time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

باستخدام جدولة الاسبقية (كما هو موضح بالشكل التالي) فان :

متوسط زمن الانتظار = (1+6+16+18) / 5 = 8.2ms



Process	Priority	CPU Burst	Start Time	Waiting Time = Start Time - Arrival Time	Finish Time	Turnaround time= Finish Time - Arrival Time
P1	3	10	**6**	6-0= **6**	**16**	16-0= **16**
P2	1	1	**0**	0-0= **0**	**1**	1-0= **1**
P3	3	2	**16**	16-0= **16**	**18**	18-0= **18**
P4	4	1	**18**	18-0= **18**	**19**	19-0= **19**
P5	2	5	**1**	1-0= **1**	**6**	6-0= **6**
Wt	**6+0+16+18+1/5 = 8.2**				**Tt**	**16+1+18+19+6/5 = 12**

5- جدولة الراوند روبن (Round Robin RR)

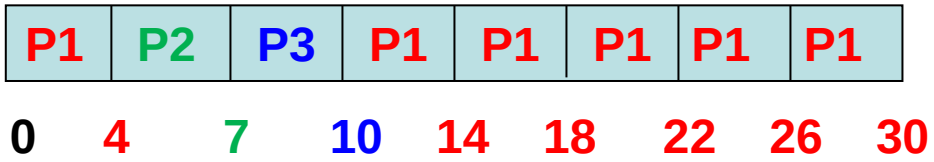
صممت **جدولة RR** خصيصا لنظام **المشاركة الزمنية (Time-Sharing)**. وهي تشبه جدولة الرتل ولكنها تتميز بانها **قابلة للتوقف (Pre-emptive)**. حيث **تعطى كل عملية شريحة من الزمن (Quantum) للتنفيذ** وهي عادة ما تكون بين **10- 100** ملي ثانية. حيث تعامل قائمة الجاهزية على **اساس رتل دائري**. حيث يخصص المعالج لكل عملية فترة من الزمن (1 شريحة من الزمن) فاذا لم تنهي العملية شغلها يتم توقيفها وتوضع في ذيل الرتل وينتقل المعالج لعملية اخرى. ام اذا انتهت قبل الزمن المخصص فإنها تسرح نفسها تلقائيا.

جدولة الراوند روبن Round Robin RR

مثال 1: لدينا **3 عمليات** وصلت بالترتيب **P1, P2, P3** بطول زمني (CPU-burst) مقاس بالملي ثانية **وبشريحة زمنية قدرها 4 ملي ثانية** وفقا للجدول الاتي.

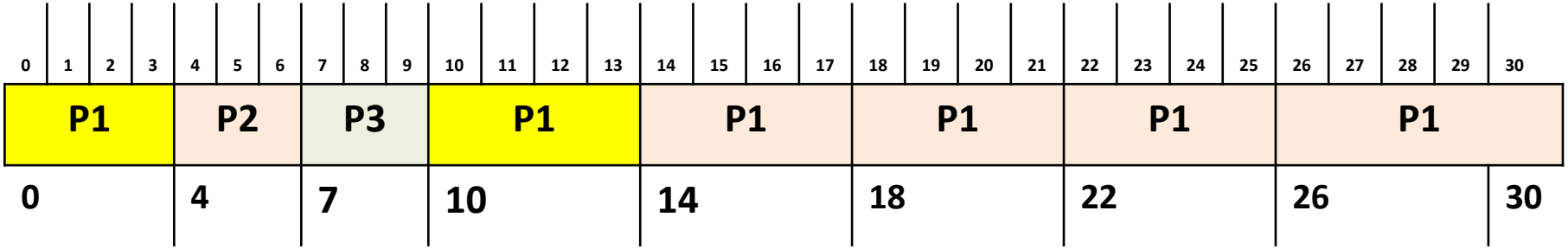
Process	Burst time
P1	24
P2	3
P3	3

باستخدام جدولة الراوند روبن RR (كما هو موضح بالشكل التالي) فان :



P2 سوف تنتظر **4 ملي ثانية**, **P3** سوف تنتظر **7 ملي ثانية**, **P1** سوف تنتظر **(10 -4)= 6** وعلى ذلك فان :

متوسط زمن الانتظار = (6+7+4)/3 = 17/3 = 5.66 ملي ثانية.



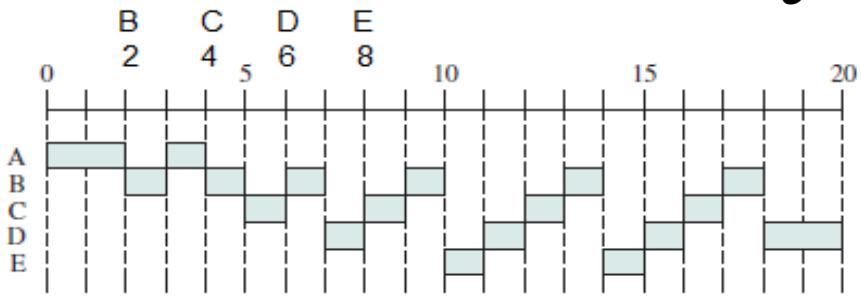
Process	Arrival Time	CPU Burst	Start Time	Waiting Time = Start Time - Arrival Time	Finish Time	Turnaround time= Finish Time - Arrival Time
P1	0	24	**10**	10-4= **6**	**30**	30-0= **30**
P2	0	3	**4**	4-0= **4**	**7**	7-0= **7**
P3	0	3	**7**	7-0= **7**	**10**	10-0= **10**
Wt	**6+4+7/3 = 5.66**				**Tt**	**30+7+10/3 = 15.66**

جدولة الراوند روبن Round Robin RR

مثال 2 : اذا كان لدينا العمليات **A، B، C، D، E** قد وصلوا في زمن محدد بالجدول التالي وكل عملية تحتاج الى زمن تنفيذ ايضا محدد بالجدول.

- فأوجد **متوسط زمن الانتظار الاجمالي؟**
- اوجد (**Tr**) **Turnaround time**؟ وكذلك نسبة **Tr / Ts** حيث **Ts** يمثل **زمن التنفيذ.**
- أوجد **زمن انتهاء العمليات Finish time؟**

الحل

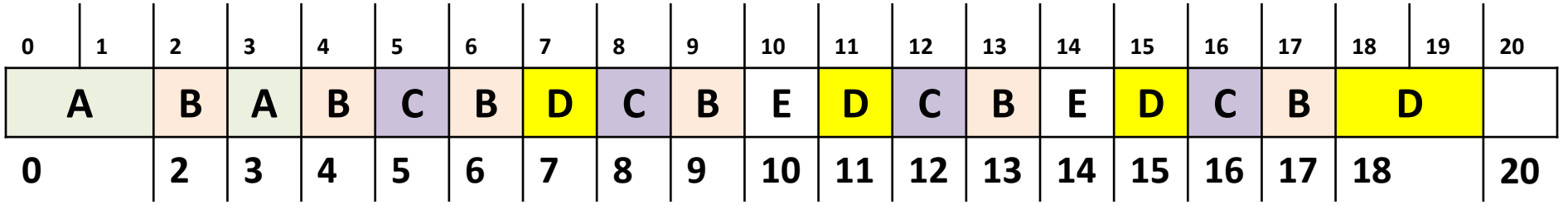


Round-robin (RR), $q = 1$

العمليات	زمن الانتظار
A	1
B	2+1+1+2+3+3=12-2=10
C	5+2+3+3=13-4=9
D	7+3+3+2=15-6=9
E	10+3=13-8=5
متوسط زمن الانتظار	34/5=6.8

العمليات	زمن الوصول	زمن التنفيذ
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

حيث الرقم المطروح يمثل زمن الوصول



Process	Arrival Time	CPU Burst	Start Time	Waiting Time = Start Time - Arrival Time	Finish Time	Turnaround time= Finish Time - Arrival Time
A	0	3	**3**	3-2= **1**	**4**	4-0= **4**
B	2	6	**17**	(4-3)+(6-5)+(9-7)+(13-10)+(17-14)= **10**	**18**	18-2= **16**
C	4	4	**16**	(8-5)+(12-9)+(16-13)= **9**	**17**	17-4= **13**
D	6	5	**18**	(11-7)+(15-12)+(18-16) = **9**	**20**	20-6= **14**
E	8	2	**14**	(14-9)= **5**	**15**	15-8= **7**
Wt	**1+10+9+9+5/5 = 6.8**				**Tt**	**4+16+13+14+7/5 = 10.8**

بجمع زمن التنفيذ من الجدول الاول مع زمن الانتظار في الجدول الثاني نحصل على الزمن المستغرق الكلي (Turnaround time) او اكما هو موضح بالجدول التالي.

RR $q=1$	A	B	C	D	E	
Finish Time	4	18	17	20	15	المتوسط
Turnaround Time (T_T)	4	16	13	14	7	10.80
T_T/T_S	1.33	2.67	3.25	2.80	3.50	2.71

العمليات	زمن الوصول	زمن التنفيذ
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

4/3

الزمن الكلي النسبي : هو نسبة الزمن الكلي للزمن المستغرق للتنفيذ وهو يوّضح زمن التأخيرُ الذي حصل للعملية.

حيث، كلما كان زمن تنفيذ العملية طويل، كلما كان بإمكانها التسامح مع التأخير.