CPU (Process) Scheduling

مقدمة :

أن البرمجة المتعددة تستخدم لزيادة كفاءة النظام . لذا أذا كانت هنالك عملية توقفت في انتظار متطلبات أخري ، فيجب الاستفادة من المعالج في معالجة عملية أخري حتى لا يظل عاطل عن العمل .

قد توجد أكثر من عملية في انتظار المعالجة ، لذا أي من هذه العمليات يتم اختياره ؟

عملية اختيار عملية من مجموعة عمليات أخري ، تسمي بالجدولة (جدولة العمليات) . تعتمد الجدولة علي عدد من المعايير منها :

1. **استخدام المعالج :** وهو زمن استخدامه الفعلي ، يعتبر من اهم المعايير في أنظمة المشاركة (غير مهم في نظام المستخدم الواحد).

2. **الطاقة الإنتاجية :** هي عدد العمليات التي يمكن تنفيذها في زمن محدد (معدل العمليات التي أنجزت).
3. **الوقت الكامل :** هو الزمن التي تقضيه العملية في النظام منذ إنشائها وحتى نهاية تنفيذها (يتناسب بصورة عكسيه مع الإنتاجية).
4. **زمن الاستجابة :** هو الزمن المستغرق من استلام الطلب حتى ظهور النتيجة.
5. **زمن الانتظار :** هو متوسط الفترة الزمنية التي تقضيها العملية في الانتظار
6. **العدل :** يجب أن تكون طريقة الجدولة عادلة .
7. **الأولويات :** تعني الأفضلية للمهمة في التنفيذ ، حيث تقوم الجدولة باختيار المهمات ذات الأولويات العليا .

معايير الجدولة (Scheduling Criteria):

هي خوارزميات توضح كيفية دخول العمليات إلي وحدة المعالجة المركزية ، تستخدم العلاقات الرياضية لتحديد زمن الدخول وزمن المعالجة :

- CPU utilization : keep the CPU as busy as possible(جعل المعالج مشغول).
- Throughput : of processes that complete their execution per time unit عدد العمليات المنجزة في).
  (زمن
- Turnaround time: amount of time to execute a particular process.(الوقت اللازم لتنفيذ مهمة ما).

- Waiting time : amount of time a process has been waiting in the ready queue

  (الزمن الذي تنتظره العملية قبل أن يأتي دورها في التنفيذ)
- Response time : amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)
- Burst Time :

  (هو الزمن الذي يستغرقه المعالج للانتهاء من المهمة)

خوارزميات الجدولة :

1. First-Come, First-Served (FCFS) Scheduling:

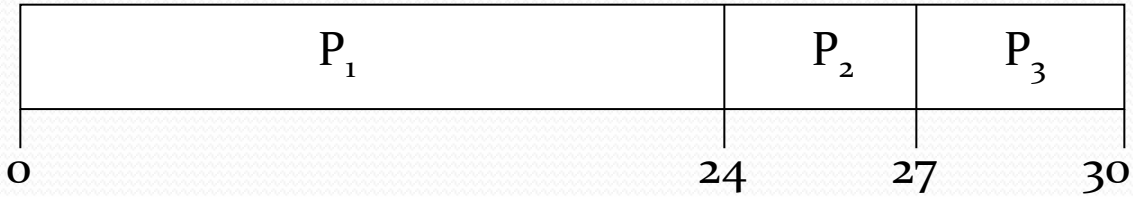
تعتبر من ابسط طرق الجدولة ، تعتمد علي فكرة من يأتي أولا يدخل أولا إلي المعالج (FIFO) ، يكون هنالك صف للعمليات في صف Ready ، عند الانتهاء من واحدة يتم الانتقال إلي الاخري في مقدمة الصف . انظر الجدول أدناه:

العملية	زمن الوصول	زمن الخدمة
P1	0	3
P2	2	6
P3	4	4
P4	6	5
P5	8	2

تنفيذ العمليات يتم بترتيب وصولها حسب الخوارزمية FCFSوهي مفيدة جدا في العمليات الطويلة لكنه غير عادل مع العمليات الصغيرة

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3 The Gantt Chart for the schedule is:

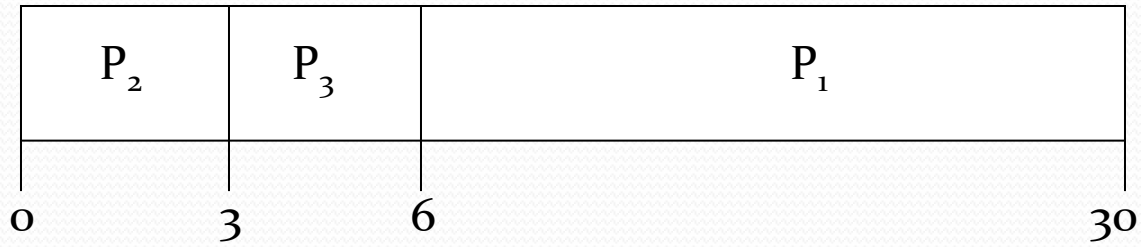


- Waiting time for P_1 = 0; P_2 = 24; P_3 = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

بفرض أن تتالي وصول العمليات قد تغير وأصبح كالتالي :

P_2 , P_3 , P_1

- The Gantt chart for the schedule is:



- Waiting time for $P_1 = 6; P_2 = 0, P_3 = 3$
- Average waiting time: $(6 + 0 + 3)/3 = 3$
- Much better than previous case
- *Convoy effect* short process behind long process

2. Shortest-Job-First (SJF) Scheduling:

العملية التي تأخذ وقتا قصيرا في المعالج تنفذ أولا . وهذه الطريقة تعطي أفضلية للعمليات الصغيرة وتنقسم إلي نوعين :

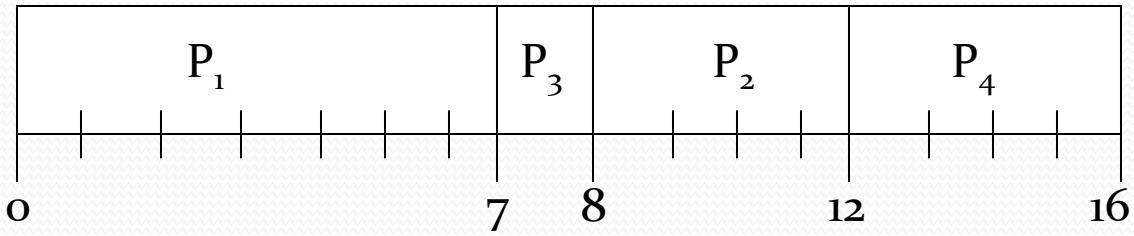
أ. **Non Preemptive(عدم الإيقاف)**: عند وصول العمليات يختار المعالج الأقل زمنا ، عند قدوم عملية جديدة اقل من
التي يعمل عليها المعالج فيتم تجاهلها .

ب. **Preemptive(الإيقاف)**:عند وصول العمليات يختار المعالج الأقل زمنا ، عند قدوم عملية جديدة اقل من التي يعمل عليها المعالج فيتم إيقاف العملية السابقة وتنفيذ الجديدة . يسمي هذا
النوع جدولة اقصر وقت متبقي
Shortest- Remaining –Time-First(SRTF)

مثال

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

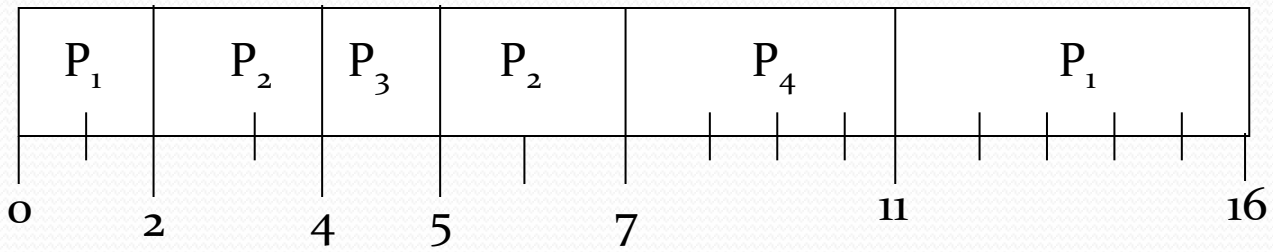
- SJF (non-preemptive)



- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (preemptive)



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

Primitive

مميزاتها :

تقلل متوسط زمن الانتظار .

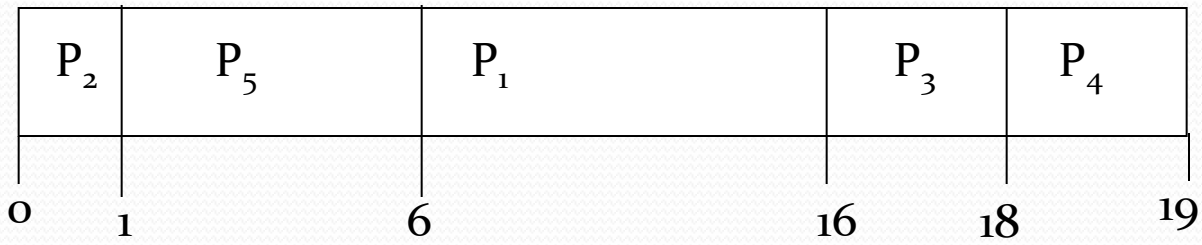
عيوبها :

بها مشكلة تسمي المجاعة Starvation.

3. Priority Scheduling:

استخدمت هذه الخوارزمية لمعالجة نقاط الضعف في خوارزمية SRTF التي تنجز المهام الصغيرة (التي تستغرق اقل فترة زمنية)، تعرف جدولة الأولوية بجدولة اعلي معدل استجابة High Response Ratio Next(HERR) وتتطلب جدولة الأولوية تخصيص مقدار (عدد صحيح) أولوية لكل عملية .

العملية	الأولوية Priority	زمن الخدمة Burst Time
P1	3	10
P2	1	1
P3	4	2
P4	5	1
P5	2	5



زمن الانتظار للعمليات كالتالي :

P1=6 , P2=0 , P3= 16 , P4=18 , p5=1

متوسط زمن الانتظار هو :

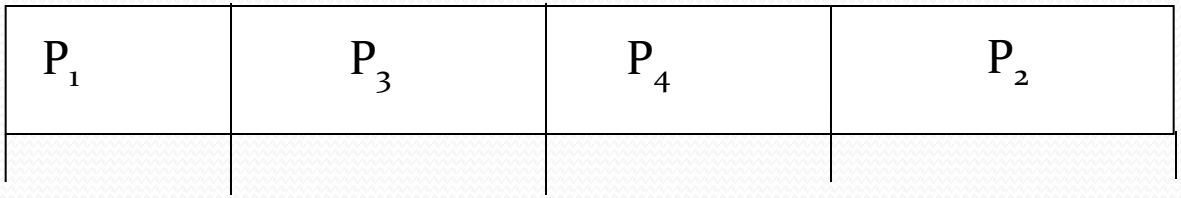
(6+0+16+18+1)5/ = 8.2

العملية	زمن الوصول	الأولوية	زمن المعالجة
P1	0	3	4
P2	1	4	3
P3	2	6	3
P4	3	5	5

إذا كان المعالج يعمل بجدول الأولوية . ارسم مخطط Gantt ليوضح كيفية تنفيذ العمليات أذا كان يعمل وفق :

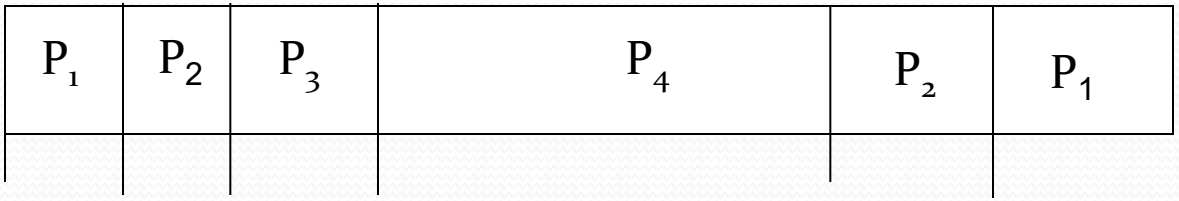
- Non Primitive.
- Primitive.

Non Primitive:



تنفذ العملية p1 إلي أن تكتمل وبعدها تكون العمليات الثلاث الاخري في الانتظار فتنفذ P3 ثم بعد ذلك يتم تنفيذ P4 و P2.

Primitive:



تنفذ العملية P1 لحين وصول P2 بعد وحدة زمنية واحدة عندئذ يتم تنفيذ P2 وبعد مرور وحدتين زمنيتين يتم تنفيذ P3 وهكذا .

4. Round Robin (RR)Scheduling:

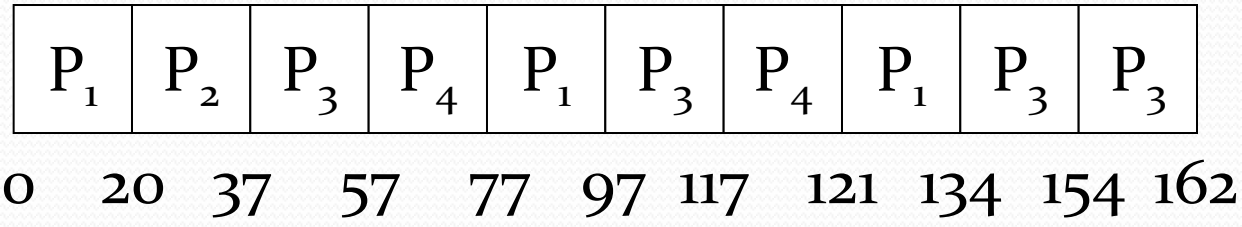
تستخدم هذه الخوارزمية مفهوم المشاركة الزمنية حيث تعمل بمبدأ FIFO ، لكن تعطي لكل مهمة فترة معينة أذا لم تكتمل ينتقل التنفيذ إلي عملية أخري و تعود العملية السابقة إلي أخر صف Ready .

Each process gets a small unit of CPU time (*time quantum*)

يكون لأي عملية زمن ثابت تعالج فيه داخل وحدة المعالجة أذا لم يكتمل تنفيذها ينتقل المعالج إلي تنفيذ عملية أخري .

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

- The Gantt chart is:



- Typically, higher average turnaround than SJF, but better *response*
- *Quantum Time = 20*

ملاحظة هامة :

يجب أن تكون الفترة الزمنية الممنوحة لأي عملية مناسبة وذلك لعدد من الأسباب :

- إذا كانت كبيرة تصبح RR هي FCFS.
- إذا كانت بسيطة تزداد عملية التبديل مما يؤثر في أداء النظام.

Reading

1. Multilevel Queue
2. Multilevel Feedback Queue