



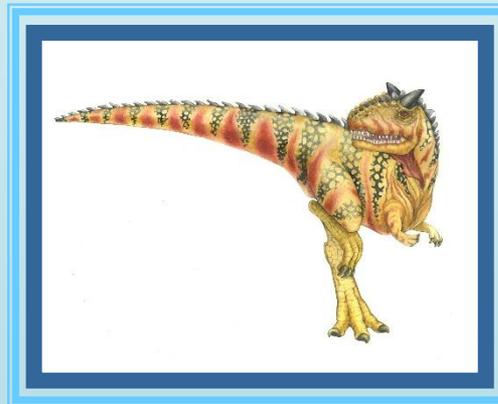
Mustansiriyah University  
College of Education  
Computers Science Department



# Operating Systems Concepts

## Chapter 4: CPU Scheduling

جدولة وحدة المعالجة المركزية



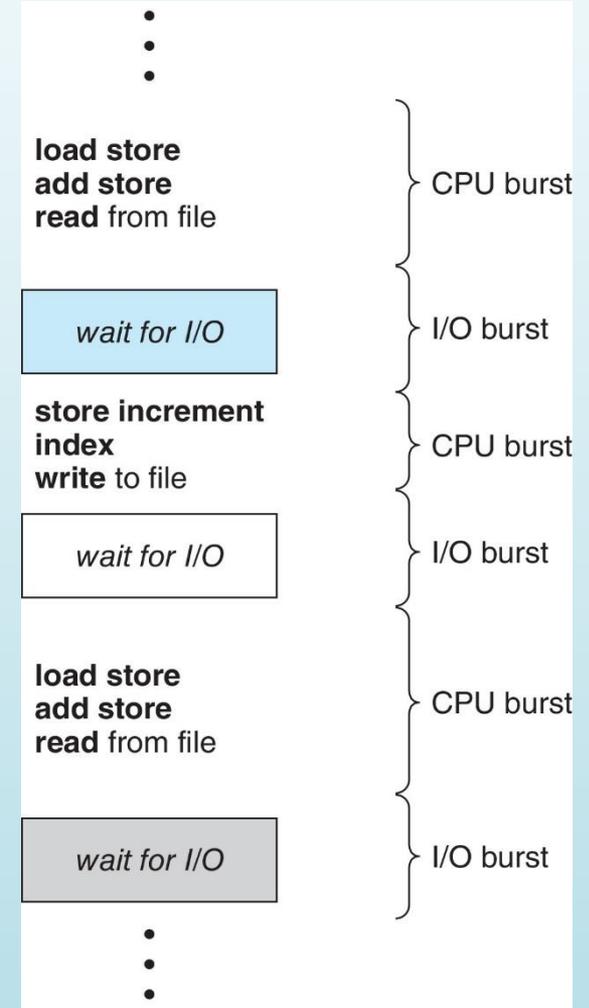
**Dr. Hesham Adnan ALABBASI**

# CPU Scheduling

- تستخدم جدولة وحدة المعالجة المركزية في الانظمة التي تعمل بمبدأ ال Multiprogramming.
- في نظام الحاسبة البسيط ، ال CPU يكون حامل (Ideal), لذلك سيكون هنالك ضياع هو وقت الانتظار. لا يتم إنجاز أي عمل مفيد.
- مع ال Multiprogramming، نحاول استخدام هذا الوقت بشكل منتج, فيكون هنالك بعض ال Processes في التنفيذ في جميع الأوقات وذلك لتحقيق أقصى قدر من الاستفادة من ال CPU.
- يتم الاحتفاظ بعدد من ال Processes في الذاكرة في وقت واحد, ويتم تنفيذ ال Process الى ان يحدث عليها مقاطعة (Interrupt) وتدخل في حالة ال Waiting كأن يكون نوع ال Interrupt هو I/O Request, فيجب أن تنتظرَ لاستكمال هذا ال Request ف يأخذ نظام التشغيل ال CPU من هذه ال Process ويعطي ال CPU الى ال Process أخرى ويستمر بهذا النمط لذلك

# Basic Concepts

- يعتمد نجاح جدولة ال CPU على خاصية ال Processes التي تمت ملاحظتها.
  - تنفيذ ال Process يكون دورة من حالتين:
    - 1- ال CPU execution
    - 2- ال I/O wait وتسمى بالاندفاع (Burst)
- ال Processes تتبادل بين هاتين الحالتين CPU burst يتبعها I/O burst ومن ثم يتبعها CPU burst وهكذا.
- أخيراً ، ينتهي ال burst النهائي لل CPU بطلب إنتهاء التنفيذ.



## CPU Scheduler (مجدول)

- عندما تصبح الـ CPU في وضع الخمول (Idle) ، يجب على نظام التشغيل أن يختار إحدى الـ Processes الموجودة في الـ ready queue لكي تنفذ.
- عملية الاختيار هذه الـ Process يتم تنفيذ عن طريق **short-term scheduler** او **CPU Scheduler**
- يقوم الـ Scheduler بأختيار Process واحدة من الـ Processes الموجودة في الذاكرة (في الـ ready queue) والتي تكون جاهزة للتنفيذ ، ويخصص او يحدد الـ CPU لتلك الـ Process .

## Dispatcher (المرسل)

- الـ Dispatcher (المرسل) يعطي السيطرة على الـ CPU للـ Process المحددة بواسطة **short-term scheduler**
- يجب أن يكون المرسل أسرع ما يمكن لأنه يتم استدعاؤه أثناء عملية تبديل الـ Process.
- يُعرف الوقت المستغرق من قبل الـ Dispatcher لإيقاف إحدى الـ Process وبدء تشغيل أخرى بأسم **Dispatch latency**

# معايير الجدولة Scheduling Criteria

تحتوي خوارزميات جدولة الـ (CPU) المختلفة على خصائص متعددة، وقد يؤدي اختيار خوارزمية معينة إلى تفضيل فئة واحدة من العمليات على أخرى.  
تم اقتراح العديد من المعايير لمقارنة خوارزميات جدولة الـ (CPU) تشمل المعايير ما يلي:

- **CPU utilization**: الاستفادة من الـ CPU: أبقاء الـ CPU مشغولة بقدر المستطاع
- **Throughput**: الانتاجية: عدد الـ Processes التي اكتمل تنفيذها خلال وحدة زمن.
- **Turnaround time**: كمية الوقت المستغرق لتنفيذ process محدد.
- **Waiting time**: وقت الانتظار: كمية الوقت الذي يقضيه الـ Process للانتظار في الـ ready queue .
- **Response time**: وقت الاستجابة: كمية الوقت المحسوب من تقديم (دخول) الـ Process الى الوقت الذي تتم اول استجابة له

معايير تحسين خوارزمية الجدولة Scheduling Algorithm Optimization Criteria are:

- Max CPU utilization, Max throughput
- Min turnaround time, Min waiting time and Min response time

# Scheduler Algorithms

خوارزميات الجدولة قد تكون:

1- استباقية **Preemptive** بمعنى يمكن استقطاع او ايقاف عمل الـ Process قيد التنفيذ والانتقال بالتنفيذ الى Process اخرى.

2- غير استباقية **Non-Preemptive** بمعنى لا يمكن استقطاع او ايقاف عمل الـ Process قيد التنفيذ والانتقال بالتنفيذ الى Process اخرى.

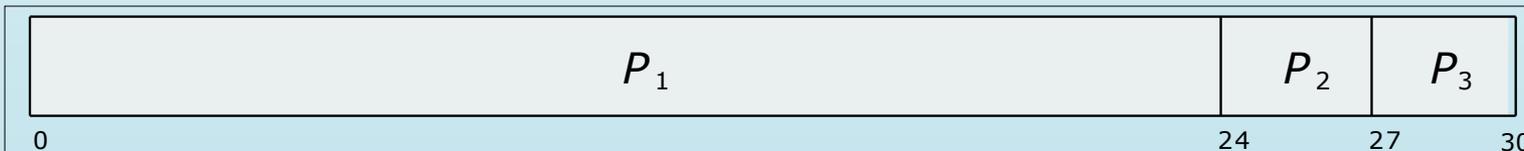
**ملاحظة: كل Process يحتاج الى وقت محدد لاكمال تنفيذه يدعى بالـ Burst time**

# 1- First- Come, First-Served (FCFS) Scheduling

عمل الخوارزمية : يتم التنفيذ اعتمادا على وصول الـ Processes وايضا وفق ترتيب وصولها, بمعنى تنفيذ الاول ثم الثاني الى الاخير وفق ترتيب الوصول المحدد في السؤال, وهذه الخوارزمية هي **Nonpreemptive**.

Example 1: Suppose that the processes arrive at time 0, in the order:  **$P_1, P_2, P_3$**  Draw the Gantt chart and calculate the average waiting?

<u>Processes</u>	<u>Burst time</u>
P1	24
P2	3
P3	3



- Waiting time:

$$P_1 = (0-0)=0$$

$$P_2 = (24-0)=24$$

$$P_3 = (27-0)=27$$

▪ **Waiting time = start time - arrival time** ▪

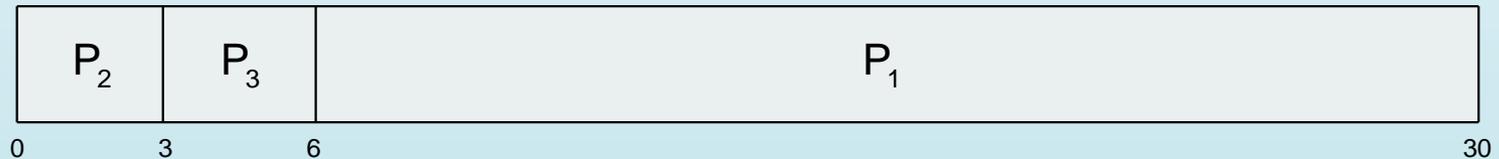
وقت الانتظار = وقت بدأ التنفيذ - وقت الوصول

- Average waiting time:  $(0 + 24 + 27)/3 = 17$  Ms.

## FCFS Scheduling (Cont.)

Example 2: Suppose that the processes arrive at time 0, in the order:  **$P_2, P_3, P_1$**  Draw the Gantt chart and calculate the average waiting?

<u>Processes</u>	<u>Burst time</u>
P1	24
P2	3
P3	3



- Waiting time:

$$P_1 = 6$$

$$P_2 = 0$$

$$P_3 = 3$$

- Average waiting time:  $(6 + 0 + 3)/3 = 3$  Ms.

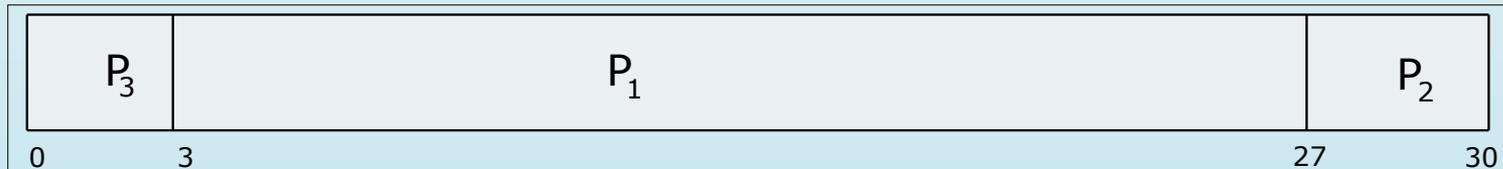
Much better than previous case

## FCFS Scheduling (Cont.)

Example 3: Draw the Gantt chart and calculate the average waiting for the processes as in the given table ?

<u>Processes</u>	<u>Burst time</u>	<u>Arrival time</u>
P1	24	2
P2	3	5
P3	3	0

<u>Processes</u>	<u>Burst time</u>	<u>Arrival time</u>
P3	3	0
P1	24	2
P2	3	5



**Waiting time: start time – arrival time** ▪

▪ Waiting time:

$$P_1 = (3-2)=1$$

$$P_2 = (27-5)=22$$

$$P_3 = (0-0)=0$$

▪ Average waiting time:  $(1 + 22 + 0)/3 = (23/3) = 7.666$  Ms.

# End of Chapter 4- Part 1

