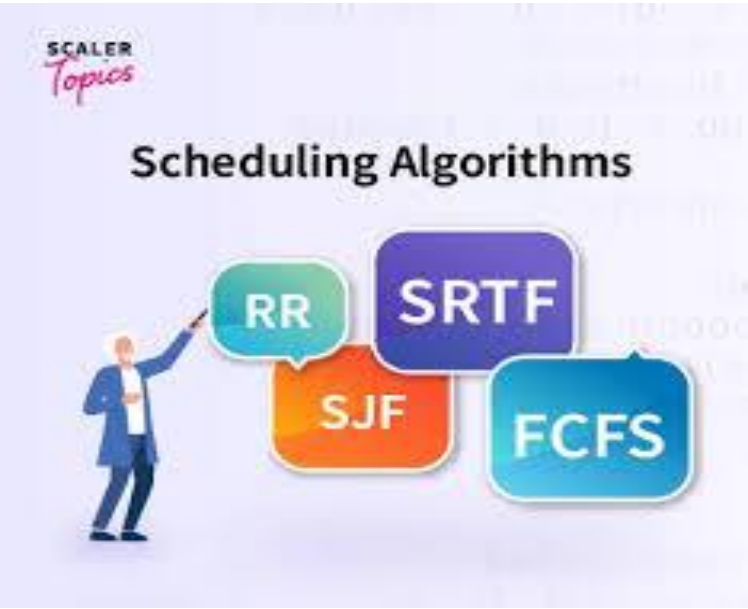
Operating Systems

Lecture # 10

Department of Computer

4th Class

Operating System Scheduling Algorithms



By

Dr. Ahmed Khudhair Abbas

Computer and Internet Center

جدولة وحدة المعالجة المركزية CPU Scheduling

في أنظمة Uni-Programming مثل MS DOS ، عندما تنتظر إحدى العمليات إجراء أي عملية إدخال / إخراج ، تظل وحدة المعالجة المركزية معطلة. هذا عبء لأنه يضيع الوقت ويسبب مشكلة الجوع Starvation ومع ذلك في أنظمة البرمجة المتعددة لا تبقى وحدة المعالجة المركزية متوقفة أثناء وقت انتظار العملية وتبدأ في تنفيذ العمليات الأخرى ويجب أن يحدد نظام التشغيل العملية التي سيتم منحها وحدة المعالجة المركزية CPU .

في أنظمة البرمجة المتعددة **systems** Multiprogramming يقوم نظام التشغيل بجدولة العمليات على وحدة المعالجة المركزية لتحقيق أقصى استفادة منها ويسمى هذا الإجراء جدولة وحدة المعالجة المركزية ويستخدم نظام التشغيل خوارزمية جدولة مختلفة لجدولة العمليات.

خوارزميات الجدولة Scheduling Algorithms

هناك العديد من الخوارزميات التي يستخدمها نظام التشغيل لجدولة العمليات على المعالج بطريقة فعالة والغرض من خوارزمية الجدولة هو:

1 .أقصى استخدام لوحدة المعالجة المركزية Maximum CPU utilization
2 .التخصيص العادل لوحدة المعالجة المركزية Fair allocation of CPU
3 .الحد الأقصى للإنتاجية Maximum throughput
4 .الحد الأدنى من الوقت المستغرق Minimum turnaround time
5 .الحد الأدنى لوقت الانتظار Minimum waiting time
6 .الحد الأدنى من وقت الاستجابة Minimum response time

يقوم برنامج جدولة العمليات بجدولة عمليات مختلفة ليتم تخصيصها لوحدة المعالجة المركزية بناءً على خوارزميات جدولة معينة. هناك ست خوارزميات شائعة لجدولة العمليات وهي:

جدولة من يأتي أولاً يخدم أولاًFirst-Come, First-Served (FCFS) Scheduling

جدولة أقصر مهمة تالية Shortest-Job-Next (SJN) Scheduling

جدولة الأولوية Priority Scheduling

أقصر وقت متبقي Shortest Remaining Time

جدولة راوند روبن Round Robin (RR) Scheduling

جدولة قوائم الانتظار متعددة المستويات Multiple-Level Queues Scheduling

هذه الخوارزميات إما غير استباقية أو استباقية **non-preemptive or preemptive** تم تصميم الخوارزميات غير الاستباقية **non-preemptive** بحيث بمجرد دخول العملية إلى حالة التشغيل لا يمكن ايقافها حتى تكمل الوقت المخصص لها وفي حين أن الجدولة الاستباقية **preemptive** تعتمد على الأولوية حيث قد يستبق المجدول عملية تشغيل ذات أولوية منخفضة في أي وقت عندما تكون الأولوية عالية تدخل العملية في حالة جاهزة.

1. جدولة من يأتي أولاً يخدم أولاً First-Come, First-Served (FCFS) Scheduling

تقوم خوارزمية جدولة من يأتي أولاً يخدم أولاً (FCFS) بجدولة الوظائف وفقًا لوقت وصولها. ستحصل المهمة التي تأتي أولاً في قائمة الانتظار الجاهزة على وحدة المعالجة المركزية أولاً. كلما قل وقت وصول الوظيفة ، زادت سرعة حصول الوظيفة على وحدة المعالجة المركزية. قد تتسبب جدولة FCFS في حدوث مشكلة الجوع إذا كان وقت الاندفاع للعملية الأولى هو الأطول بين جميع الوظائف ومن اهم مزايا هذه الطريقة البساطة وسهولة التطبيق ومن يأتي أولا يخدم أولا اما من اهم العيوب لهذه الخوارزمية فهي:

1 .طريقة الجدولة ليست وقائية **non-preemptive** وسوف تستمر العملية حتى الاكتمال.

2 .بسبب الطبيعة غير الوقائية للخوارزمية قد تحدث مشكلة الجوع Starvation

3 .على الرغم من سهولة تنفيذها إلا أنها ضعيفة في الأداء نظرًا لأن متوسط وقت الانتظار أعلى مقارنة بخوارزميات الجدولة الأخرى.

مثال Example

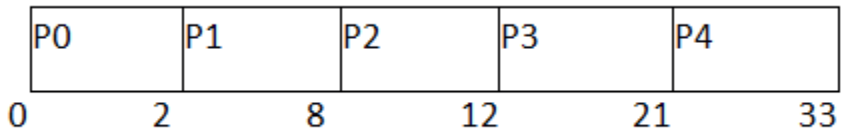
في الجدول التالي، توجد 5 عمليات بمعرف العملية P0 و P1 و P2 و P3 و .P4 تصل P0 في الوقت 0 ، P1 في الوقت 1 ، P2 في الوقت 2 ، P3 تصل في الوقت 3 وتصل العملية P4 في الوقت 4 في قائمة الانتظار الجاهزة ويتم حساب وقت الاستجابة ووقت الانتظار باستخدام الصيغة التالية.

1. Turn Around Time = Completion Time - Arrival Time
2. Waiting Time = Turnaround time - Burst Time

يتم تحديد متوسط وقت الانتظار من خلال جمع وقت الانتظار المعني لجميع العمليات وقسمة المجموع على إجمالي عدد العمليات.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P0	0	2	2	2	0
P1	1	6	8	7	1
P2	2	4	12	10	6
P3	3	9	21	18	9
P4	4	12	33	29	17

Avg Waiting Time=31/5

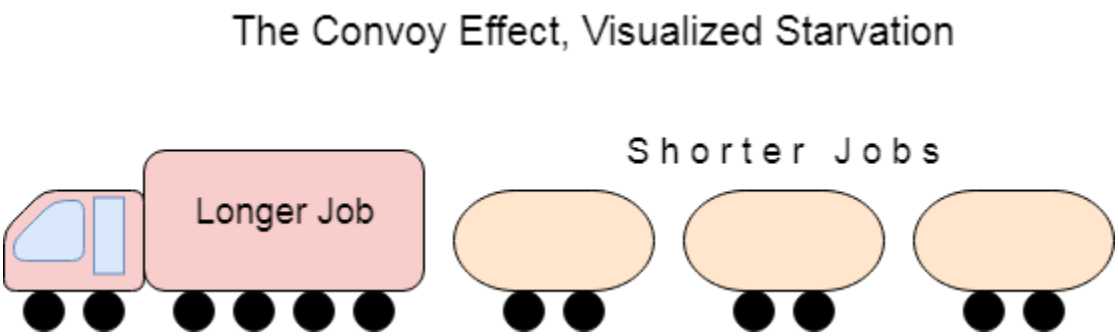


(Gantt chart)

تأثير المواكبة او المجاعة Convoy effect or Starvation **في FCFS**

قد يعاني FCFS من تأثير المواكبة إذا كان وقت تنفيذ المهمة الأولى هو الأعلى بين الجميع. كما هو الحال في الحياة الواقعية إذا كانت القافلة تمر عبر الطريق فقد يتم حظر الأشخاص الآخرين حتى تمر بشكل كامل ويمكن محاكاة ذلك في نظام التشغيل أيضًا.

إذا حصلت وحدة المعالجة المركزية على عمليات وقت الوصول الأعلى في الطرف الأمامي لقائمة الانتظار الجاهزة، فقد يتم حظر عمليات وقت الوصول المنخفض مما يعني أنها قد لا تحصل على وحدة المعالجة المركزية مطلقًا إذا كانت المهمة في التنفيذ تستغرق وقتًا طويلاً للغاية. وهذا ما يسمى تأثير المواكبة أو الجوع Convoy effect or Starvation.



Example

In the Example, We have 3 processes named as **P1, P2 and P3**. The Burt Time of process P1 is highest. The Turnaround time and the waiting time in the following table, are calculated by the formula:

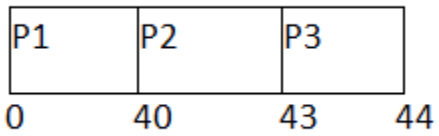
Turn Around Time = Completion Time - Arrival Time

Waiting Time = Turn Around Time - Burst Time

في السيناريو الأول، تصل العملية P1 إلى المرتبة الأولى في قائمة الانتظار بالرغم من ذلك فان وقت وصول العملية هو الأعلى بين الجميع. نظرًا لأن خوارزمية الجدولة التي نتبعها هي FCFS ، فإن وحدة المعالجة المركزية ستنفذ العملية P1 أولاً.

في هذا الجدول سيكون متوسط وقت انتظار النظام مرتفعًا جدًا ويتعين على العمليات الأخرى P2 و P3 انتظار دورها لمدة 40 وحدة زمنية على الرغم من أن وقت وصولها منخفض للغاية. هذا الجدول يعاني من الجوع starvation

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	**0**	**40**	**40**	**40**	**0**
P2	**1**	**3**	**43**	**42**	**39**
P3	**1**	**1**	**44**	**43**	**42**

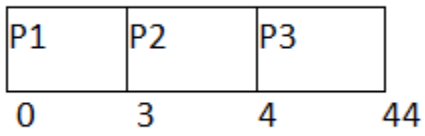


Avg waiting Time = 81/3

في السيناريو الثاني إذا كانت العملية P1 قد وصلت إلى آخر قائمة الانتظار والعمليات الأخرى P2 و P3 في وقت سابق ، فلن تكون مشكلة الجوع موجودة.

يوضح المثال التالي الانحراف في أوقات الانتظار لكلا السيناريوهات على الرغم من أن طول الجدول الزمني هو نفسه 44 وحدة ، إلا أن وقت الانتظار سيكون أقل في هذا الجدول.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
1	**1**	**40**	**44**	**43**	**3**
2	**0**	**3**	**3**	**3**	**0**
3	**0**	**1**	**4**	**4**	**3**



Avg Waiting Time = 6/3

2. جدولة أقصر مهمة أولاً Shortest Job First (SJF) Scheduling

تقوم خوارزمية جدولة SJF بجدولة العمليات وفقًا لوقت وصولها. في جدولة SJF ، ستتم جدولة العملية ذات أقل وقت تنفيذ من بين قائمة العمليات المتاحة في قائمة الانتظار الجاهزة ومن اهم مزايا هذه الطريقة هو تحقيقها الحد الأقصى من الإنتاجية والحد الأدنى لمتوسط الانتظار ووقت الاستجابة . اما عيوبها فقد تعاني من مشكلة الجوع Starvation .

Example

In the following example, there are five jobs named as P1, P2, P3, P4 and P5. Their arrival time and burst time are given in the table below.

PID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
p1	1	7	8	7	0
p2	3	3	13	10	7
p3	6	2	10	4	2
p4	7	10	31	24	14
p5	9	8	21	12	4

	P1	P3	P2	P5	P4	
0	1	8	10	13	21	31

Avg Waiting Time = 27/5

كما نلاحظ لم تصل أي عملية في الوقت 0 وبالتالي ستكون هناك فتحة فارغة في مخطط جانت من الوقت 0 إلى 1 (الوقت الذي تصل فيه العملية الأولى) ووفقًا للخوارزمية، يقوم نظام التشغيل بجدولة العملية التي لها أقل وقت تنفيذ بين العمليات المتاحة في قائمة الانتظار الجاهزة.

حتى الآن، لدينا عملية واحدة فقط في قائمة الانتظار الجاهزة ومن ثم سيقوم المجدول بجدولة ذلك إلى المعالج بغض النظر عن وقت تنفيذه وسيتم تنفيذ هذا حتى 8 وحدات زمنية. حتى ذلك الحين، وصلنا ثلاث عمليات أخرى إلى قائمة الانتظار الجاهزة، ومن ثم سيختار المجدول العملية بأقل وقت للتنفيذ.

من بين العمليات الواردة في الجدول، سيتم تنفيذ P3 بعد ذلك نظرًا لأنه يحتوي على أقل وقت تنفيذ بين جميع العمليات المتاحة وهذه هي الطريقة التي سيستمر بها الإجراء في خوارزمية جدولة أقصر مهمة أولاً.(SJF)

خوارزمية جدولة أقصر وقت متبقي أولاً Shortest Remaining Time First (SRTF)

هذه الخوارزمية هي النسخة الاستباقية **preemptive version** of **SJF scheduling** لجدولة .SJF في SRTF ويمكن إيقاف تنفيذ العملية بعد فترة زمنية معينة. عند وصول كل عملية يقوم برنامج الجدولة قصير المدى بجدولة العملية بأقل وقت اندفاع متبقي بين قائمة العمليات المتاحة والعملية الجارية وبمجرد توفر جميع العمليات في قائمة الانتظار الجاهزة ، لن يتم إجراء أي إجراءات استباقية وستعمل الخوارزمية كجدولة .SJF يتم حفظ سياق العملية في كتلة التحكم في العملية عند إزالة العملية من التنفيذ ويتم جدولة العملية التالية. يتم الوصول إلى هذا PCB عند التنفيذ التالي لهذه العملية.

In this Example, there are five jobs P1, P2, P3, P4, P5 and P6. Their arrival time and burst time are given below in the table.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time	Response Time
p1	0	8	20	20	12	0
p2	1	4	10	9	5	1
p3	2	2	4	2	0	2
p4	3	1	5	2	1	4
p5	4	3	13	9	6	10
p6	5	2	7	2	0	5

P1	P2	P3	P4	P5	P6	P7	P8	P9
0	1	2	3	4	5	6	7	8

Avg Waiting Time = 24/6

The Gantt chart is prepared according to the arrival and burst time given in the table.

1. نظرًا لأنه في الوقت 0 ، فإن العملية الوحيدة المتاحة هي P1 مع وقت اندفاع وحدة المعالجة المركزية 8. هذه هي العملية الوحيدة المتاحة في القائمة وبالتالي تمت جدولتها.
2. تصل العملية التالية إلى الوحدة الزمنية 1. نظرًا لأن الخوارزمية التي نستخدمها هي SRTF وهي خوارزمية استباقية، يتم إيقاف التنفيذ الحالي ويتحقق المجدول من العملية بأقل وقت للتنفيذ.
3. حتى الآن ، هناك عمليتان متاحتان في قائمة الانتظار الجاهزة. نفذ نظام التشغيل P1 لوحدة زمنية واحدة حتى الآن ؛ وقت التنفيذ المتبقي لـ P1 هو 7 وحدات. وقت تنفيذ العملية P2 هو 4 وحدات. ومن ثم تمت جدولة العملية P2 على وحدة المعالجة المركزية وفقًا للخوارزمية.
4. تصل العملية التالية P3 إلى الوحدة الزمنية 2. في هذا الوقت ، يتم إيقاف تنفيذ العملية P3 ويتم البحث في العملية ذات أقل وقت للتنفيذ المتبقي. نظرًا لأن العملية P3 تحتوي على وحدتين من الوقت ، فسيتم منحها الأولوية على الآخرين.

5. تصل العملية التالية P4 إلى الوحدة الزمنية 3. عند هذا الوصول ، سيوقف المجدول تنفيذ P4 ويتحقق من العملية التي لديها أقل وقت للتنفيذ بين العمليات المتاحة (P1 و P2 و P3 و P1 .(P4 و P2 لهما الوقت المتبقي 7 وحدات و 3 وحدات على التوالي. P3و P4 لهما وقت متبقي 1 وحدة لكل منهما. نظرًا لأن كلاهما متساويان ، فسيتم إجراء الجدولة وفقًا لوقت وصولهما. يصل P3 في وقت أبكر من P4 وبالتالي سيتم جدولته مرة أخرى.
6. تصل العملية التالية P5 إلى الوحدة الزمنية 4. حتى هذا الوقت ، أكملت العملية P3 تنفيذها ولم تعد موجودة في القائمة. سيقارن المجدول وقت الاندفاع المتبقي لجميع العمليات المتاحة. بما أن زمن التنفيذ للعملية P4 هو 1 وهو الأقل من بين كل ذلك، فسيتم جدولته.
7. تصل العملية التالية P6 إلى الوحدة الزمنية 5 ، حتى هذا الوقت ، أكملت العملية P4 تنفيذها. لدينا 4 عمليات متاحة حتى الآن، وهي (7) P1 و (3) P2 و (3) P5 و .(2) P6 وقت التنفيذ لـ P6 هو الأقل بين جميع الأوقات ، ومن ثم تمت جدولة P6.، أصبحت جميع العمليات متاحة ، وبالتالي ستعمل الخوارزمية الآن مثل .SJF سيتم تنفيذ P6 حتى اكتماله ثم سيتم جدولة العملية مع أقل وقت متبقي.

بمجرد وصول جميع العمليات، لا يتم إجراء أي إجراءات وقائية وستعمل الخوارزمية مثل.SJF

Example: Given the arrival time and burst time of 3 jobs in the table below. Calculate the Average waiting time of the system.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	9	13	13	4
P2	1	4	5	4	0
P3	2	9	22	20	11

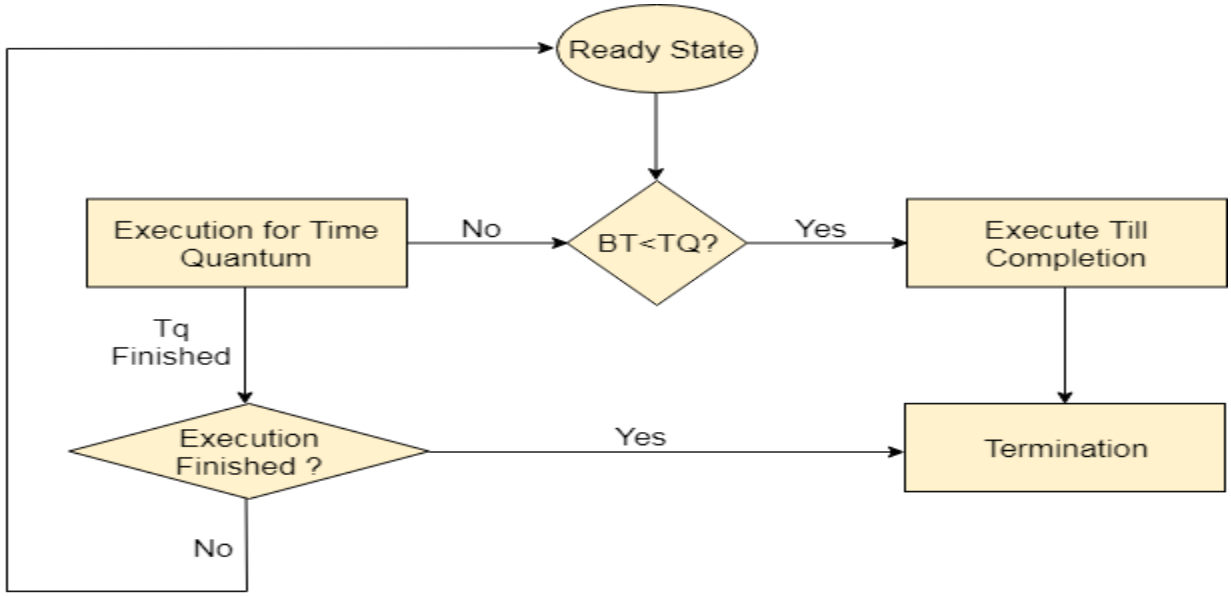
توجد ثلاث وظائف P1 و P2 و .P3 يصل P1 في الوحدة الزمنية 0 ؛ سيتم جدولته أولاً في الوقت حتى وصول العملية التالية. يصل P2 في وحدة زمنية واحدة. وقت تنفيذها هو 4 وحدات وهو الأقل بين الوظائف في قائمة الانتظار. ومن ثم سيتم جدولتها بعد ذلك وفي الوقت 2 ، سيصل P3 مع وقت تنفيذ 9. حيث أن وقت التنفيذ المتبقي لـ P2 هو 3 وحدات وهي الأقل من بين الوظائف المتاحة. ومن ثم سيستمر المعالج في تنفيذه حتى اكتماله. نظرًا لأن جميع الوظائف قد وصلت ، فلن يتم إجراء أي إجراءات استباقية الآن وسيتم تنفيذ جميع الوظائف حتى الانتهاء وفقًا لـ .SJF

P1	P2	P3	P4	P5	
0	1	2	5	13	22

Avg Waiting Time = (4+0+11)/3 = 5 units

خوارزمية جدولة روبن Round Robin Scheduling Algorithm

تعد خوارزمية جدولة Round Robin واحدة من أكثر خوارزمية الجدولة شيوعًا والتي يمكن تنفيذها بالفعل في معظم أنظمة التشغيل. هذه هي النسخة الاستباقية Preemptive version من جدولة من يأتي أولاً يُخدم أولاً. تركز الخوارزمية على مشاركة الوقت. في هذه الخوارزمية، يتم تنفيذ كل عملية بطريقة دورية. يتم تحديد شريحة زمنية معينة في النظام تسمى كمية الوقت Quantum يتم تخصيص وحدة المعالجة المركزية لكل عملية موجودة في قائمة الانتظار الجاهزة لهذا الوقت الكمي، إذا اكتمل تنفيذ العملية خلال ذلك الوقت، فستنتهي العملية وإلا ستعود العملية إلى قائمة الانتظار الجاهزة Ready queue وتنتظر حتى يكتمل الدور التالي للتنفيذ.



مزايا Advantages

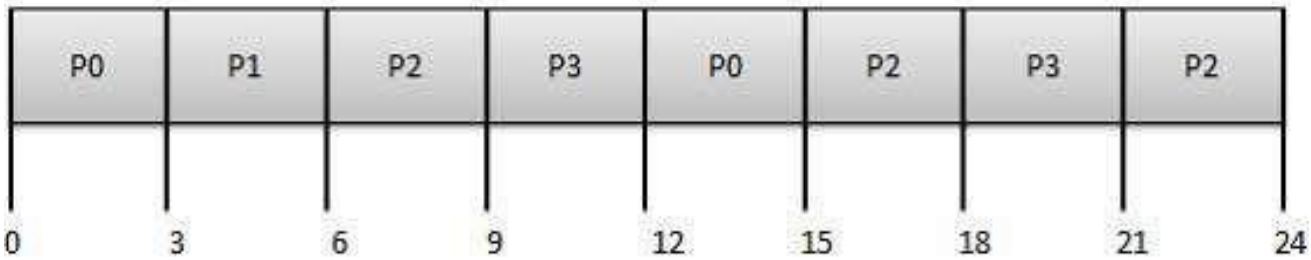
- يمكن أن يكون قابلاً للتنفيذ فعليًا في النظام لأنه لا يعتمد على وقت التنفيذ
- لا يعاني من مشكلة الجوع starvation
- تحصل جميع الوظائف على تخصيص ثابت لوحدة المعالجة المركزية.

سلبيات Disadvantages

- كلما زاد مقدار الوقت ، زاد وقت الاستجابة في النظام.
- كلما انخفض مقدار الوقت ، زادت كلفة التبديل في النظام.
- تحديد الوقت المثالي time quantum هو في الحقيقة مهمة صعبة للغاية في النظام.

Process	Arrival Time	Execute Time
P0	0	5
P1	1	3
P2	2	8
P3	3	6

Quantum = 3



Wait time of each process is following

Process	Wait Time : Service Time - Arrival Time
P0	(0-0) + (12-3) = 9
P1	(3-1) = 2
P2	6-2) + (15-9) = 10
P3	(9-3) + (18-12) = 12

Average Wait Time: (9+2+10+12) / 4 = 8.25

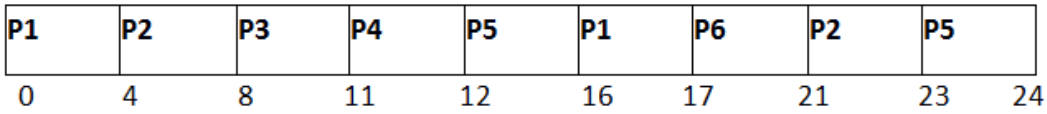
RR Scheduling Example

In the following example, there are six processes named as P1, P2, P3, P4, P5 and P6. Their arrival time and burst time are given below in the table. The time quantum of the system is 4 units.

Process ID	Arrival Time	Burst Time
P1	0	5
P2	1	6
P3	2	3
P4	3	1
P5	4	5
P6	6	4

According to the algorithm, we have to maintain the ready queue and the Gantt chart. The structure of both the data structures will be changed after every scheduling.

GANTT chart



The completion time, Turnaround time and waiting time will be calculated as shown in the table below.

As, we know,

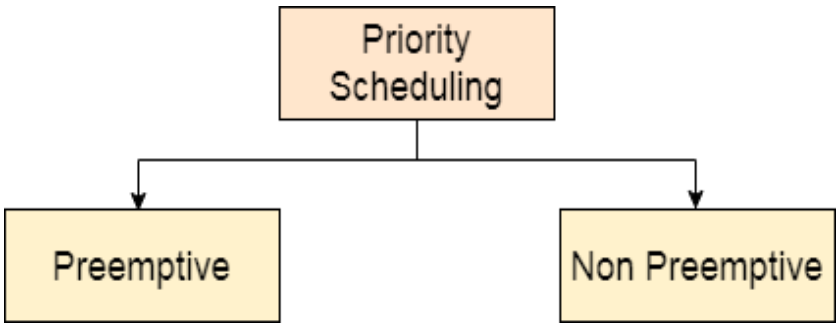
1. Turn Around Time = Completion Time - Arrival Time
2. Waiting Time = Turn Around Time - Burst Time

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	5	17	17	12
P2	1	6	23	22	16
P3	2	3	11	9	6
P4	3	1	12	9	8
P5	4	5	24	20	15
P6	6	4	21	15	11

Avg Waiting Time = (12+16+6+8+15+11)/6 = 76/6 units

جدولة الأولوية Priority Scheduling

في جدولة الأولوية، يوجد رقم أولوية معين لكل عملية. في بعض الأنظمة كلما انخفض الرقم زادت الأولوية. بينما في الحالات الأخرى كلما زاد الرقم زادت الأولوية ويتم إعطاء العملية ذات الأولوية الأعلى بين العمليات المتاحة وحدة المعالجة المركزية. ويوجد نوعان من خوارزمية جدولة الأولوية. أحدهما هو جدولة الأولوية الاستباقية **Preemptive** بينما الآخر هو جدولة الأولوية غير الاستباقية **Non-Preemptive**



قد يتغير أو لا يتغير رقم الأولوية المعين لكل عملية. إذا لم يتغير رقم الأولوية من تلقاء نفسه طوال العملية ، فإنه يسمى أولوية ثابتة **Static priority** بينما إذا استمر في تغيير نفسه على فترات منتظمة ، فإنه يسمى الأولوية الديناميكية **Dynamic priority**

جدولة الأولوية غير الاستباقية Non Preemptive Priority Scheduling

في جدولة الأولوية غير الاستباقية، تتم جدولة العمليات وفقًا لرقم الأولوية المعين لها. بمجرد جدولة العملية، سيتم تشغيلها حتى اكتمالها. بشكل عام، كلما انخفض رقم الأولوية، زادت أولوية العملية.

Example

In the Example, there are 7 processes P1, P2, P3, P4, P5, P6 and P7. Their priorities, Arrival Time and burst time are given in the table.

Process ID	Priority	Arrival Time	Burst Time
P1	2	0	3
P2	6	2	5
P3	3	1	4
P4	5	4	2
P5	7	6	9
P6	4	5	4
P7	10	7	10

يمكننا إعداد مخطط جانت وفقًا لجدولة الأولوية غير الاستباقي .Non Preemptive تصل العملية P1 في الوقت 0 مع وقت تنفيذ 3 وحدات ورقم الأولوية 2. نظرًا لعدم وصول أي عملية أخرى حتى الآن، سيقوم نظام التشغيل بجدولتها على الفور وفي الوقت نفسه، تم تنفيذP1 ووصلت عمليتان أخريان P2 و .P3 نظرًا لأن أولوية P3 هي 3 ، فإن وحدة المعالجة المركزية ستنفذ P3على.P2 وفي الوقت نفسه تنفيذP3 ، تتوفر جميع العمليات في قائمة الانتظار الجاهزة. ستعطى الأولوية للعملية ذات أقل رقم أولوية. نظرًا لأن P6 له رقم أولوية تم تعيينه على أنه 4 ، فسيتم تنفيذه بعد P3 مباشرةً. وبعدP6 ، يكون P4 هو أقل رقم أولوية بين العمليات المتاحة ؛ سيتم تنفيذه طوال وقت التنفيذ و نظرًا لأن جميع الوظائف متوفرة في قائمة الانتظار الجاهزة ، فسيتم تنفيذ جميع الوظائف وفقًا لأولوياتها. إذا كانت هناك وظيفتان لهما رقم أولوية مماثل مخصص لهما، فسيتم تنفيذ المهمة ذات أقل وقت وصول.

P1	P3	P6	P4	P2	P5	P7	
0	3	7	11	13	18	27	37

From the GANTT Chart prepared, we can determine the completion time of every process. The turnaround time, waiting time and response time will be determined.

1. Turn Around Time = Completion Time - Arrival Time
2. Waiting Time = Turn Around Time - Burst Time

Process Id	Priority	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
P1	2	0	3	3	3	0	0
P2	6	2	5	18	16	11	13
P3	3	1	4	7	6	2	3
P4	5	4	2	13	9	7	11
P5	7	6	9	27	21	12	18
P6	4	5	4	11	6	2	7
P7	10	7	10	37	30	18	27

Avg Waiting Time = (0+11+2+7+12+2+18)/7 = 52/7 units

جدولة الأولوية الاستباقية Preemptive Priority Scheduling

في جدولة الأولوية الاستباقية في وقت وصول العملية في قائمة الانتظار الجاهزة تتم مقارنة أولويتها مع أولوية العمليات الأخرى الموجودة في قائمة الانتظار الجاهزة وكذلك مع العملية التي يتم تنفيذها بواسطة وحدة المعالجة المركزية في تلك المرحلة من الوقت سيتم إعطاء وحدة المعالجة المركزية التي لها أولوية قصوى بين جميع العمليات المتاحة.

يتمثل الاختلاف بين جدولة الأولوية الاستباقية وجدولة الأولوية غير الاستباقية في أنه، في الجدولة الأولوية الاستباقية يمكن إيقاف العملية التي يتم تنفيذها عند وصول عملية ذات أولوية أعلى.

بمجرد توفر جميع الوظائف في قائمة الانتظار الجاهزة، ستعمل الخوارزمية كجدولة أولوية غير استباقية، مما يعني أن المهمة المجدولة ستعمل حتى الانتهاء ولن يتم إجراء أي إجراءات استباقية

Example: There are 7 processes P1, P2, P3, P4, P5, P6 and P7 given. Their respective priorities, Arrival Times and Burst times are given in the table below.

Process Id	Priority	Arrival Time	Burst Time
P1	2(L)	0	1
P2	6	1	7
P3	3	2	3
P4	5	3	6
P5	4	4	5
P6	10(H)	5	15
P7	9	15	8

GANTT chart

P1	**P2**	**P3**	**P5**	**P4**	**P2**	**P7**	**P6**	
0	**1**	**2**	**5**	**10**	**16**	**22**	**30**	**45**

The Completion Time of each process is determined with the help of GANTT chart. The turnaround time and the waiting time can be calculated by the following formula.

1. Turnaround Time = Completion Time - Arrival Time
2. Waiting Time = Turn Around Time - Burst Time

Process Id	Priority	Arrival Time	Burst Time	Completion Time	Turn around Time	Waiting Time
P1	2	0	1	1	1	0
P2	6	1	7	22	21	14
P3	3	2	3	5	3	0
P4	5	3	6	16	13	7
P5	4	4	5	10	6	1
P6	10	5	15	45	40	25
P7	9	6	8	30	24	16

Avg Waiting Time = (0+14+0+7+1+25+16)/7 = 63/7 = 9 units