

Chapter 6: CPU Scheduling

الاستخدام الأقصى لوحدة المعالجة المركزية (CPU Utilization)

- يتحقق من خلال البرمجة المتعددة (Multiprogramming)، حيث تعمل عدة برامج في وقت واحد. ويكون فيه استخدام لموارد الجهاز بشكل فعال ويضمن أن النظام يعمل بكفاءة عالية..

1. CPU Burst:

- هو الفترة الزمنية التي تقضيها العملية في تنفيذ التعليمات على وحدة المعالجة المركزية (CPU) دون التوقف.

2. I/O Burst:

- هو الفترة الزمنية التي تقضيها العملية في انتظار إدخال أو إخراج البيانات، مثل قراءة البيانات من القرص الصلب أو الكتابة إليها. حيث أن خلال هذه الفترة، تكون العملية غير نشطة على وحدة المعالجة المركزية، حيث تنتظر العملية إتمام عملية الإدخال/الإخراج.

ملاحظة:

- يتبع تنفيذ وحدة المعالجة المركزية (CPU burst) عادةً تنفيذ إدخال/إخراج (I/O burst).

جدولة وحدة المعالجة المركزية (CPU Scheduler)

1. المحدد الزمني لوحدة المعالجة المركزية:

- يقوم المحدد القصير المدى (Short-term scheduler) باختيار العمليات من قائمة الانتظار الجاهزة (ready queue) ويوزع وحدة المعالجة المركزية على واحدة منها.

2. قرارات جدولة وحدة المعالجة المركزية:

1. الانتقال من حالة التشغيل إلى حالة الانتظار.
2. الانتقال من حالة التشغيل إلى حالة الجاهزية (عند حدوث انقطاع).
3. الانتقال من حالة الانتظار إلى حالة الجاهزية.
4. إنهاء العملية.

3. الجدولة غير المسبوقة والمسبوقة (preemptive) (non-preemptive)

- الجدولة في الحالتين 1 و 4 تعتبر غير مسبوقة (non-preemptive)، حيث لا يمكن أخذ وحدة المعالجة المركزية من العملية إلا إذا انتهت أو انتقلت إلى حالة الانتظار.

- جميع حالات الجدولة الأخرى 2 و 3 تعتبر مسبوقة (preemptive)، حيث يمكن أخذ وحدة المعالجة المركزية من العملية.

وحدة التحكم (Dispatcher)

- تقوم وحدة التحكم (Dispatcher module) بإعطاء السيطرة على وحدة المعالجة المركزية (CPU) للعملية التي يختارها المجدول القصير الأمد (short-term scheduler). يتضمن ذلك:

1. تغيير السياق (switching context).
2. التحويل إلى وضع المستخدم (switching to user mode).
3. الانتقال إلى الموقع المناسب في البرنامج المستخدم لإعادة تشغيل ذلك البرنامج.

- **زمن التوزيع (Dispatch latency):** هو الوقت الذي تستغرقه وحدة التحكم لإيقاف عملية واحدة وبدء تشغيل عملية أخرى. (أي الانتقال من عملية إلى أخرى)

معايير الجدولة (Scheduling Criteria)

1. الإنتاجية (Throughput): عدد العمليات التي تكتمل خلال وحدة زمنية.
2. زمن التحويل (Turnaround time): مقدار الوقت اللازم لتنفيذ عملية معينة من البداية حتى الانتهاء.
3. زمن الانتظار (Waiting time): مقدار الوقت الذي قضته عملية في الانتظار في قائمة الانتظار الجاهزة.
4. زمن الاستجابة (Response time): مقدار الوقت الذي يستغرقه الرد على طلب معين حتى يتم إنتاج الاستجابة الأولى، وليس المخرجات (في بيئة المشاركة الزمنية).

معايير تحسين خوارزميات الجدولة (Scheduling Algorithm Optimization Criteria)

1. زيادة استخدام وحدة المعالجة المركزية (Max CPU utilization). كلما زادت كان ادا الجهاز عالي.
2. زيادة الإنتاجية (Max throughput). كلما زادت كان ادا الجهاز عالي.
3. تقليل زمن التحويل (Min turnaround time). كلما قلت كان ادا الجهاز عالي.
4. تقليل زمن الانتظار (Min waiting time). كلما قلت كان ادا الجهاز عالي.
5. تقليل زمن الاستجابة (Min response time). كلما قلت كان ادا الجهاز عالي.

أولاً: جدولة FCFS (First-Come, First-Served Scheduling)

تنفذ فيه العمليات حسب ترتيب وصولها.

العمليات ووقت التنفيذ burst:

العملية Process	وقت التنفيذ Arrival Burst	وقت الانتظار
P1	24	0 - 24
P2	3	24 - 27
P3	3	27 - 30

ترتيب الوصول: P1، P2، P3.

حسابات وقت الانتظار:

$$-P1: 24 - 24 = 0$$

$$-P2: 27 - 3 = 24$$

$$-P3: 30 - 3 = 27$$

$$\text{متوسط وقت الانتظار} = (27 + 24 + 0) / 3 = 17$$

عيوبها

١. عندما تتواجد عمليات قصيرة خلف عملية طويلة، يؤدي ذلك إلى زيادة وقت الانتظار للعمليات القصيرة.

نوع الجدولة:

تعتبر هذه الحالة غير مسبقة (Non-preemptive): أي لا يمكن أخذ وحدة المعالجة المركزية من العملية إلا إذا انتهت أو انتقلت إلى حالة الانتظار

ثانياً جدولاً أقصر وقت أولاً (SJF - Shortest-Job-First)

- جدول SJF تعتمد على ربط كل عملية بطول التنفيذ التالي لوحدة المعالجة المركزية (CPU burst). حيث تُعطى الأولوية للعمليات الأقصر زمناً.

- تعتبر SJF مثالية، حيث تعطي أقل متوسط وقت انتظار لمجموعة معينة من العمليات. التحديات.

عيوبها

- يجب معرفة طول الطلب (العملية) التالي لوحدة المعالجة المركزية. أي كم تحتاج وقت لتنفيذها -ملاحظة: يمكن تقدير الطول بناءً على الانفجارات السابقة.

١. نوع الجدولة:

preemptive تُسمى Shortest-Remaining-Time-First، حيث يمكن إيقاف عملية في حالة وجود عملية أخرى ذات وقت تنفيذ أقصر.

٢. Non-preemptive في حال احتاجت العملية لوحدة الإدخال والإخراج

العملية Process	وقت التنفيذ Arrival Time	وقت الانتظار
P1	6	3 : 9
P2	8	16 : 24
P3	7	9 : 16
P4	3	0 : 3

ترتيب العمليات حسب اقل وقت تنفيذ : $p_2 - p_3 - p_1 - p_4$

وقت الانتظار لكل عملية

1. $P_1: 9 - 6 = 3$
2. $P_2: 24 - 8 = 16$
3. $P_3: 16 - 7 = 9$
4. $P_4: 3 - 3 = 0$

متوسط وقت الانتظار = $4 / (0 + 9 + 16 + 3) = 7$

ثالثاً: جدولة الأولويات (Priority Scheduling)

تُعطى الأولوية للعمليات بناءً على أولوية معينة، قد تكون زمن التنفيذ أو حاجة المستخدم.
تُخصص وحدة المعالجة المركزية للعملية ذات أعلى أولوية (أصغر عدد صحيح يعني أعلى أولوية).

أنواع الجدولة

1. **(Preemptive)**: يمكن إيقاف عملية ذات أولوية أقل إذا ظهرت عملية ذات أولوية أعلى.
2. **(Nonpreemptive)**: لا يمكن إيقاف العملية حتى تنتهي أو تنتقل إلى حالة الانتظار.

العلاقة مع SJF

- تعتبر SJF نوعاً من جدولة الأولويات حيث تكون الأولوية معكوسة لوقت الانفجار المتوقع التالي لوحدة المعالجة المركزية. حيث تكون الأولوية في جدولة SJF للعمليات التي لها وقت اقصر للتنفيذ.

التحديات

- مشكلة التجويع (**Starvation**): العمليات ذات الأولوية المنخفضة قد لا تُنفذ أبداً.

الحل

- **(Aging)**: مع مرور الوقت، تزداد أولوية العملية، مما يساعد في ضمان أن جميع العمليات ستحصل على فرصة للتنفيذ.

مثال على جدولة الأولويات (Priority Scheduling)

العملية Process	Burst Time وقت الانفجار	الأولوية Priority	وقت الانتظار
P1	10	3	6 : 16
P2	1	1	0 : 1
P3	2	4	16 : 18
P4	1	5	18 : 19
P5	5	2	1 : 6

حسابات وقت الانتظار

- P1: $16 - 10 = 6$
- P2: $1 - 1 = 0$
- P3: $18 - 2 = 16$
- P4: $19 - 1 = 18$
- P5: $6 - 5 = 1$

متوسط وقت الانتظار

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

رابعاً: الجدولة الدائرية (Round Robin - RR)

- كل عملية تحصل على فترة زمنية محددة (Time Quantum) وتتم جدولة العمليات بالتناوب.
- تشبه طريقة FCFS (First-Come, First-Served)، ولكن يتم إضافة خاصية الإيقاف المؤقت (preemption) للتبديل بين العمليات.

كيفية العمل

- يتم تحديد وحدة زمنية، تُسمى quantum time أو time slice، وعادةً ما تتراوح بين ١٠ مللي ثانية إلى ١٠٠ مللي ثانية.
- يتم التعامل مع قائمة العمليات كدائرة، حيث يخصص جدول وحدة المعالجة المركزية لكل عملية في قائمة الانتظار لفترة زمنية تصل إلى 1 quantum.

خطوات الجدولة

- يبدأ جدول وحدة المعالجة المركزية بتخصيص الوقت للعملية الأولى في القائمة.
- إذا لم تنتهِ العملية خلال وقتها المخصص، يتم إيقافها وإعادة إضافتها إلى نهاية القائمة.
- يتم تكرار هذه العملية لجميع العمليات في القائمة.

مثال على الجدولة الدائرية (Round Robin - RR) مع وحدة زمنية = ٤ لكل عملية

العملية Process	وقت الانفجار Arrival Burst	وقت الانتظار
P1	24	0 : 4
P2	3	4 : 7
P3	3	7 : 10
P1	24	10 : 14
P1	24	14 : 18
P1	24	18 : 22
P1	24	22 : 24

حساب وقت الدوران ووقت الانتظار في الجدولة الدائرية

١. وقت الدوران (Turnaround Time)

Turn Around time = Completion time – Arrival time

٢. وقت الانتظار (Waiting Time)

Waiting time = Turn Around time – Burst time

Process ID	Completion Time	Turnaround Time	Waiting Time
P1	30	$30 - 0 = 30$	$30 - 24 = 6$
P2	7	$7 - 0 = 7$	$7 - 3 = 4$
P3	10	$10 - 0 = 10$	$10 - 3 = 7$

Average Turn Around time
 $= (30 + 7 + 10) / 3$
 $= 47 / 3 = 15.66 \text{ ms}$

Average waiting time
 $= (6 + 4 + 7) / 3$
 $= 17 / 3 = 5.66 \text{ ms}$

خامساً: قائمة الانتظار متعددة المستويات Multilevel Queue Scheduling

تقوم بتقسيم العمليات إلى فئات مختلفة، كل فئة لها خوارزمية جدولة خاصة بها.

تقسيم قوائم العمليات

- الواجهة الأمامية (Foreground): تشمل العمليات التفاعلية التي تتطلب استجابة سريعة.
- الخلفية (Background): تشمل العمليات الدفعات التي يمكن أن تعمل دون الحاجة إلى تفاعل مباشر.

- خوارزمية جدولة خاصة بكل قائمة:

- **الواجهة الأمامية:** تستخدم خوارزمية Round Robin (RR)، والتي تعطي كل عملية وقتاً محدداً (شريحة زمنية) للتنفيذ بالتناوب.
- **الواجهة الخلفية:** تستخدم خوارزمية First-Come, First-Served (FCFS)، حيث يتم تنفيذ العمليات حسب ترتيب وصولها.

جدولة بين القوائم

- جدولة ذات أولوية ثابتة: يتم تنفيذ جميع العمليات من الواجهة الأمامية أولاً، ثم يتم الانتقال إلى الخلفية. هذا قد يؤدي إلى الحرمان، حيث يمكن أن تبقى العمليات في الخلفية دون تنفيذ لفترات طويلة.

- يتم تخصيص نسبة معينة من وقت CPU لكل قائمة:

- ٨٠٪ من الوقت للواجهة الأمامية باستخدام RR.

- ٢٠٪ من الوقت للخلفية باستخدام FCFS.