CPU Scheduling جدولة المعالجة

د. فادي تركاوي

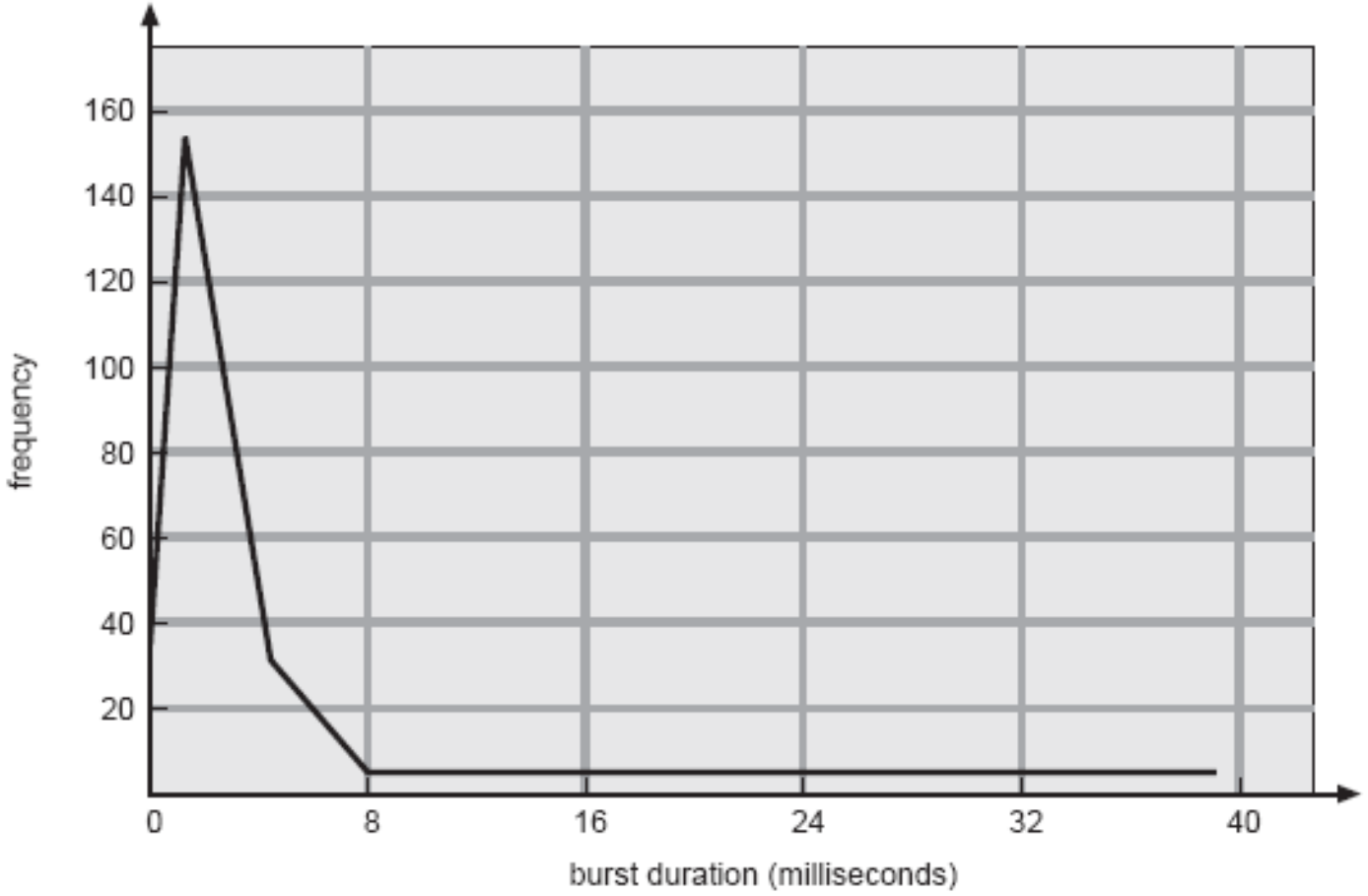
مجدول المعالج CPU Scheduler

- مجدول المعالج (و يدعى أحيانا بال Dispatcherأو Scheduler أو المجدول قصير الوقت):
  - يقوم بإختيار العمليات من صف العمليات الجاهزة و يقوم بتشغيله على المعالج.
  - يفترض بأن كل العمليات في الذاكرة و واحدة منها يتم تنفيذها من قبل المعالج.
  - من أهم العمليات في البيئات المتعددة البرمجية و الهدف هو زيادة إستخدام المعالج (الإستخدام الأمثل للمعالج).

سلوك تنفيذ العملية
Process Execution Behavior

- كما ذكرنا سابقاً المعالجات مستقلة و تتنافس من أجل الحصول على المصادر (compete for the resources).
- المعالجة يتم تنفيذها إما في دورة CPU أو في دورة I/O
  - عالج لفترة (على المعالج)
  - أتمم أعمال ال I/O
  - كرر هاتين العمليتين بشكل دائم
- بناءاً عليه العمليات لها نوعين آساسيين:
  - مرتبطة بالمعالج و هي العملية التي تعتمد في أكثر وقتها على المعالج و تكون مرتبطة بشكل قليل بالدخل و الخرج
  - مرتبطة الدخل و الخرج و هي العملية التي تعمل أكثر الأحيان دخل و خرج والقليل من عمليات الحوسبة على المعالج

CPU-Burst for a typical Process



مميزات جدولة المعالجة
Scheduling Criteria

- CPU Utilization MAX الإستخدام الأمثل للمعالج و تضمن هذه الميزة كون المعالج مشغول أكثر وقته
- Throughput MAX لضمان كون المعالج يعالج عدد من المعالجات ضمن وحدة الوقت
- Timearound Time MIN هو الوقت المطلوب لتنفذ معالجة محددة
- Waiting Time MIN كمية الوقت الذي تبقى فيه المعالجة على صفوف الإنتظار
- Respone Time MIN كمية الوقت الذي تأخذه معالجة منذ لحظة طلبها المعالجة إلى اللحظة التي يتم فيها الإستجابة ببدء المعالجة

CPU Utilization

استغلال كل وقت وحدة المعالجة المركزية(CPU) في تنفيذ العمليات، أي أن تكون وحده المعالجة المركزية مشغولة بقدر الإمكان ليتم استغلالها الاستغلال الأمثل

وعادة يمثل بنسبة مئوية يمكن حسابها باستخدام أحد القانونين :

أ- نسبة استغلال المعالج = (وقت المعالج الكلي – الوقت الذي قضاه فارغا)/ (وقت المعالج الكلي) * 100

ب- نسبة استغلال المعالج = (وقت التنفيذ للعمليات الكلي)/ (وقت العمليات الكلي + الوقت المستغرق في تبديل المحتوى)* 100

ومما يؤثر على هذا العامل حقيقة هو عدد مرات التبديل التي تتم فكلما زاد هذا العدد كلما قل استغلال المعالج وهذا منطقي جدا أليس كذلك ؟ !

ما نريده نحن ونصبو إليه هو أعلى استغلال ممكن للمعالج إذ أننا لا نريد الآن شغل وقت فراغ المعالج فقط بل شغله بما ينفع أيضا فلا نريد تضييع وقته في عمليات التبديل.

الإنتاجية Throughput

عدد العمليات التي يتم تنفيذها في الوحدة الزمنية الواحدة

Turnaround Time

الوقت اللازم لإنهاء تنفيذ عمليه محدده، و هو مجموع الفترات التي أمضاها في

أ- الانتظار قبل الدخول إلى الذاكرة

ب- الانتظار في طابور الاستعداد (ready queue)

ت- التنفيذ على وحدة المعالجة المركزية

ث- تنفيذ عمليات الإدخال والإخراج

Waiting Time

هو الوقت الذي تستغرقه العملية في الانتظار داخل مصفوفة الانتظار(ready queue) قبل دخولها إلى وحدة المعالجة المركزية (CPU)

Response Time

الوقت من آمر تنفيذ العملية حتى ظهور أول نتيجة لها، يستخدم هذا عادة في الأنظمة التفاعليـــة (interactive) system التي يكون بها المستخدم طرفا.

وما نطمح إليه هو الحصول عليه هو استغلال للمعالج بأقصى حد ممكن (maximum CPU utilization) وأعلى نسبة من العمليات التي تكتمل في وحدة زمنية(Throughput)

كما نرغب في الحصول على أقل وقت انتظار(Waiting time) وأقل زمن استجابة (Response time) وأقل زمن دوري للعملية (Turnaround time)

يأتي أوّلاً ينفّذ أوّلاً
First-Come-First-Served (FCFS)

- أسماء أخرى First-In-First-Out FIFO أو Run-Until-Done
- Burst Time هو الوقت الذي تحتاجه المعالجة للتنفيذ
- طريقة العمل:
  - أختار عملية من الصف الجاهز (Ready Queue) و ذلك **حسب ترتيب الوصول وبعدها قم تشغيل المعالجة بشكل غير قابل للتوقيف حتى نهاية المعالجة**

يأتي أوّلاً ينفّذ أوّلاً
First-Come-First-Served (FCFS)

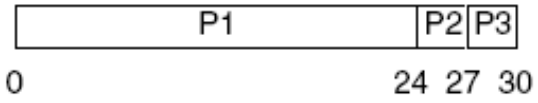
فلسفة هذه الطريقة تعتمد على من يصل أولاً من العمليات هو من يدخل أولاً إلى وحدة المعالجة المركزية (CPU) ،وهي تعتبر غير قابلة للإجهاض (Non-preemptive) أي أن هذه العملية لا تخرج من وحدة المعالجة المركزية إلا بعد انتهائها من التنفيذ ولا يتدخل لب النظام (kernel) في إجهاض العملية .

خصائص هذه الطريقة

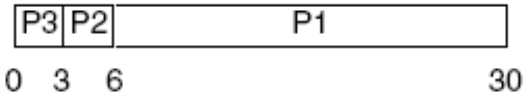
1. أبسط خوارزميات جدولة المهام على الإطلاق .
2. معدل وقت الانتظار فيها ليس بالضرورة أن يكون الأقصر .
3. الأداء (performance) هنا يعيبه عدم الاستغلال الأمثل للمعالج وهذا سببه (convoy effect)ونعني به أن هناك عمليات قصيرة ويمكن إنجازها بسرعة ولكنها رغم ذلك تضطر للانتظار بسبب وجود عمليات أطول منها في طور التنفيذ .
4. يعتبر غير ملائم أبداً للاستخدام في الأنظمة التفاعلية (interactive system) وهذا العيب ناشئ وبشكل واضح عن كون هذه الخوارزمية غير قابلة للإجهاض (non preemptive scheduling)

- الوقت الوسطي للإنتظار هو:
(0 + 24 + 27) / 3 =17
- الوقت الوسطي للإنتظار في حال تغيير تسلسل الوصول هو:
(0 + 3 + 6) / 3 = 3

Process (Arrival Order)	P1	P2	P3
Burst Time	24	3	3
Arrival Time	0	0	0



Process (Arrival Order)	P3	P2	P1
Burst Time	3	3	24
Arrival Time	0	0	0



تقييم FCFS

- Non-preemptive
- Fairness يظلم المعالجات القصيرة بسبب الوقت الطويل الذي تستهلكه المعالجات الطويلة
- Starvation التجويع غير ممكن هنا

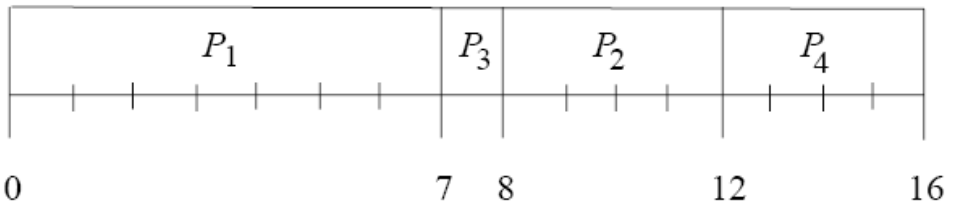
جدولةShortest-Job-First Scheduling
العملية ذات الوقت الأقصر أوّلا

- تخصص كل عملية بطول التنفيذ الذي تحتاجه العملية من المعالج. تستخدم هذه الأطوال لجدولة العملية ذات الوقت الأقصر أوّلاً.
- لها نوعان:
- Non-Preemptive في حال تم أختيار المعالجة فإنها سوف تتم للنهاية
- Preemptive في حال وصول معالجة أقصر من المعالجة التي تنفذ حالياً ففي هذه الحالة يتم تنفيذ المعالجة الأقصر و توقف المعالجة الحالية
- SJF يعتبر مثالي من ناحية زمن الإنتظار القصير من أجل مجموعة من العمليات

جدولة الوقت الأقصر أوّلاً
Shortest-Job-First

- يخصّص هنا لك معالجة الوقت الزمني
- SJF (non -preemptive)
- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

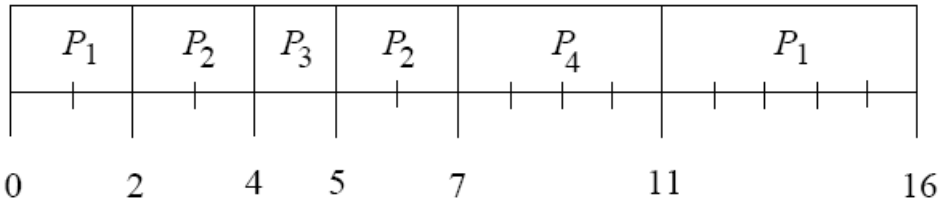
	Arrival Time	Time Burst
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4



SJF Preemptive

- Preemptive SJF
- Average waiting time = (9 + 1 + 0 + 2)/4 = 3

	Arrival Time	Time Burst
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4



تقييم جدولة العمل القصير أولا
SJF Evaluation

- في حالة Non-preemptive
- Response time جيد في حالة المعالجات القصيرة
  - المعالجات الطويلة ربما تنتظر وقت طويل في حال وجود عدد كبير من المعالجات القصيرة
- Throughputعالي
- Fairness تؤثر سلبا على المعالجات الطويلة
- Starvation ممكن حصوله في حال المعالجات الطويلة

الجدولة بالأفضلية Priority Scheduling

- يخصص هنا لكل عملية رقم يمثل أفضلية هذه المعالجة على العمليات الأخرى
- هنا يتم إختيار العملية ذات الأفضلية الأعلى لكي يتم تنفيذها
- أنواعه:
  - Preemptive قابل للإجهاض
  - Non-Preemptive غير قابل للإجهاض
- SJF هو يعتبر نوع من الجدولة بالإفضلية (الوقت)

تقييم الجدولة بالإفضلية و مبدأ التقادم

- هنا قد نواجه مشكلة Starvation التجويع بسبب أن المعالجات ذات الأفضلية القليلة قد يتم تأخيرها
- لحل هذه المشكلة نتبع مبدأ Aging الذي يعني زيادة الأفضلية لكل عملية و في كل مرحلة تنفيذ

التداور Round Robin

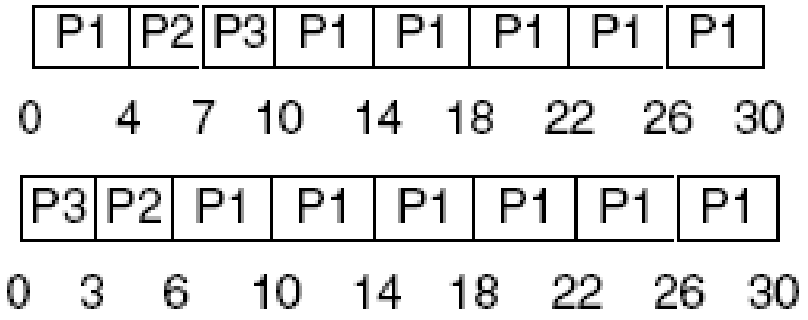
- في هذه النمط من الجدولة تحصل كل عملية على جزء صغير من وقت المعالج (شريحة زمنية Time slice)عادة بين (١٠-١٠٠) ميلي ثانية و بعد إنتهاء هذا الوقت يتم إجهاض (Preempt) العملية وإضافتها للصف الجاهز

- بعدها يتم تنفيذ العملية التالية على المعالج

وقت الإنتظار الوسطي

	Arrival Time	Time Burst
P1	0	٢٤
P2	0	٣
P3	0	٣

- Average Waiting Time= (6+4+7) / 3
  ٥,٦٦
- في حال عكس التنفيذ P3 ثم P2 ثم P1 يكون الزمن الوسطي:
- (0 + 3 + 6) / 3 = 3



تقييم التداور في المعالجة

- Preemptive عند إنتهاء الشريحة الزمنية المعطاة للعملية
- Response Time جيد في حال المعالجات القصيرة
- Throughput يعتمد على الشريحة الزمنية

الجدولة متعددة مستويات الصفوف
Multilevel Queue Scheduling

- العمليات المتنوعة تحتاج لجدولة متنوعة
- و يتم هنا تخصيص عدد من الصفوف الجاهزة Multilevel Ready Queuesو تخصيص كل صف بأفضلية مختلفة
- أختر العملية من الصف الملئ بالعمليات ذات الأفضلية الأعلى
- كل صف من هذه الصفوف قد يتم جدولته إما Preemptively أو Nonpreemptively على سبيل المثال:
  - ٨٠% من زمن المعالج لعمليات ال Foreground بإستخدام RR إستجابة أفضل
  - 20%من زمن المعالج لعمليات ال Background بإستخدام FCFS إستجابة أقل لكن أبسط من حيث التنفيذ