

جدولة وحدة المعالجة

CPU Scheduling



Chapter 5



Lecture 8 and 9



جدولة وحدة المعالجة المركزية

CPU scheduling

- **إن جدولة وحدة المعالجة** هي أساس نظم التشغيل المتعددة البرمجة، حيث يستطيع نظام التشغيل من خلال تبديل التنفيذ بين الإجراءات، من جعل الحاسوب أكثر إنتاجية.
- **إن الهدف الأساسي من تعدد البرمجة** هو أن يكون لدينا دوماً إجراءات تنفذ، وذلك لرفع معدل استخدام وحدة المعالجة إلى أعلى حد ممكن، ففي نظام أحادي المعالج، لا يمكن وجود أكثر من إجراءات واحدة في حال التنفيذ، وإذا وجدت أكثر من إجراءات في النظام، فإن واحدة منها ستكون في حالة التنفيذ، وأما البقية فستنتظر إلى أن تحرر وحدة المعالجة وتعاد جدولتها.
- **يقوم نظام التشغيل بانتقاء إحدى الإجراءات الجاهزة للتنفيذ**، وعندما تصبح وحدة المعالجة في حالة راحة يجري اتخاذ قرارات الجدولة في الظروف التالية:
 - **تبدل حالة إجراءات من حالة التنفيذ إلى حالة الانتظار، أو من حالة التنفيذ إلى حالة الجاهزية، أو من حالة الانتظار إلى حالة الجاهزية، أو عند انتهاء الإجراءات.**



المجدولات في وحدة المعالجة schedulers

• مهمة المجدول هو تنظيم دخول الإجراءات إلى المعالج ، هنالك ثلاثة أنواع من المجدولات، ولكل مجدول عمل يقوم به :

1. **المجدول طويل الأمد (مجدول الأعمال) Long term scheduler**: يختار من بين

مجموعة الإجراءات المراد تنفيذها، والموجودة على القرص الصلب، من أجل تحميلها في الذاكرة لبدء تنفيذها. وتكون الفترة بين اختيار إجرائية وأخرى عدة دقائق . ويتحكم بعدد البرامج في الذاكرة ، فالمجدول يقوم بـ :

✓ اختيار الإجراءات التي ستحمل في الذاكرة (إدخال الإجراءات إلى صف الانتظار).

✓ يحدد أي إجرائية ستبدأ اعتماداً على الترتيب والأفضلية.

✓ لا يُستخدم في أنظمة التقسيم الزمني (Time sharing systems) .

2. **المجدول متوسط الأمد Medium term scheduler** : يضاف في بعض نظم التشغيل (خاصة نظم المشاركة بزمان المعالج)، حيث يعتمد على إزالة الإجراءات من/إلى الذاكرة أي التبديل بين الإجراءات المنفذة والمراد تنفيذها، من أجل تحسين الإجراءات التي يجري تنفيذها ، فالمجدول :
- ✓ **يجدول الإجراءات بناءً على الموارد التي تحتاج (ذاكرة/ دخل/ خرج).**
 - ✓ **يقوم بتعليق suspend الإجراءات التي لم تتوفر لها مواردها حالياً.**
 - ✓ **عادة تكون الذاكرة ذات مورد محدود وهنا يعمل مدير الذاكرة كمجدول متوسط الأمد (استخدام الذاكرة الافتراضية).**

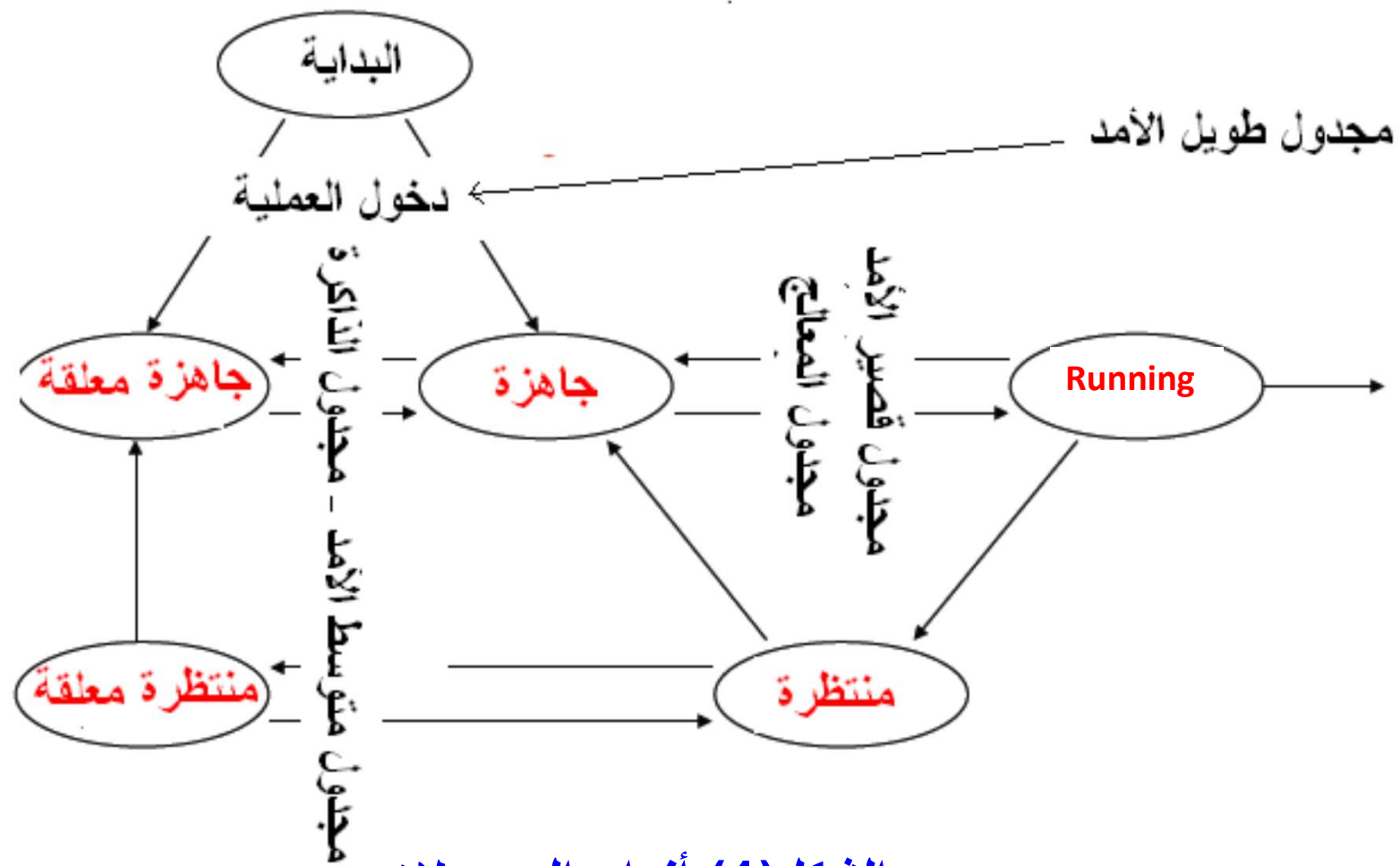
3. **المجدول قصير الأمد (مجدول وحدة المعالجة) Short term scheduler**: يختار إجراءات من بين الإجراءات الجاهزة للتنفيذ، وذلك لكي تُسند إلى وحدة المعالجة، والفرق الأساسي بين المجدولين هو في تواتر تنفيذهما، فالمجدول قصير الأمد ينتخب إجراءات جديدةً لوحدة المعالجة بتواتر كبير (يستخدم مرة على الأقل كل 100 ميلي ثانية)، بينما ينفذ المجدول طويل الأمد بتواتر أقل بكثير (من مرتبة دقائق)، فهو غالباً يُستدعى عندما تغادر إجراءات ما النظام.

✓ يقوم المجدول باختيار إجراءات من الإجراءات الموجودة بالذاكرة والجاهزة للتنفيذ (ready queue) وحجز وحدة المعالجة لها.

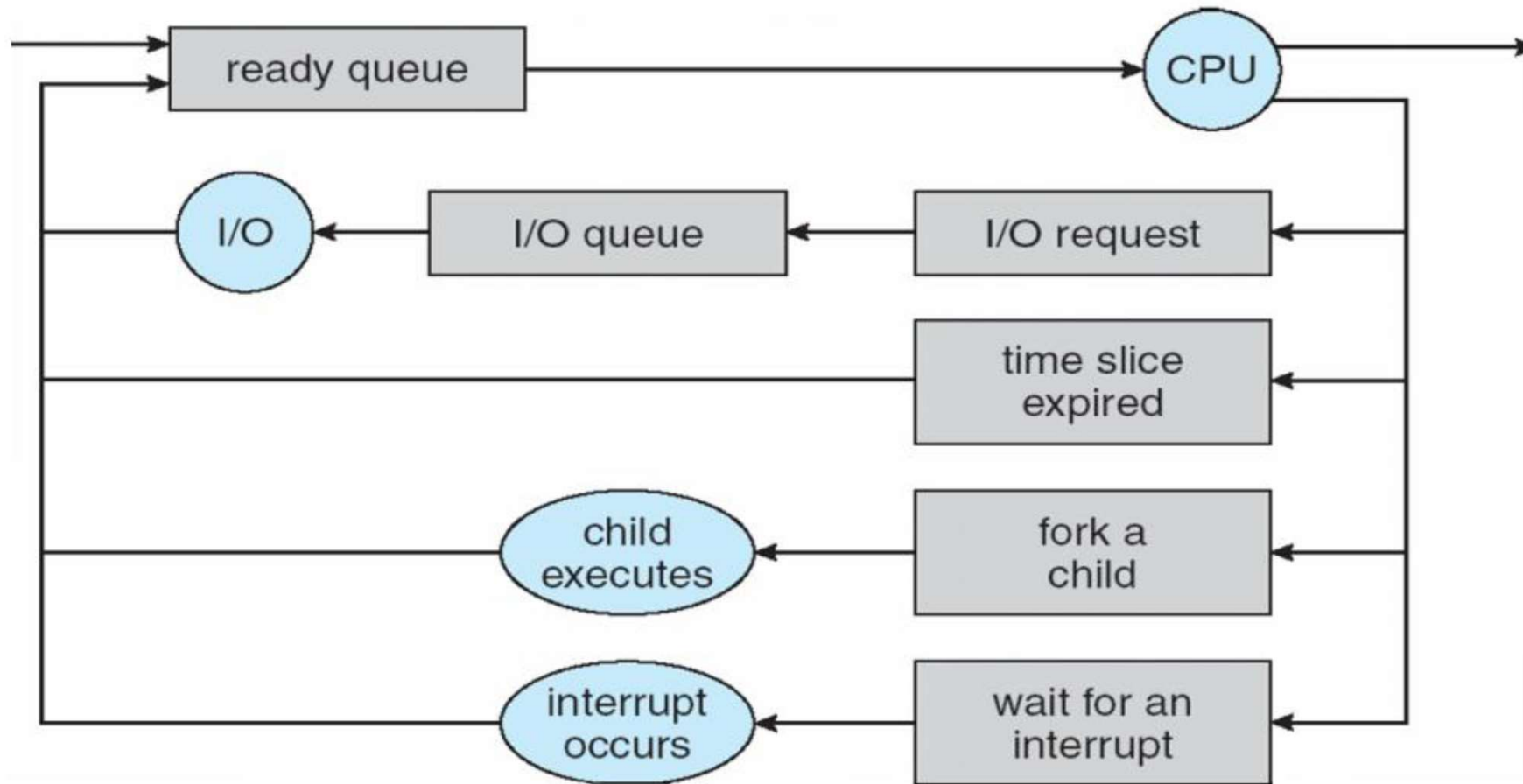
✓ تحديد الإجراءات التالية التي ستستخدم وحدة المعالجة بعد انتهائها من الإجراءات الحالية كما هو مبين في الشكل (2) .

✓ يجب أن يكون المجدول سريع جداً.

✓ إذا احتاجت إجراءات مورد أو دخل/خرج ستزال من وحدة المعالجة وتحول إلى حالة الانتظار وإدخال إجراءات أخرى من صف الانتظار إلى وحدة المعالجة.



يبين الشكل (1) أنواع المجدولات



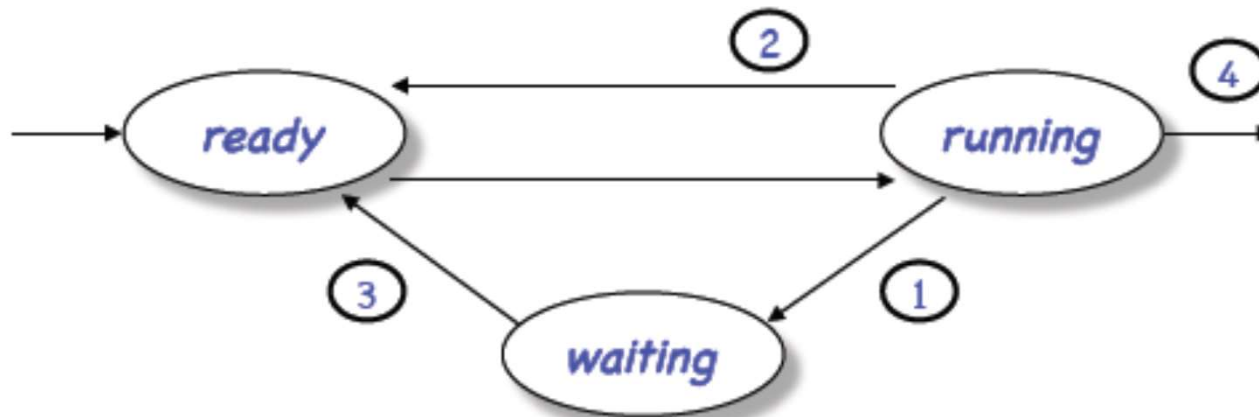
يبين الشكل (2) جدول وحدة المعالجة المركزية

• يتخذ المجدول قراراته عند ما تتحول الإجراءات من:

1. من حالة التنفيذ **running** إلى حالة الانتظار **waiting**.
2. من حالة التنفيذ **running** إلى حالة الجاهزية **ready**.
3. من حالة الانتظار **waiting** إلى حالة جاهزية **ready**.
4. منتهية **Terminates**.

• **في الحالتين 1 و 4** ستخرج الإجراءات إجبارياً إما لانتهائها أو لأنها تحتاج دخل أو خرج، في هذه الحالة لابد للمجدول من اختيار إجراءات بديلة، فليس لديه خيار آخر. في هذه الحال نقول أن المجدول **non-preemptive**.

"Who is going to use the CPU next?!"



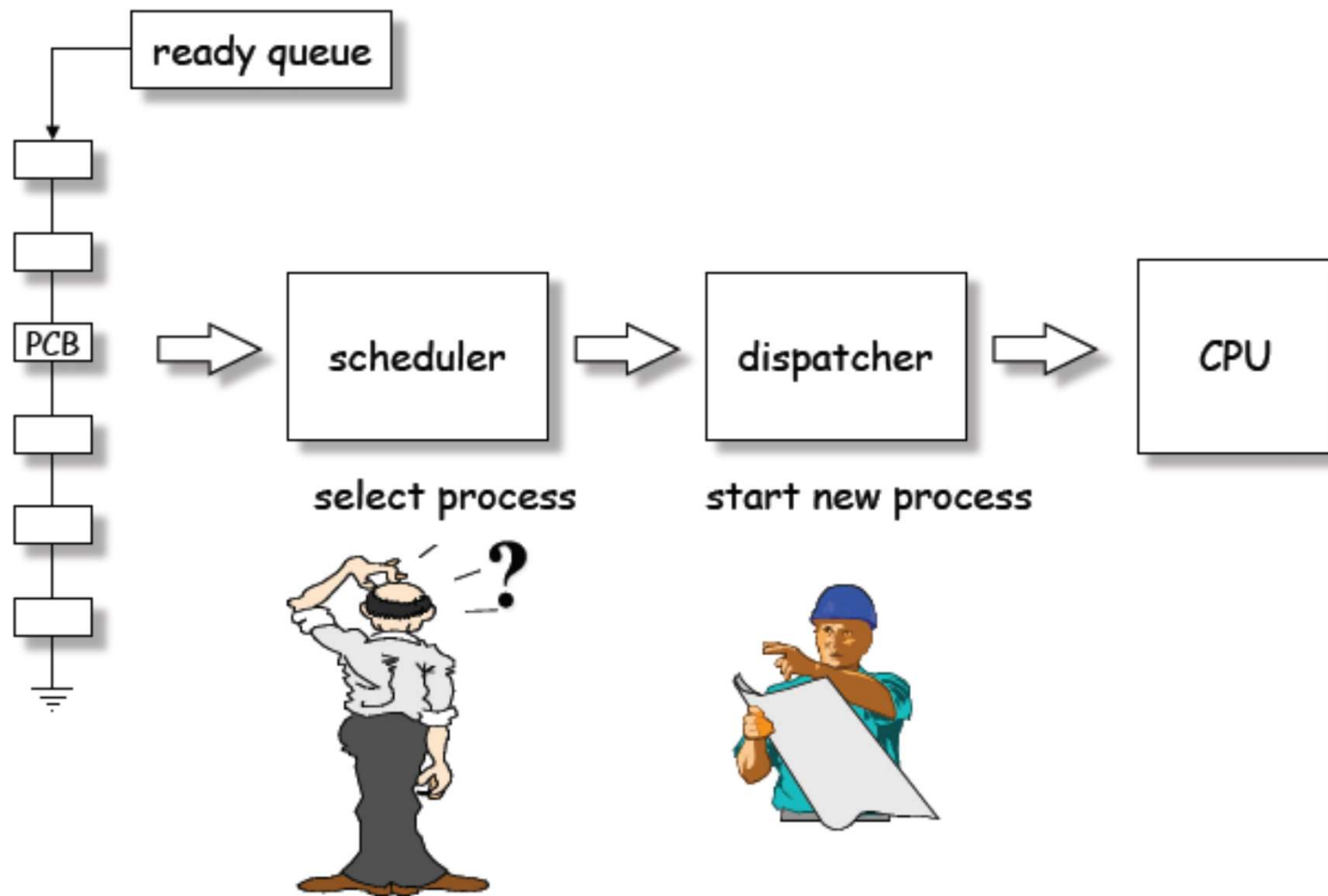
- **أما في الحالتين 4 و3** فإن الإجرائية تخرج اختياراً، وقد تتواصل، هنا للمجدول حق الاختيار في إخراجها أو لا، هنا نقول أن المجدول قابل للتوقف preemptive .

ملاحظة :

- الجدولة غير قابلة للتوقف non-preemptive scheduling : عند ما تدخل إجرائية المعالج وتبدأ التنفيذ فإنها لن تترك وحدة المعالجة أبداً إلا في إحدى الحالتين :
 1. **الاكتمال .**
 2. **التحول إلى حالة الانتظار .**

□ المرسل Dispatcher

- المرسل هو جزء من المجدول يقوم بإرسال الإجراءات التي يختارها المجدول قصير الأمد (مجول وحدة المعالجة) ومن وظائفه:
 - ✓تبديل السياق switching context : حفظ بيانات الإجرائية الموقفة وتحميل بيانات الإجرائية التي تم اختيارها لتعمل في وحدة المعالجة .
 - ✓التحول إلى وضع المستخدم user mode .
 - ✓القفز إلى الموقع المناسب في برنامج المستخدم لإعادة تشغيل البرنامج.
- المرسل لابد من أن يكون سريعاً جداً، لأنه سيعمل مع كل لإجرائية.
- **زمن تلبث الإرسال Dispatch latency** : هو الزمن الذي يستغرقه المرسل في توقيف إجرائية وتشغيل أخرى (تبديل إجرائية) .



تحسين أداء وحدة المعالجة

• **تعتبر وحدة المعالجة ال CPU** أهم الموارد التي تتزاحم عليها الاجرائيات، لذلك كان لا بد من إيجاد خوارزميات لترتيب وتنظيم عمليات الحصول على هذا المورد من قبل الاجرائيات ، ويسمى ذلك بجدولة وحدة المعالجة المركزية ويقوم بها مكون برمجي يسمى المجدول، ويسعى المجدول **لتحسين أداء وحدة المعالجة لتحقيق الأهداف التالية:**

1. **زيادة نسبة استثمار** وحدة المعالجة المركزية CPU utilization
2. **زيادة الإنتاجية** (عدد الاجرائيات التي يتم إنجازها خلال فترة زمنية محددة)
3. **التسريع بالاستجابة للإجرائيات :** أي جعل العمليات تبدأ بالحصول على أحد مخرجاتها في أسرع وقت ممكن .
4. **التقليل من زمن انتظار** الإجرائيات في الحصول على وحدة المعالجة .
5. **التقليل من زمن الاكتمال** أو الزمن الكلي للإجرائية وهو الفترة الزمنية الفاصلة بين لحظة وصول الإجرائية إلى رتل **الإجرائيات الجاهزة** ولحظة اكتمال تنفيذها (أي الانتهاء من التنفيذ) .



معايير الجدولة

- تتمتع خوارزميات الجدولة المختلفة لوحدات المعالجة بخصائص مختلفة، لذلك يجب اختيار الخوارزمية الأنسب للاستخدام في حالة معينة.
- المعايير المستخدمة للمقارنة بين الخوارزميات:
 - ✓ معدل استخدام وحدة المعالجة: يجب أن تكون وحدة المعالجة مشغولة قدر الإمكان لاستغلالها بالشكل الأمثل
 - ✓ معدل الإنتاجية (التدفق): عدد الإجراءات التي يتم تنفيذها في واحدة الزمن.
 - ✓ الزمن الكلي: هو الزمن اللازم لتنفيذ الإجراءات، أي الزمن بين طلب تنفيذ الإجراءات وإنائها ويساوي: $\text{الزمن الكلي} = \text{زمن التنفيذ} + \text{زمن الانتظار}$.
 - ✓ زمن الانتظار: هو الوقت الذي تقضيه إجراءات منتظرة في رتل الجاهزات (يساوي مجموع فترات الانتظار في رتل الجاهزات).
 - ✓ زمن الاستجابة: الزمن بين إرسال طلب وحدوث أول استجابة (الفترة الزمنية قبل حدوث الاستجابة).



خوارزمية الجدولة (القادم أولاً – يُخدم أولاً FCFS)

First-come First-served (FCFS)

- تعتبر هذه الخوارزمية من أبسط خوارزميات جدولة وحدة المعالجة، فالإجرائية التي تطلب وحدة المعالجة أولاً، هي التي تحصل عليها أولاً .
- تنفذ الإجراءات حسب ترتيب وصولها لوحدة المعالجة بواسطة المجدول قصير الأمد، حيث تنفذ أول عملية وصلت إلى صف الانتظار أولاً ثم التي وصلت بعدها ثانياً، وهكذا.
- **وهي خوارزمية بسيطة في عملها وواضحة.**
- من الناحية التقنية يمكن تحقيق هذه الخوارزمية من خلال رتل، فعند دخول إجرائية إلى رتل الجاهزيات، تربط كتلة التحكم بالإجرائية الخاصة بها إلى مؤخرة الرتل ، وعند تحرير وحدة المعالجة، يجري اسنادها إلى الإجرائية التي في مقدمة الرتل.
- الخوارزمية مناسبة للإجرائيات الكبيرة عندما تجد فرصة للتنفيذ، وتسبب مشكلة للإجرائيات القصيرة إذا كانت خلف إجراءات كبيرة .

• سيئات هذه الخوارزمية:

- ✓ ما إن تعطى وحدة المعالجة إلى إجرائية ما حتى تحتكرها، ولا تحررها إلا عند انتهاء التنفيذ أو طلب عملية دخل/خرج (بينما من الضروري حصول الإجراءات على وحدة المعالجة خلال فترات زمنية نظامية).
- ✓ قد يحدث انتظار مجموعة من الإجراءات الصغيرة، خروج إجرائية كبيرة من وحدة المعالجة، وهذا ما يخفض من معدل استخدام وحدة المعالجة والتجهيزات.



مثال Example

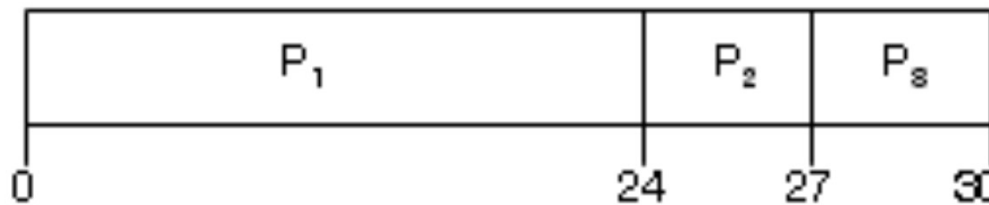
- الجدول التالي يوضح إجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية القادم أولاً يخدم أولاً .
- حيث يمثل العمود الأول اسم الإجراءات بينما العمود الثاني يمثل الزمن الذي تحتاجه كل إجراء لتكمل عملها داخل المعالج، والمطلوب :

الزمن	الإجرائية
24	P1
3	P2
3	P3

1. احسب متوسط زمن الانتظار waiting time .
2. احسب متوسط زمن الاكتمال completion time .
3. احسب متوسط زمن الانتظار إذا كان ترتيب الوصول عكسي (p2 ثم p3 ثم p1) وضح الفرق بين المتوسطين.

□ الحل :

إذا وصلت العمليات بالترتيب P1 ثم P2 ثم P3، وتم تخديمها بترتيب القادم أولاً يخدم أولاً، فإن المجدول سينفذ العمليات في وحدة المعالجة حسب ترتيبها، لنرسم مخطط غانت Gantt Chart التالي :



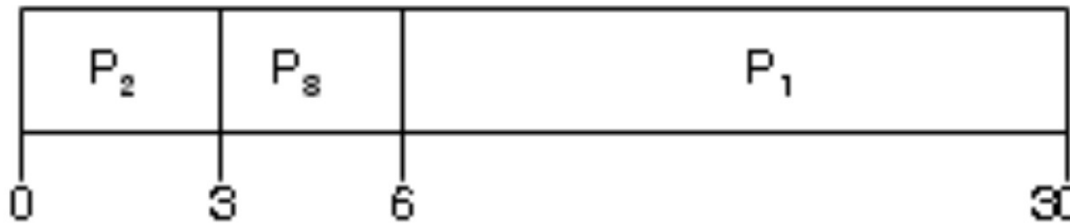
1. متوسط زمن الانتظار :

$$(0+24+27)/3 = 17 \text{ mS}$$

2. متوسط زمن الاكتمال :

$$(24 + 27 + 30) / 3 = 27 \text{ mS}$$

3. إذا وصلت الإجراءات بالترتيب العكسي فإن شكل الإجراءات داخل وحدة المعالجة سيكون كما في الشكل التالي لمخطط غانت :



ا. متوسط زمن الانتظار :

$$(0+3+6)/3 = 3 \text{ mS}$$

اا. متوسط زمن الاكتمال :

$$(3 + 6 + 30) / 3 = 13 \text{ mS}$$

• حيث نلاحظ أن متوسط زمن الانتظار إذا نفذت الإجراءات الكبيرة في النهاية أفضل من متوسط زمن الانتظار إذا تم تنفيذ الإجراءات الكبيرة أولاً.



خوارزمية الجدولة (الأقصر عملاً أولاً – SJF)

Shortest-Job-First (SJF)

- تربط هذه الخوارزمية بكل إجرائية دفعة الـ CPU التالية المتعلقة بها، فعندما تتحرر وحدة المعالجة تعطى للإجرائية ذات دفعة التنفيذ التالية الأصغر زمناً، وإذا وجدت إجرائيتان لهما طول دفعة واحد يجري الاختيار حسب FCFS .
- فإن المجدول سيختار الإجرائية التي تحتاج زمن أقل أولاً.
- إن هذه الخوارزمية تحقق أصغر زمن انتظار وسطي، لمجموعة من الإجرائيات المعطاة، فمن خلال تمرير إجرائية قبل إجرائية أخرى أطول منها، يقصر زمن انتظار الإجرائية القصيرة، بدلاً من تطويل زمن انتظار الإجرائية الطويلة .
- لكن الصعوبة الحقيقية في هذه الخوارزمية هي معرفة مدة الطلب التالي لوحدة المعالجة، أي معرفة طول الوقت الذي ستحتاجه العملية داخل وحدة المعالجة مسبقاً، فغالباً لا يعلم المجدول ذلك مما يصعب معرفة الإجرائية الأقصر.

- سيكون الأداء ضعيف في حالة وصول إجراءات قصيرة بعد بدء تنفيذ إجراءات طويلة.
 - كذلك قد يحدث حرمان للإجراءات الطويلة (تحرّم من التنفيذ لوجود إجراءات قصيرة).
 - يقدم خدمة جيدة للإجراءات القصيرة.
 - تفشل الخوارزمية مع الإجراءات التي كانت سابقاً قصيرة لكنها أصبحت الآن طويلة .
- تنقسم الخوارزمية إلى نوعين هما :

1. غير قابلة للتوقف Non-preemptive : فإذا وردت إجراءات جديدة إلى رتل الجاهزيات، وكانت دفعة الـ CPU التالية لها أقصر مما بقي للإجراءات التي يجري تنفيذها ، تتابع الإجراءات تنفيذها دون توقف، حتى انتهاء الفترة المخصصة لها.
2. قابلة للتوقف Preemptive : إذا وصلت إجراءات جديدة إلى صف الانتظار وكان زمنها أقصر من الزمن المتبقي للإجراءات التي تنفذ حالياً بالمعالج سيقوم المجدول بإيقاف الإجراءات التي تنفذ، ويتم إسناد وحدة المعالجة إلى الإجراءات الجديدة التي وصلت حديثاً .

مثال 1 Example 1

- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية SJF ، وغير قابلة للتوقيف والتي وصلت في نفس الوقت (أي دون اعتبار لزمان الوصول).
- والمطلوب :

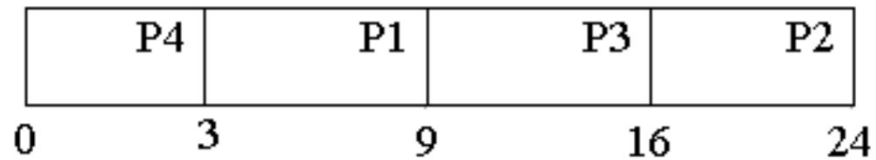
1. ارسم مخطط غانت

2. احسب متوسط زمن الانتظار waiting time .

3. احسب متوسط زمن الاكتمال completion time .

الإجرائية	الزمن Burst time
P1	6
P2	8
P3	7
P4	3

□ الحل :



1. متوسط زمن الانتظار :

$$(0 + 3 + 9 + 16) / 4 = 7 \text{ mS}$$

2. متوسط زمن الاكتمال :

$$(3 + 9 + 16 + 24) / 4 = 13 \text{ mS}$$



مثال 2 Example 2

- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية SJF ، والقابلة للتوقيف والتي وصلت في أزمان مختلفة .

- والمطلوب :

الإجرائية	الزمن الوصول	الزمن Burst time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

1. ارسم مخطط غانت
2. احسب متوسط زمن الانتظار waiting time .
3. احسب متوسط زمن الاكتمال completion time .

□ الحل :

P1	P2	P4	P1	P3	
0	1	5	10	17	26

✓ زمن انتظار الإجرائية = زمن الانتظار الكلي - زمن الوصول

✓ زمن انتظار P1

$$(0 + 9) - 0 = 9$$

✓ زمن انتظار P2

$$1 - 1 = 0$$

✓ زمن انتظار P3

$$17 - 2 = 15$$

✓ زمن انتظار P4

$$5 - 3 = 2$$

1. متوسط زمن الانتظار :

$$(9 + 0 + 15 + 2) / 4 = 6.75 \text{ mS}$$

2. متوسط زمن الاكتمال :

$$(17 + 5 + 26 + 10) / 4 = 14.5 \text{ mS}$$



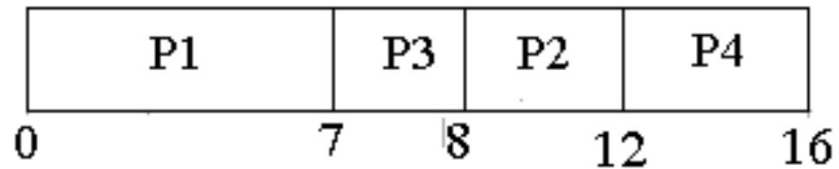
مثال 3 Example 3

- مثال آخر على إجراءات لم تصل في وقت واحد باستخدام خوارزمية الأقصر أولاً غير قابلة للتوقف .
- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية SJF ، وغير القابلة للتوقيف والتي وصلت في أزمان مختلفة .
- والمطلوب :

العملية	الزمن الوصول	الزمن Burst time
P1	0	7
P2	2	4
P3	4	1
P4	5	4

1. ارسم مخطط غانت
2. احسب متوسط زمن الانتظار waiting time .
3. احسب متوسط زمن الاكتمال completion time .

□ الحل :



✓ زمن انتظار الإجرائية = زمن الانتظار الكلي - زمن الوصول

✓ زمن انتظار P1

$$0 - 0 = 10$$

✓ زمن انتظار P2

$$8 - 2 = 6$$

✓ زمن انتظار P3

$$7 - 4 = 3$$

✓ زمن انتظار P4

$$12 - 5 = 7$$

1. متوسط زمن الانتظار :

$$(0 + 6 + 3 + 7) / 4 = 4 \text{ mS}$$

2. متوسط زمن الاكتمال :

$$(7 + 8 + 12 + 16) / 4 = 10.75 \text{ mS}$$



مثال 4 Example 4

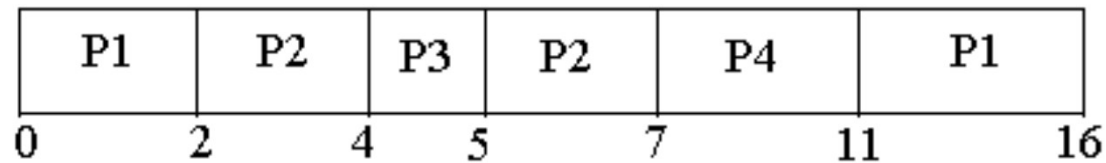
- هنا سنحل المثال (3) السابق بطريقة الأقصر أولاً القابلة للتوقف .
- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية SJF ، القابلة للتوقيف والتي وصلت في أزمان مختلفة .

- والمطلوب :

العملية	الزمن الوصول	الزمن Burst time
P1	0	7
P2	2	4
P3	4	1
P4	5	4

1. ارسم مخطط غانت
2. احسب متوسط زمن الانتظار waiting time .
3. احسب متوسط زمن الاكتمال completion time

□ الحل :



✓ زمن انتظار الإجرائية = زمن الانتظار الكلي - زمن الوصول

✓ زمن انتظار P1

$$(0 + (11 - 2)) - 0 = 9$$

✓ زمن انتظار P2

$$(2 + (5 - 2)) - 2 = 1$$

✓ زمن انتظار P3

$$4 - 4 = 0$$

✓ زمن انتظار P4

$$7 - 5 = 2$$

1. متوسط زمن الانتظار :

$$(9 + 1 + 0 + 2) / 4 = 3 \text{ mS}$$

2. متوسط زمن الاكتمال :

$$(16 + 7 + 5 + 11) / 4 = 9.75 \text{ mS}$$



خوارزمية الجدولة (حسب الأولوية)

Priority Scheduling

- **في خوارزمية الجدولة وفق الأولوية**، تحدد أولوية بكل إجرائية، ويجري اختيار الإجرائية ذات الأولوية العليا.
- **نلاحظ أن خوارزمية SJF هي حالة خاصة من خوارزمية الجدولة وفق الأولوية** مع اعتبار أولوية الإجرائية هي مقلوب زمن دفعة الـ CPU التالية.
- **يمكن أن تعرف الأولويات إما داخلياً أو خارجياً**، حيث تستخدم الأولويات المعرفة داخلياً مقادير قابلة للقياس (مثل الحدود الزمنية، متطلبات الذاكرة، عدد الملفات المفتوحة) من أجل حساب الأولوية، أما الأولويات الخارجية فيجري تعيينها بواسطة معايير خارجية على نظام التشغيل (مثل أهمية الإجرائية، المبلغ المدفوع).
- **كل إجرائية لديها رقم مرفق معها**، هذا الرقم يحدد أهمية الإجرائية، حيث سيتم تنفيذ الإجرائية ذات الأولوية الأعلى (أقل رقم).

- فإذا كنا نعتبر أن الرقم الأقل يمثل الأولوية الأعلى، فهذا يعني أنه إذا كان لدي إجراءات برقم الأولوية 5 وهناك إجراءات برقم الأولوية 7، فسينفذ المجدول الإجرائية ذات الرقم 5 قبل الإجرائية ذات الرقم 7، لأن أولويتها أعلى. **هناك نوعين هما:**

1. **قابلة للتوقف Preemptive**

2. **غير قابلة للتوقف non-preemptive**

- **المشكلة في خوارزمية الأولوية هو الحرمان Starvation**، حيث لن تجد الإجراءات ذات الأولوية الدنيا فرصة للتنفيذ داخل المعالج طالما أن هناك إجراءات ذات أولوية أعلى منها، حيث سينفذ المعالج الإجراءات ذات الأولوية العليا و كلما قرب دور الإجراءات ذات الأولوية الدنيا للتنفيذ داخل المعالج قد تصل إجراءات أخرى لها أولوية أعلى منها فتحرمها من استخدام المعالج.

- **حل مشكلة الحرمان هو النضوج Aging** ، وذلك بزيادة أولوية الإجراءات منخفضة الأولوية تدريجياً، مع استمرار بقائها بانتظار .



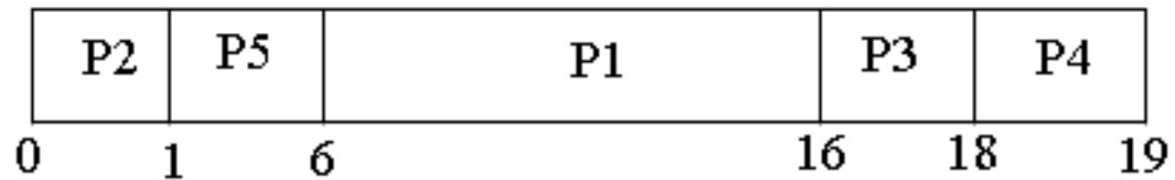
مثال 1 Example 1

- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية الأولوية ، غير القابلة للتوقيف والتي وصلت في وقت واحد.
- والمطلوب :

العملية	رقم الأولوية	الزمن
P1	3	10
P2	1	1
P3	3	2
P4	4	1
P5	2	5

1. ارسم مخطط غانت
2. احسب متوسط زمن الانتظار waiting time .
3. احسب متوسط زمن الاكتمال completion time

□ الحل :



1. متوسط زمن الانتظار :

$$(6 + 0 + 16 + 18 + 1) / 5 = 8.2 \text{ mS}$$

2. متوسط زمن الاكتمال :

$$(16 + 1 + 18 + 19 + 6) / 5 = 12 \text{ mS}$$



مثال 2 Example 2

- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية الأولوية ، القابلة للتوقيف والتي وصلت في أوقات مختلفة .

- والمطلوب :

1. ارسم مخطط غانت
2. احسب متوسط زمن الانتظار waiting time .
3. احسب متوسط زمن الاكتمال completion time

العملية	رقم الأولوية	الزمن	زمن الوصول
P1	3	10	0
P2	1	1	2
P3	3	2	4
P4	4	1	7
P5	2	5	9

العملية	الزمن Burst time	زمن الوصول	رقم الأولوية
P1	10	0	3
P2	1	2	1
P3	2	4	3
P4	1	7	4
P5	5	9	2

خوارزمية الجدولة (التقسيم الزمني -الدائرية RR)

Round Robin (RR) Scheduling

- يتم التعامل في هذه الخوارزمية مع رتل الجاهزيات بطريقة دوارة، وتنتقل وحدة المعالجة بين إجراءات الرتل، بحيث أن لكل إجرائية مدة زمنية لا تتجاوز حصة زمنية محددة مسبقاً.
- تبدأ وحدة المعالجة بتنفيذ الإجرائية الأولى في رتل الجاهزيات، ومن ثم إذا كان للإجرائية دفعة وحدة معالجة أقصر من حصة زمنية واحدة، عندها تتخلى عن وحدة المعالجة عند انتهاء دفعة التنفيذ، وتعود إلى مؤخرة رتل الجاهزيات.
- أما إذا كانت دفعة وحدة المعالجة أطول من حصة زمنية واحدة، عندها مع اكتمال الحصة الزمنية، تجري مقاطعة الإجرائية، فيتوقف تنفيذها ويجري تبديل السياق مع الإجرائية الأولى في رتل الجاهزيات، بينما تعود هذه الإجرائية إلى مؤخرة الرتل.
- يعتمد أداء هذه الخوارزمية على حجم الحصة الزمنية، فإذا كانت كبيرة جداً تتحول إلى FCFS، وإذا كانت صغيرة جداً يصبح هناك تشارك كامل بالمعالج.

- هذا النوع من الخوارزميات صمم خصيصاً للنظم التي تستخدم المشاركة بالزمن (time sharing) ، فكل إجرائية حصة زمنية داخل المعالج تخرج بعدها من المعالج لتدخل الإجرائية التي تليها فتأخذ نفس الحصة الزمنية التي أخذتها الإجرائية الأولى، وهكذا يتم تقسيم زمن المعالج بحيث تأخذ كل إجرائية حصة بالمعالج، ثم تبدأ من جديد.
- فالخوارزمية تعمل بطريقة تشبه خوارزمية القادم أولاً يخدم أولاً، لكن مع إمكانية توقيف كل إجرائية إذا اكتملت الفترة الزمنية المقررة لها داخل المعالج ، حيث تحدد الحصة الزمنية (quantum or time slice) بقيمة صغيرة تتراوح بين 10 إلى 100 ملي ثانية.
- يقوم المجدول بتقسيم زمن المعالج بين الإجرائيات فتعطي كل إجرائية حصة زمنية محددة داخل المعالج، ويتم تحديد الفترة الزمنية لكل الإجرائيات فيما يسمى Quantum ، فإذا كانت قيمة Quantum هي 10 ، فهذا يعني أن المجدول سيسمح لكل عملية أن تنفذ في المعالج لمدة 10 ملي ثانية، ثم تخرج (ترجع إلى نهاية صف الانتظار)، وتدخل الإجرائية التي تليها، لتأخذ 10 ملي ثانية وهكذا.

□ ملاحظات :

- إذا كان الزمن الذي تحتاجه العملية أقل من الحصة المقررة، فستأخذ العملية ما تحتاجه من الحصة وتخرج لتدخل العملية التي تليها.
- **مثلا** إذا كانت العملية تحتاج 3 ملي ثانية وكانت الحصة المقررة للعملية هي 10 ملي ثانية، فستمكث العملية بالمعالج 3 ملي ثانية وليست 10 ملي ثانية .
- لا توجد عمليات غير قابلة للتوقف في هذه الخوارزمية فكل العمليات ستجبر على التوقف عند اكتمال حصتها بالمعالج.



مثال 1 Example 1

- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية الجدولة الدائرية وحددت لها حصة زمنية Quantum تساوي 4 ms .
- والمطلوب :

الزمن Burst time	الإجرائية
24	P1
3	P2
3	P3

1. ارسم مخطط غانت
2. احسب متوسط زمن الانتظار waiting time .
3. احسب متوسط زمن الاكتمال completion time

□ الحل :

P1	P2	P3	P1	P1	P1	P1	P1	
0	4	7	10	14	18	22	26	30

✓ زمن انتظار P1 : $0 + (10 - 4) = 6$

✓ $P2 = 4$ زمن انتظار

✓ $p3 = 7$ زمن انتظار

1. متوسط زمن الانتظار

$$(6 + 4 + 7) / 3 = 5.67 \text{ mS}$$

2. متوسط زمن الاكتمال

$$(30 + 7 + 10) / 3 = 15.67 \text{ mS}$$



مثال 2 Example 2

- الجدول التالي يوضح الإجراءات في صف الانتظار ونريد تنفيذها حسب خوارزمية الجدولة الدائرية وحددت لها حصة زمنية Quantum تساوي 20 mS .
- والمطلوب :

1. ارسم مخطط غانت

2. احسب متوسط زمن الانتظار waiting time .

3. احسب متوسط زمن الاكتمال completion time

الزمن Burst time	الإجرائية
53	P1
17	P2
68	P3
24	P4

□ الحل :

P1	P2	P3	P4	P1	P3	P4	P1	P3	P3	
0	20	37	57	77	97	117	121	134	154	162

$$0 + (77 - 20) + (121 - 97) = 81$$

✓ زمن انتظار P1 :

$$p2 = 20 \quad \checkmark \text{ زمن انتظار}$$

$$37 + (97 - 57) + (134 - 117) = 94$$

✓ زمن انتظار P3 :

$$57 + (117 - 77) = 97$$

✓ زمن انتظار P4 :

• متوسط زمن الانتظار :

$$(81 + 20 + 94 + 97) / 4 = 73 \text{ mS}$$

• متوسط زمن الاكتمال :

$$(134 + 37 + 162 + 121) / 4 = 113.5 \text{ mS}$$