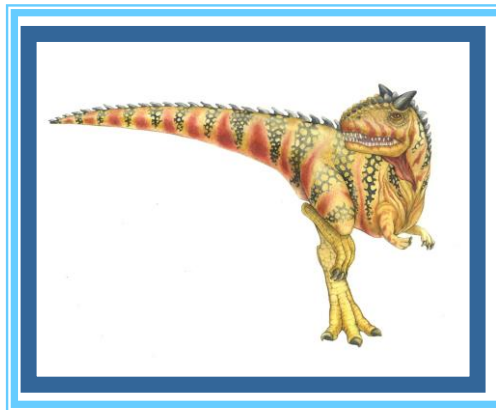
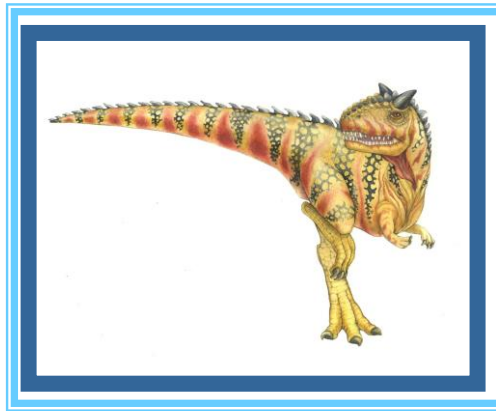


# Chapter 1: Introduction

---



# الفصل 1 مقدمة نظم تشغيل





# Chapter 1: Introduction

---

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Distributed Systems
- Special-Purpose Systems
- Computing Environments
- Open-Source Operating Systems





# الفصل 1 مقدمة

- ماذا تفعل أنظمة التشغيل
- تنظيم نظام الكمبيوتر
- هندسة نظام الكمبيوتر
- هيكل نظام التشغيل
- عمليات نظام التشغيل
- إدارة العمليات
- إدارة الذاكرة
- إدارة التخزين
- الحماية والأمن
- الأنظمة الموزعة
- أنظمة الأغراض الخاصة
- بيئات الحوسبة
- أنظمة تشغيل مفتوحة المصدر





# Objectives

- To provide a grand tour of the major operating systems components
- To provide coverage of basic computer system organization

## الأهداف

- لتقديم جولة كبيرة لمكونات أنظمة التشغيل الرئيسية.
- لتوفير تغطية تنظيم نظام الكمبيوتر الأساسي.





# What is an Operating System?

---

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner





# ما هو نظام التشغيل؟

- برنامج يعمل كوسيط بين مستخدم الكمبيوتر وجهاز الكمبيوتر
- أهداف نظام التشغيل:
- تنفيذ برامج المستخدم وتسهيل حل مشاكل المستخدم
- جعل نظام الكمبيوتر مناسباً للاستخدام
- استخدام أجهزة الكمبيوتر بطريقة فعالة





# Computer System Structure

---

- **Computer system can be divided into four components:**
  - Hardware – provides basic computing resources
    - 4 CPU, memory, I/O devices
  - Operating system
    - 4 Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - 4 Word processors, compilers, web browsers, database systems, video games
  - Users
    - 4 People, machines, other computers







# هيكل نظام الكمبيوتر

■ يمكن تقسيم نظام الكمبيوتر إلى أربعة مكونات :

• الأجهزة - توفر موارد الحوسبة الأساسية

4 وحدة المعالجة المركزية والذاكرة وأجهزة الإدخال / الإخراج

\* نظام التشغيل

4 يتحكم وينسق استخدام الأجهزة بين مختلف التطبيقات والمستخدمين

\* برامج التطبيقات - تحديد طرق استخدام موارد النظام لحل مشكلات الحوسبة للمستخدمين

4 معالجات النصوص ، المجمعات ، متصفحات الويب ، أنظمة قواعد البيانات ،

ألعاب الفيديو

\* المستخدمون

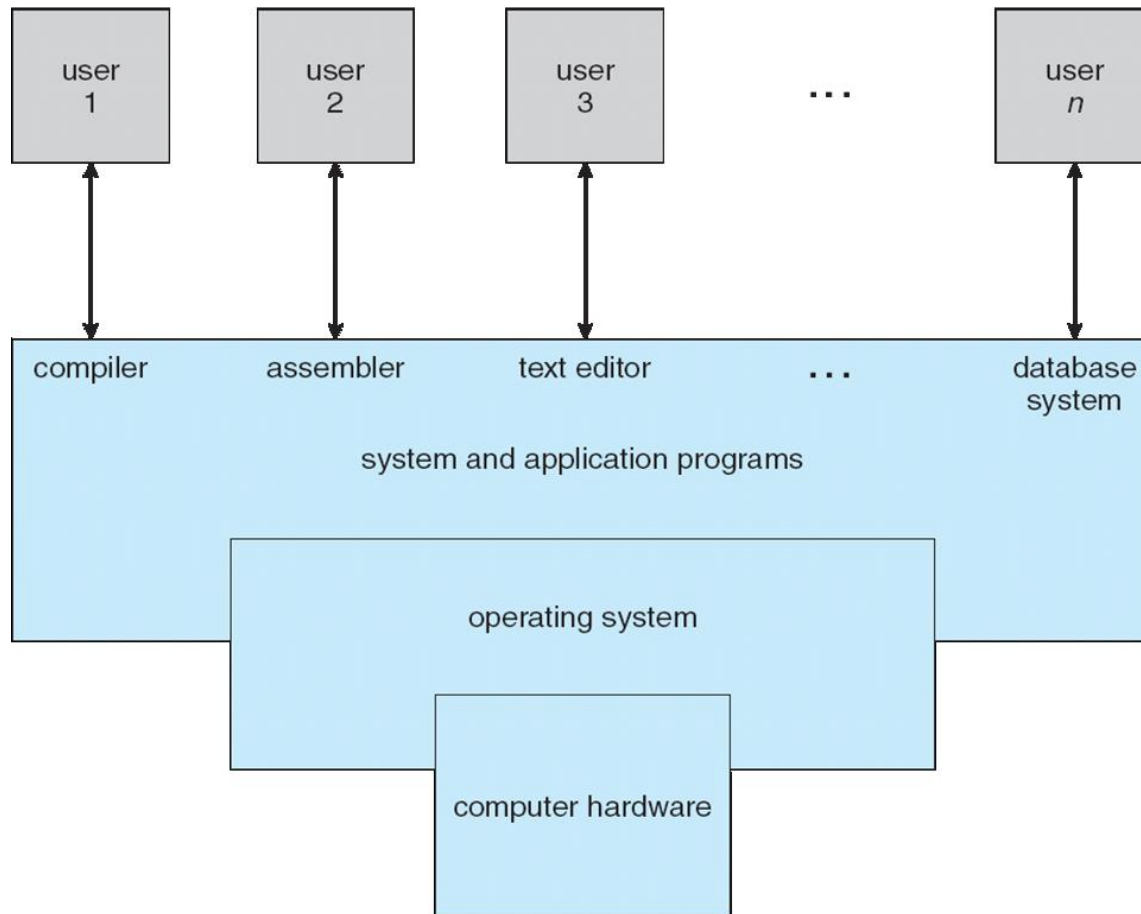
4 أشخاص ، آلات ، أجهزة كمبيوتر أخرى





# Four Components of a Computer System

## أربعة مكونات لنظام الكمبيوتر





# Operating System Definition

---

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer





# تعريف نظام التشغيل

## 1- نظام التشغيل هو مخصص للموارد

- يدير جميع الموارد
- يقرر بين الطلبات المتضاربة للاستخدام الفعال والعادل للموارد

## 2- نظام التشغيل هو برنامج تحكم

- يتحكم في تنفيذ البرامج لمنع الأخطاء والاستخدام غير السليم للكمبيوتر





# Operating System Definition (Cont)

---

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is good approximation
  - But varies wildly
- “The one program running at all times on the computer” is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program





# تعريف نظام التشغيل (تابع)

- لا يوجد تعريف مقبول عالميًا
- يعتبر "كل شيء يشحنه البائع عند طلب نظام تشغيل" تقديرًا جيدًا \* ولكن يختلف تمام
- "البرنامج الوحيد الذي يعمل في جميع الأوقات على الكمبيوتر" هو النواة. كل شيء آخر هو إما برنامج نظام (يأتي مع نظام التشغيل) أو برنامج تطبيق





# Computer Startup

---

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution





# بدء تشغيل الكمبيوتر

- يتم تحميل برنامج bootstrap عند التشغيل أو إعادة التشغيل
- يتم تخزينها عادةً في ROM أو EPROM، والمعروفين عمومًا باسم البرامج الثابتة
- تهيئة جميع جوانب النظام
- تحميل نواة نظام التشغيل وابدأ التنفيذ

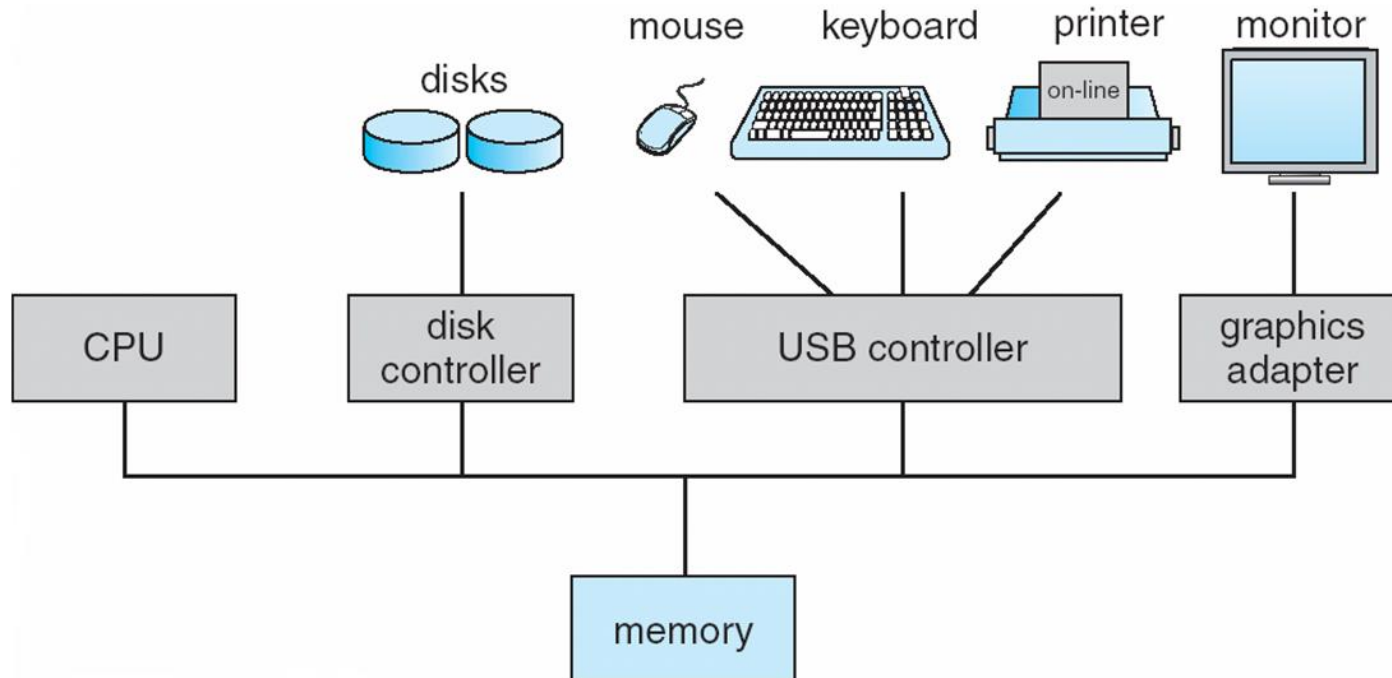






# Computer System Organization

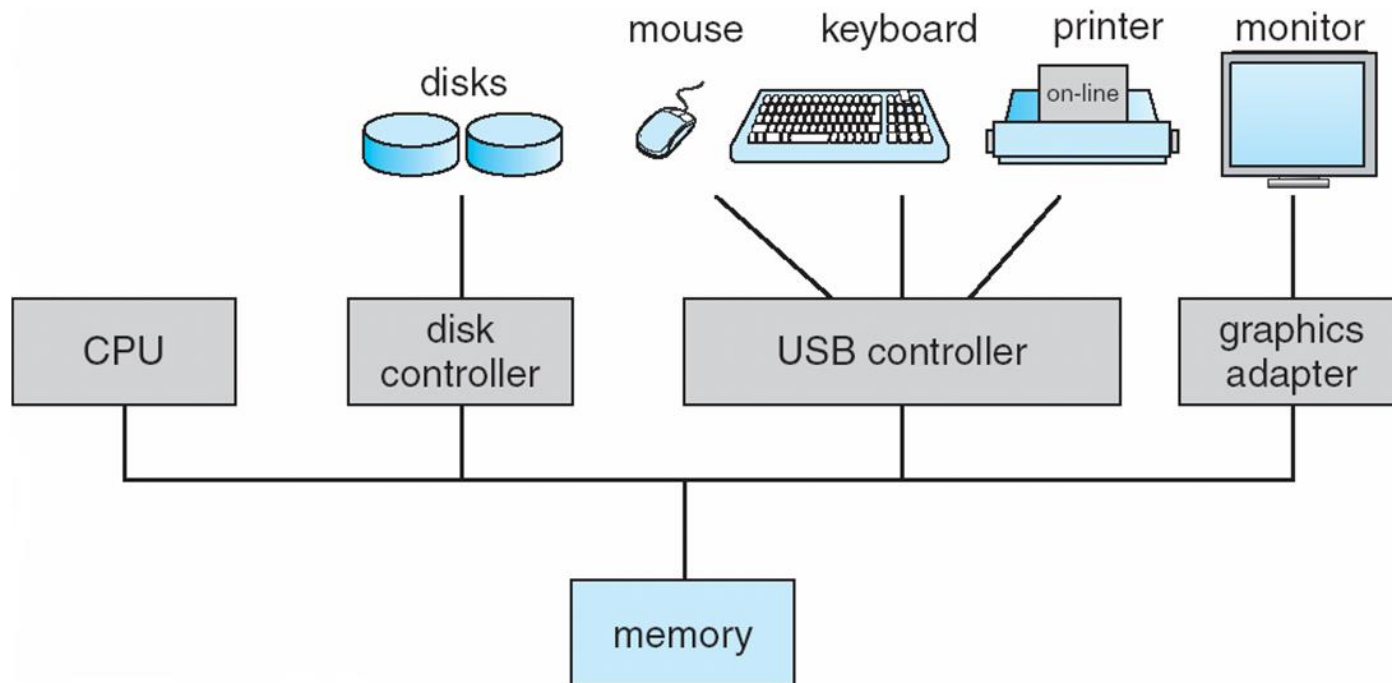
- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles





# تنظيم أنظمة الكمبيوتر

- تشغيل نظام الكمبيوتر
- واحد أو أكثر من وحدات المعالجة المركزية ، تتصل وحدات التحكم في الجهاز من خلال ناقل مشترك يوفر الوصول إلى الذاكرة المشتركة
- التنفيذ المتزامن لوحدات المعالجة المركزية والأجهزة التي تتنافس على دورات الذاكرة





# Computer-System Operation

---

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an *interrupt*





# تشغيل نظام الكمبيوتر

- يمكن أن تعمل أجهزة الإدخال / الإخراج ووحدة المعالجة المركزية بشكل متزامن
- كل جهاز تحكم مسؤول عن نوع جهاز معين
- يحتوي كل جهاز تحكم على مخزن مؤقت محلي
- تنقل وحدة المعالجة المركزية البيانات من / إلى الذاكرة الرئيسية إلى / من المخازن المؤقتة المحلية
- الإدخال / الإخراج من الجهاز إلى المخزن المؤقت المحلي لوحدة التحكم
- تُعلم وحدة التحكم في الجهاز وحدة المعالجة المركزية بأنها انتهت من عملها عن طريق التسبب في مقاطعة





# Common Functions of Interrupts

---

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*
- A *trap* is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**





# الوظائف المشتركة للمقاطع

- عمليات نقل المقاطعة إلى روتين خدمة المقاطعة بشكل عام ، من خلال **ناقل المقاطعة** ، الذي يحتوي على عناوين جميع إجراءات الخدمة
- يجب أن تحفظ بنية المقاطعة عنوان التعليمات المتقطعة
- يتم تعطيل المقاطعات الواردة أثناء معالجة مقاطعة أخرى لمنع مقاطعة مفقودة
- **الفخ** هو مقاطعة يتم إنشاؤها بواسطة البرامج وتحدث إما عن طريق خطأ أو طلب مستخدم
- نظام التشغيل **يحرکه المقاطعة**





# Interrupt Handling

---

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt





# معالجة المقاطعة

- يحافظ نظام التشغيل على حالة وحدة المعالجة المركزية عن طريق تخزين السجلات وعداد البرنامج
- يحدد نوع الانقطاع الذي حدث:

\* الاقتراع

\* نظام المقاطعة المتجه

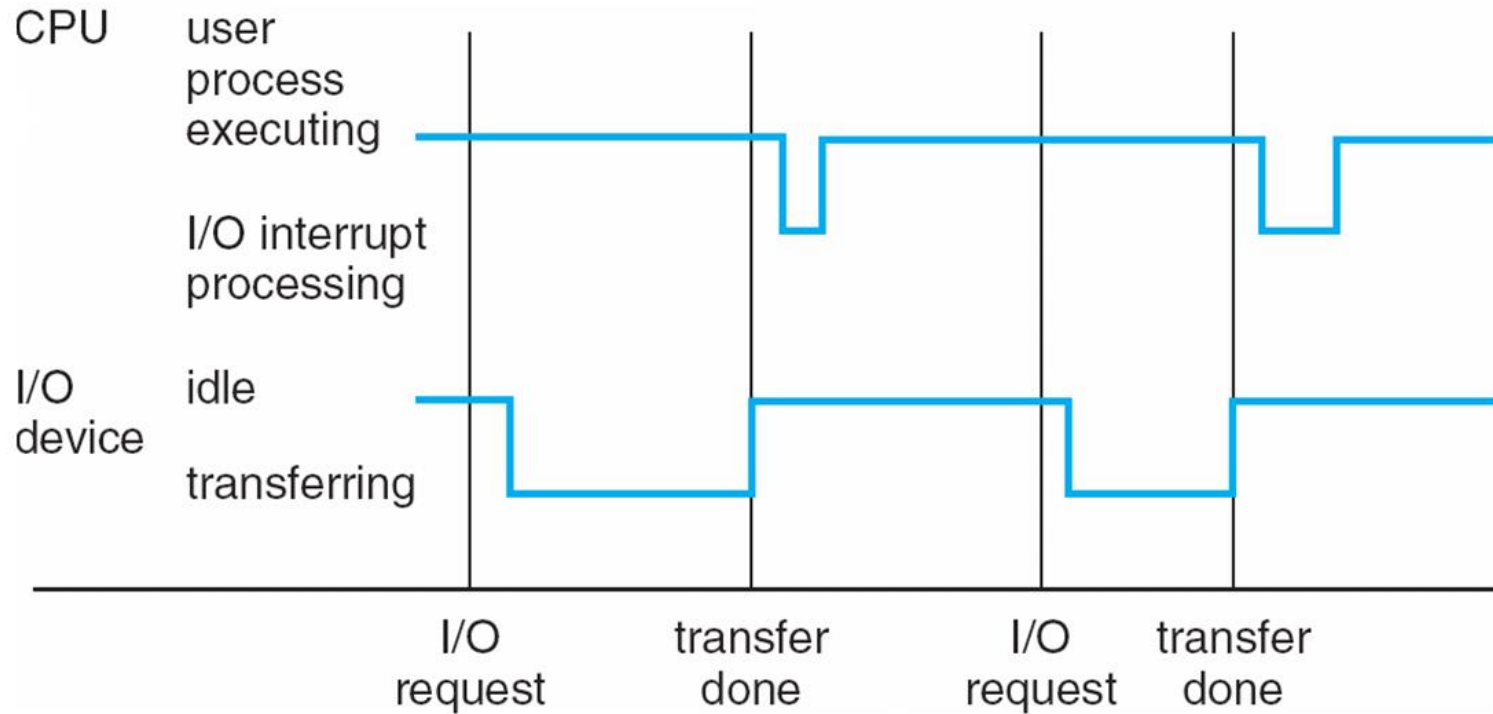
- تحدد الأجزاء المنفصلة من التعليمات البرمجية الإجراء الذي يجب اتخاذه لكل نوع من أنواع المقاطعة







# Interrupt Timeline





# I/O Structure

---

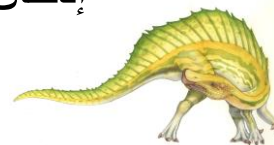
- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns to user program without waiting for I/O completion
  - **System call** – request to the operating system to allow user to wait for I/O completion
  - **Device-status table** contains entry for each I/O device indicating its type, address, and state
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt





# هيكل الإدخال / الإخراج

- بعد بدء O / I، يعود التحكم إلى برنامج المستخدم فقط عند اكتمال الإدخال / الإخراج
  - \* انتظر التعليمات تبطل وحدة المعالجة المركزية حتى المقاطعة التالية
  - \* حلقة الانتظار (جهاد للوصول إلى الذاكرة)
  - \* على الأكثر طلب O / I واحد معلق في كل مرة ، لا توجد معالجة O / I متزامنة
- بعد بدء O / I، يعود التحكم إلى برنامج المستخدم دون انتظار اكتمال الإدخال / الإخراج
- استدعاء النظام - طلب إلى نظام التشغيل للسماح للمستخدم بالانتظار حتى اكتمال الإدخال / الإخراج
- يحتوي جدول حالة الجهاز على إدخال لكل جهاز إدخال / إخراج يشير إلى نوعه وعنوانه وحالته
- فهارس نظام التشغيل في جدول جهاز الإدخال / الإخراج لتحديد حالة الجهاز وتعديل إدخال الجدول ليشمل المقاطعة





# Direct Memory Access Structure

---

- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte





# هيكل الوصول المباشر للذاكرة

- تستخدم لأجهزة الإدخال / الإخراج عالية السرعة القدرة على نقل المعلومات بسرعة قريبة من سرعات الذاكرة
- تنقل وحدة التحكم في الجهاز كتل البيانات من التخزين المؤقت مباشرة إلى الذاكرة الرئيسية دون تدخل وحدة المعالجة المركزية
- يتم إنشاء مقاطعة واحدة فقط لكل كتلة ، بدلاً من مقاطعة واحدة لكل بايت





# Storage Structure

---

- Main memory – only large storage media that the CPU can access directly
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity
- Magnetic disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer





# هيكل التخزين

- الذاكرة الرئيسية - وسائط التخزين الكبيرة فقط التي يمكن لوحدة المعالجة المركزية الوصول إليها مباشرة
  - التخزين الثانوي - امتداد للذاكرة الرئيسية التي توفر سعة تخزين كبيرة غير متطايرة
  - الأقراص المغناطيسية - وهي عبارة عن أطباق معدنية أو زجاجية صلبة مغطاة بمادة تسجيل مغناطيسية
- \* ينقسم سطح القرص بشكل منطقي إلى مسارات مقسمة إلى قطاعات
- \* تحدد وحدة التحكم بالقرص التفاعل المنطقي بين الجهاز والكمبيوتر





# Storage Hierarchy

---

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a last *cache* for secondary storage







# التسلسل الهرمي للتخزين

- نظم التخزين مرتبة في التسلسل الهرمي

\* سرعة

\* كلفة

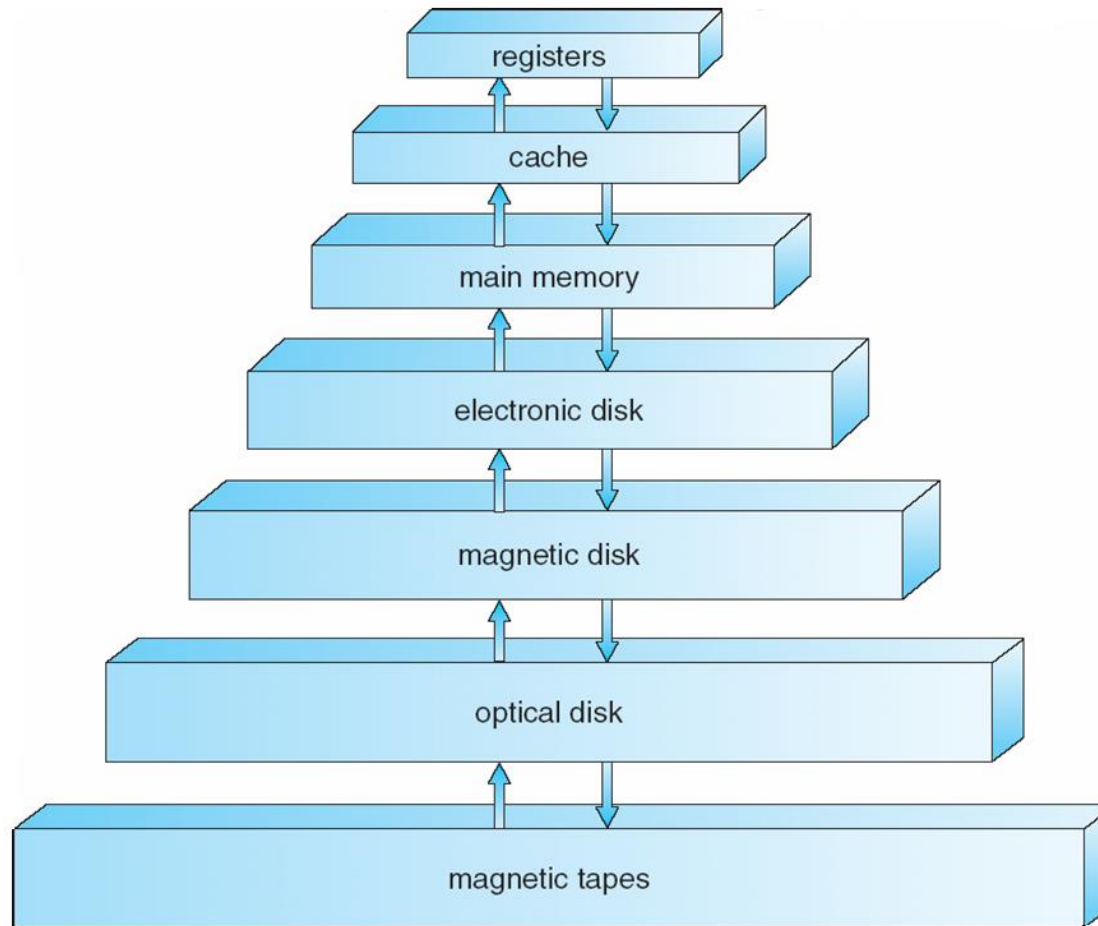
\* النقلب

- التخزين المؤقت - نسخ المعلومات إلى نظام تخزين أسرع ؛ يمكن اعتبار الذاكرة الرئيسية بمثابة ذاكرة تخزين مؤقت أخيرة للتخزين الثانوي





# Storage-Device Hierarchy





# Caching

---

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy





# التخزين المؤقت

- مبدأ هام ، يتم إجراؤه على عدة مستويات في الكمبيوتر (في الأجهزة ، ونظام التشغيل ، والبرامج)
- نسخ المعلومات قيد الاستخدام من التخزين الأبطأ إلى الأسرع مؤقتًا
- \* فحص أسرع للتخزين (ذاكرة التخزين المؤقت) أولاً لتحديد ما إذا كانت المعلومات موجودة
- \* إذا كان الأمر كذلك ، فإن المعلومات المستخدمة مباشرة من ذاكرة التخزين المؤقت (سريع)
- \* إذا لم يتم نسخ البيانات إلى ذاكرة التخزين المؤقت واستخدامها هناك
- ذاكرة التخزين المؤقت أصغر من التخزين المؤقت
- \* إدارة ذاكرة التخزين المؤقت مشكلة تصميم مهمة
- \* حجم ذاكرة التخزين المؤقت وسياسة الاستبدال





# Computer-System Architecture

---

- Most systems use a single general-purpose processor (PDAs through mainframes)
  - Most systems have special-purpose processors as well
- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance
  - Two types
    1. Asymmetric Multiprocessing
    2. Symmetric Multiprocessing





# هندسة نظام الكمبيوتر

- تستخدم معظم الأنظمة معالجًا واحدًا متعدد الأغراض (أجهزة المساعد الرقمي الشخصي من خلال أجهزة الكمبيوتر المركزية)  
\* تحتوي معظم الأنظمة أيضًا على معالجات ذات أغراض خاصة
- تزايد استخدام النظم متعددة المعالجات وأهميتها

\* تُعرف أيضًا باسم الأنظمة المتوازية ، الأنظمة شديدة التقارن.  
\* تشمل المزايا:

1. زيادة الإنتاجية
  2. اقتصاد الحجم
  3. زيادة الموثوقية - التدهور الجيد أو تحمل الخطأ
- \* نوعين

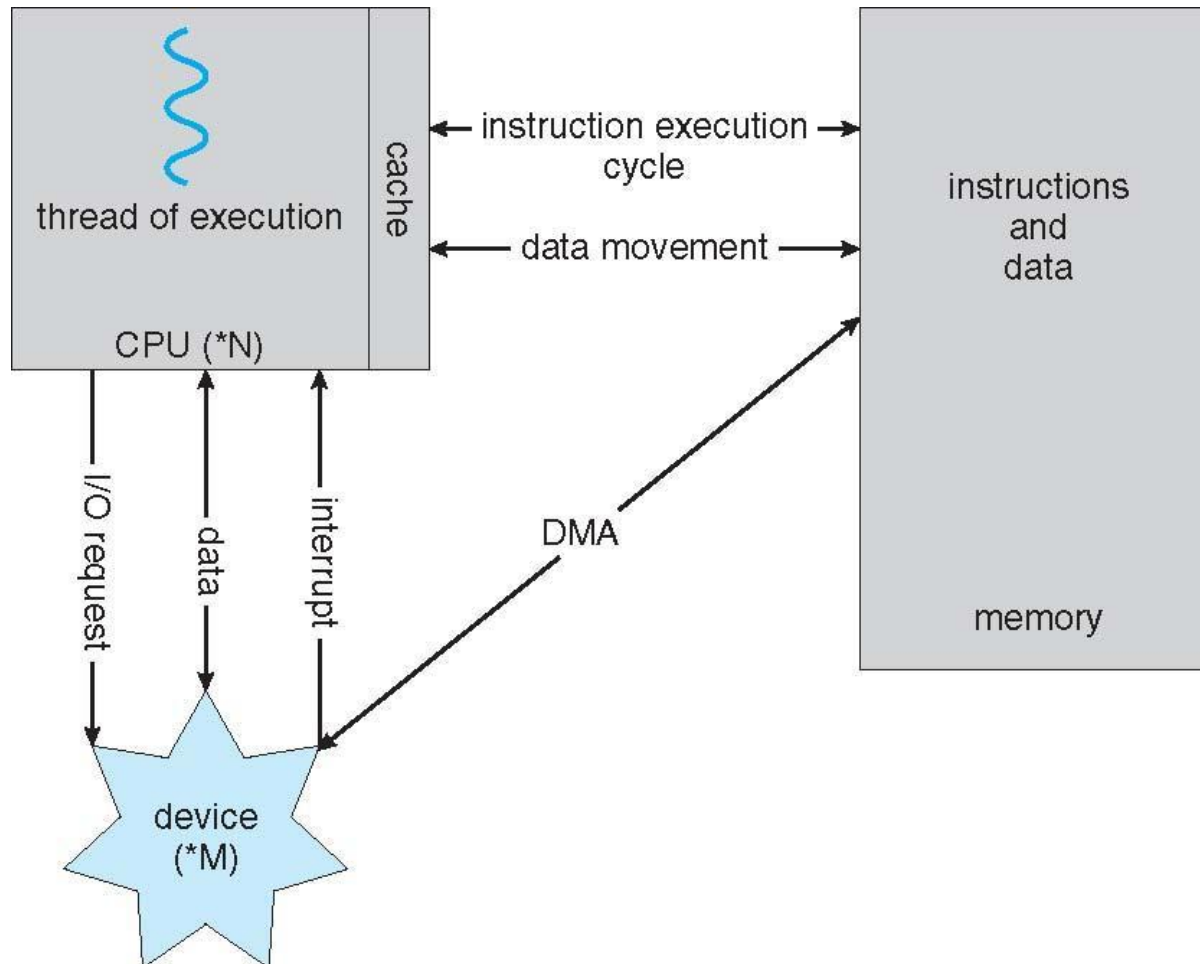
1. المعالجة المتعددة غير المتماثلة
2. المعالجة المتعددة المتماثلة





# How a Modern Computer Works

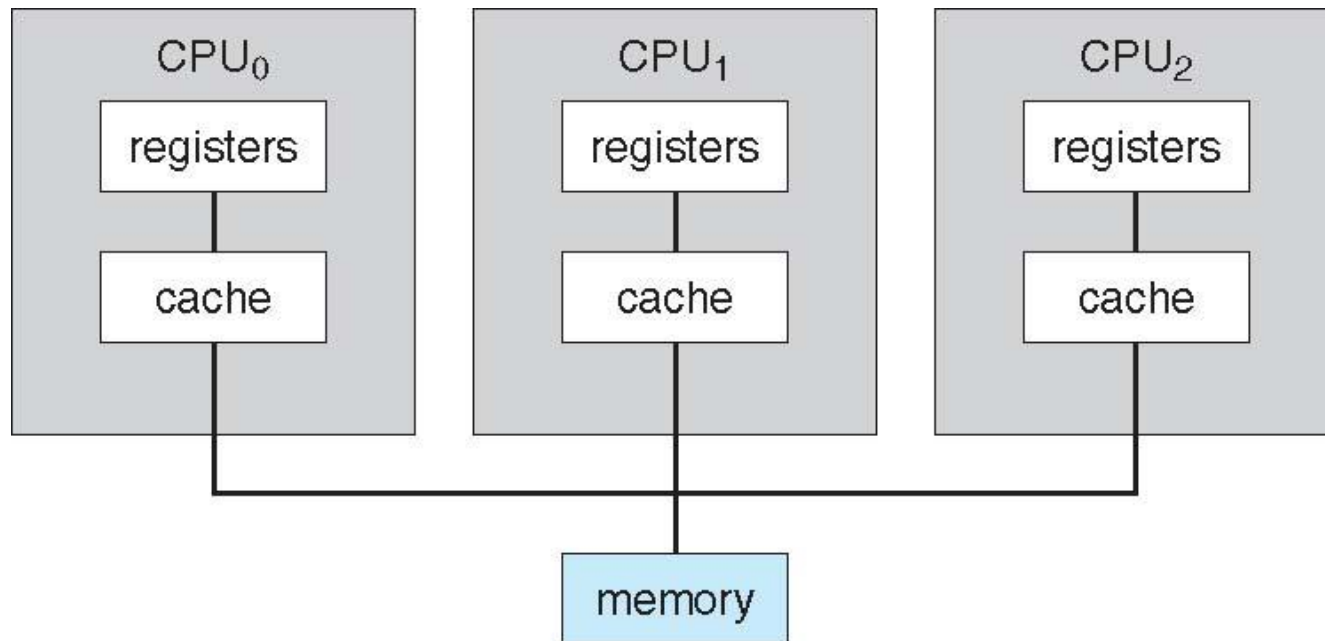
## كيف يعمل الكمبيوتر الحديث





# Symmetric Multiprocessing Architecture

الهندسة المعمارية المتماثلة للمعالجة

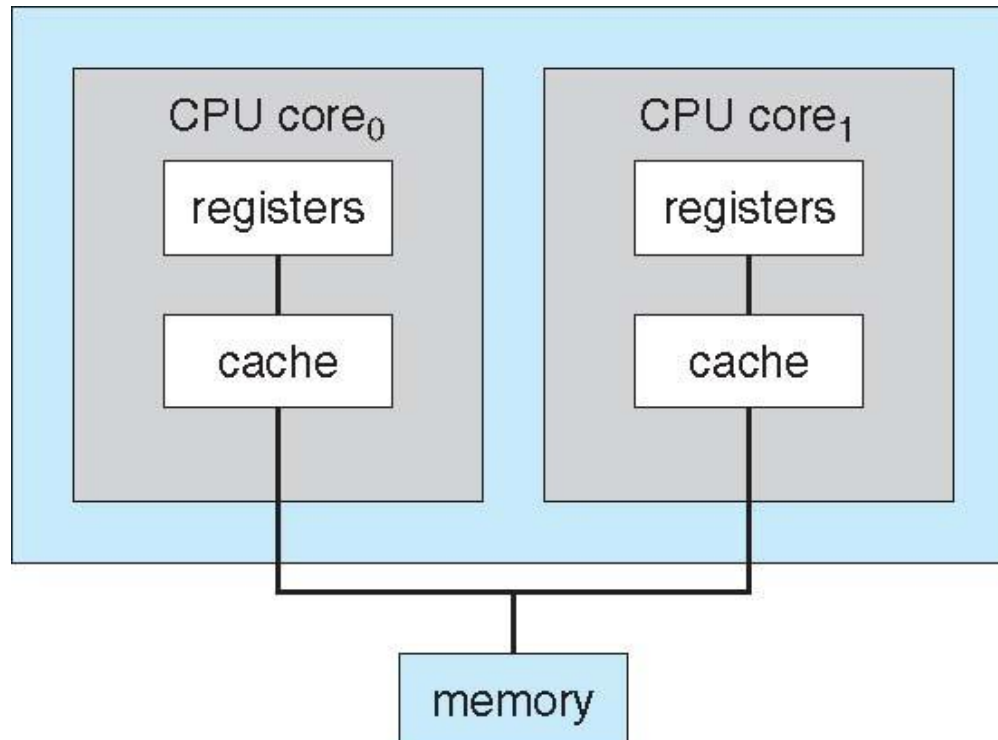






# A Dual-Core Design

تصميم ثنائي النواة





# Clustered Systems

---

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a [storage-area network \(SAN\)](#)
  - Provides a [high-availability](#) service which survives failures
    - 4 [Asymmetric clustering](#) has one machine in hot-standby mode
    - 4 [Symmetric clustering](#) has multiple nodes running applications, monitoring each other
  - Some clusters are for [high-performance computing \(HPC\)](#)
    - 4 Applications must be written to use [parallelization](#)





■ مثل الأنظمة متعددة المعالجات ، لكن أنظمة متعددة تعمل معًا

\* مشاركة التخزين عادةً عبر شبكة منطقة التخزين (SAN)

\* يوفر خدمة عالية التوفر تنجو من الفشل

4 مجموعات غير متماثلة بها آلة واحدة في وضع الاستعداد الساخن

4 مجموعات متماثلة لها عقد متعددة تقوم بتشغيل التطبيقات ، ومراقبة بعضها البعض

\* بعض المجموعات مخصصة للحوسبة عالية الأداء (HPC)

يجب كتابة 4 تطبيقات لاستخدام الموازاة





# Operating System Structure

---

- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job
- **Timesharing (multitasking)** is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be  $< 1$  second
  - Each user has at least one program executing in memory **process**
  - If several jobs ready to run at the same time **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory





# هيكل نظام التشغيل

## • البرمجة المتعددة اللازمة لتحقيق الكفاءة:

\* لا يمكن لمستخدم واحد إبقاء وحدة المعالجة المركزية وأجهزة الإدخال / الإخراج مشغولة في جميع الأوقات  
\* تنظم البرمجة المتعددة الوظائف (التعليمات البرمجية والبيانات) لذا فإن وحدة المعالجة المركزية لديها دائمًا واحد لتنفيذه

\* يتم الاحتفاظ بمجموعة فرعية من إجمالي الوظائف في النظام في الذاكرة  
\* اختيار وظيفة واحدة وتشغيلها عن طريق **جدولة الوظيفة**  
\* عندما يتعين عليه الانتظار (للإدخال / الإخراج على سبيل المثال) ، يتحول نظام التشغيل إلى وظيفة أخرى

• **تعد المشاركة بالوقت (تعدد المهام)** امتدادًا منطقيًا تقوم فيه وحدة المعالجة المركزية بتبديل الوظائف بشكل متكرر بحيث يمكن للمستخدمين التفاعل مع كل وظيفة أثناء تشغيلها ، مما يؤدي إلى إنشاء حوسبة **تفاعلية**

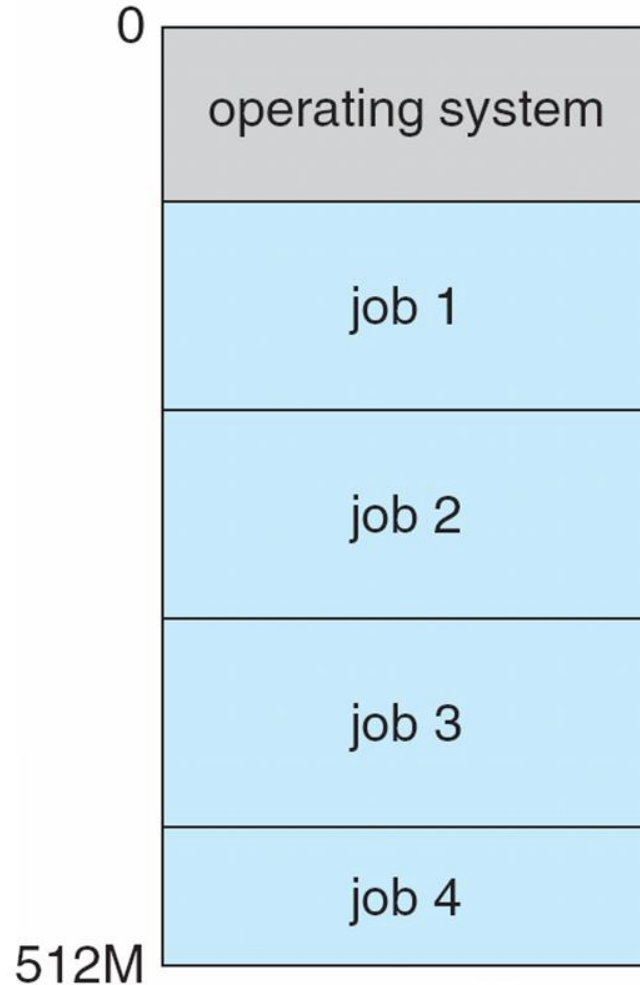
\* يجب أن يكون **وقت الاستجابة** أقل من ثانية واحدة  
\* كل مستخدم لديه برنامج واحد على الأقل ينفذ في **عملية** الذاكرة  
\* إذا كانت عدة مهام جاهزة للتشغيل في نفس الوقت ، **جدولة وحدة المعالجة المركزية**  
\* إذا كانت العمليات لا تتلاءم مع الذاكرة ، فإن **التبديل** يحركها للداخل وللخارج للتشغيل  
\* تسمح **الذاكرة الظاهرية** بتنفيذ العمليات غير الموجودة في الذاكرة بالكامل





# Memory Layout for Multiprogrammed System

## تخطيط الذاكرة لنظام متعدد البرامج





# Operating-System Operations

---

- Interrupt driven by hardware
- Software error or request creates **exception** or **trap**
  - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - 4 Provides ability to distinguish when system is running user code or kernel code
    - 4 Some instructions designated as **privileged**, only executable in kernel mode
    - 4 System call changes mode to kernel, return from call resets it to user





# عمليات نظام التشغيل

- مقاطعة مدفوعة بالأجهزة

- خطأ في البرنامج أو طلب يخلق استثناء أو فخ

\* القسمة على الصفر ، طلب خدمة نظام التشغيل

- تشمل مشاكل العملية الأخرى الحلقة اللانهائية والعمليات التي تعدل بعضها البعض أو نظام التشغيل

- تسمح عملية الوضع المزدوج لنظام التشغيل بحماية نفسه ومكونات النظام الأخرى

\* وضع المستخدم ووضع النواة

\* يتم توفير بت الوضع بواسطة الأجهزة

4 يوفر القدرة على التمييز عند تشغيل النظام لكود المستخدم أو كود النواة

4 تم تعيين بعض الإرشادات على أنها ذات امتياز ، وقابلة للتنفيذ فقط في وضع kernel

4 يقوم استدعاء النظام بتغيير الوضع إلى kernel، والعودة من إعادة تعيينه إلى المستخدم

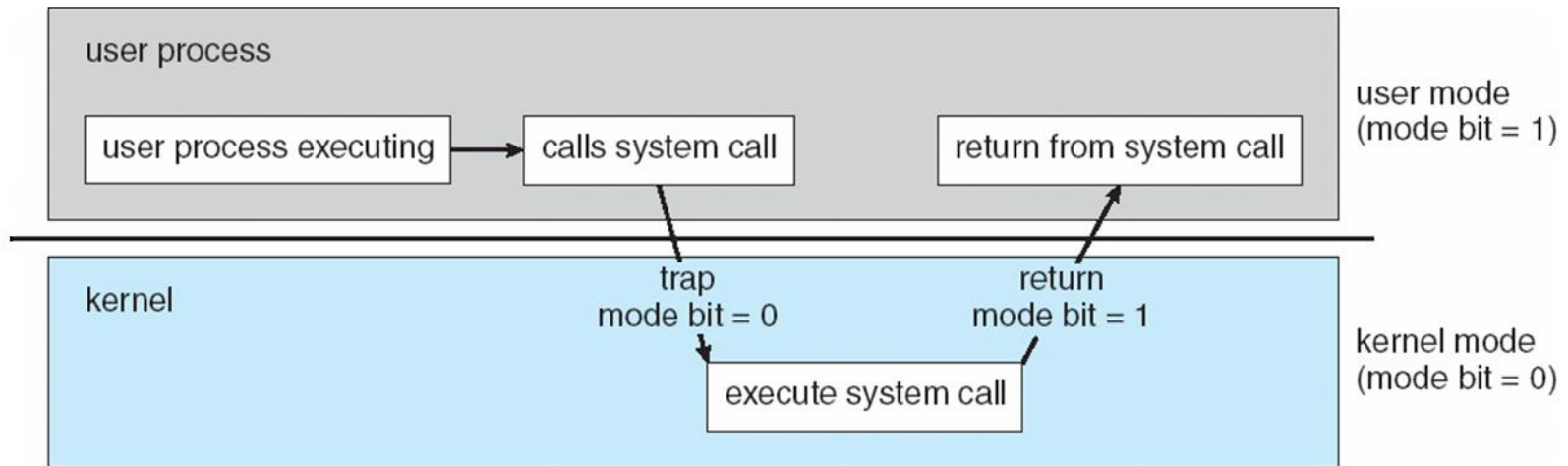






# Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time





# الانتقال من وضع المستخدم إلى وضع Kernel

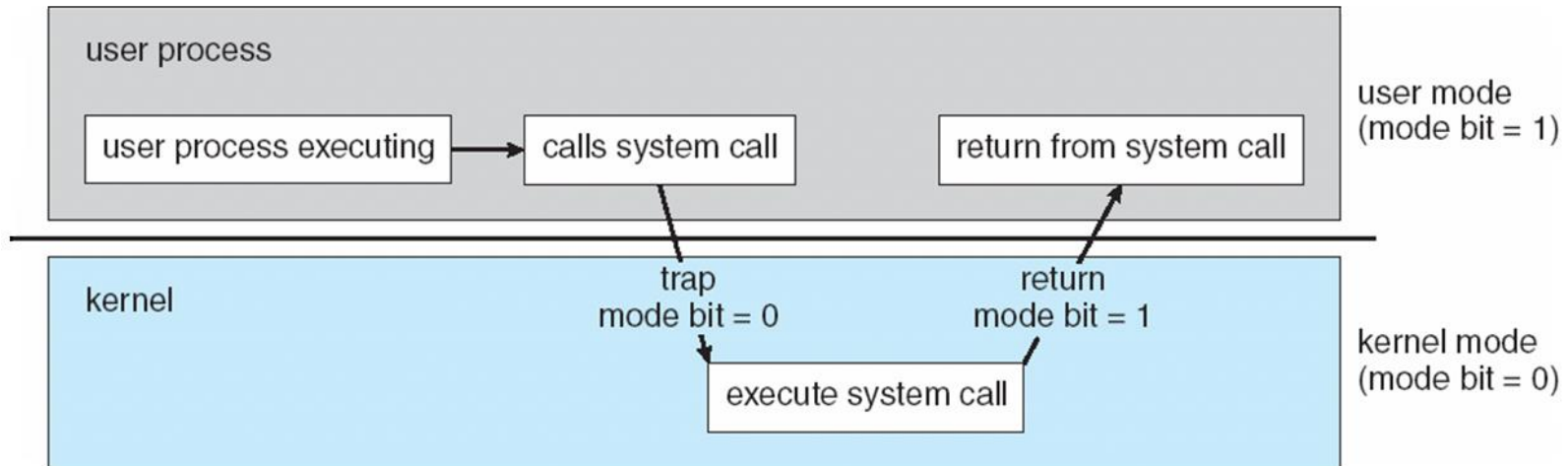
- الموقت لمنع حلقة لا حصر لها / عملية الموارد

\* ضبط المقاطعة بعد فترة محددة

\* نظام التشغيل التناقص العداد

\* عندما يولد عداد صفر مقاطعة

\* الإعداد قبل جدولة العملية لاستعادة التحكم أو إنهاء البرنامج الذي تجاوز الوقت المخصص





# Process Management

---

- A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads





# ادارة العمليات

- العملية عبارة عن برنامج قيد التنفيذ. إنها وحدة عمل داخل النظام. البرنامج كيان سلبي ، العملية كيان نشط.
- تحتاج العملية إلى موارد لإنجاز مهمتها
- \* وحدة المعالجة المركزية ، الذاكرة ، الإدخال / الإخراج ، الملفات.
- \* بيانات التهيئة
- إنهاء العملية يتطلب استعادة أي موارد قابلة لإعادة الاستخدام
- تحتوي العملية أحادية الخيوط على **عداد برنامج** واحد يحدد موقع التعليمات التالية للتنفيذ
- \* تنفذ العملية التعليمات بالتتابع ، واحدًا تلو الآخر ، حتى اكتمالها
- تحتوي العملية متعددة الخيوط على عداد برنامج واحد لكل مؤشر ترابط
- عادةً ما يحتوي النظام على العديد من العمليات ، بعضها مستخدم ، وبعض أنظمة التشغيل تعمل بشكل متزامن على وحدة معالجة مركزية واحدة أو أكثر
- \* التزامن عن طريق مضاعفة وحدات المعالجة المركزية بين العمليات / الخيوط





# Process Management Activities

---

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling





# أنشطة إدارة العمليات

نظام التشغيل مسؤول عن الأنشطة التالية فيما يتعلق بإدارة العملية:

- إنشاء وحذف عمليات المستخدم والنظام
- تعليق واستئناف العمليات
- توفير آليات لمزامنة العملية
- توفير آليات عملية الاتصال
- توفير آليات للتعامل مع المأزق





# Memory Management

---

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed





# إدارة الذاكرة

- جميع البيانات في الذاكرة قبل وبعد المعالجة
- جميع التعليمات في الذاكرة من أجل تنفيذها
- تحدد إدارة الذاكرة ما هو موجود في الذاكرة ومتى
- \* تحسين استخدام وحدة المعالجة المركزية واستجابة الكمبيوتر للمستخدمين
- أنشطة إدارة الذاكرة

- \* تتبع أي أجزاء من الذاكرة يتم استخدامها حالياً ومن قبل من
- \* تحديد العمليات (أو أجزاء منها) والبيانات المراد نقلها إلى الذاكرة وخارجها
- \* تخصيص مساحة الذاكرة وإلغاء تخصيصها حسب الحاجة







# Storage Management

---

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - 4 Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - 4 Creating and deleting files and directories
    - 4 Primitives to manipulate files and dirs
    - 4 Mapping files onto secondary storage
    - 4 Backup files onto stable (non-volatile) storage media







# Mass-Storage Management

---

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed
  - Varies between WORM (write-once, read-many-times) and RW (read-write)







# Performance of Various Levels of Storage

- Movement between levels of storage hierarchy can be explicit or implicit

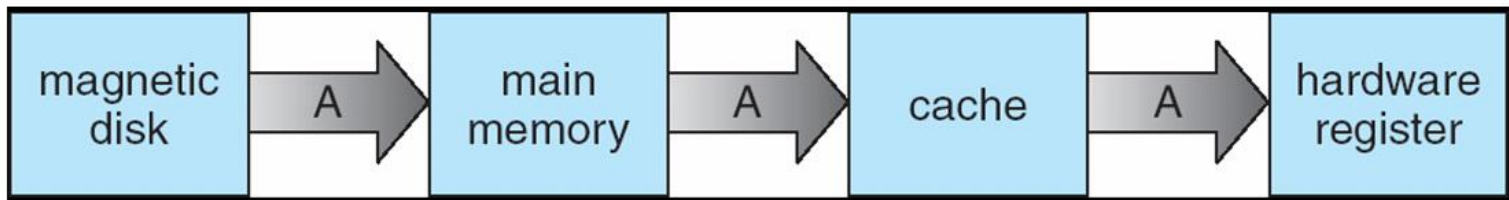
Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape





# Migration of Integer A from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



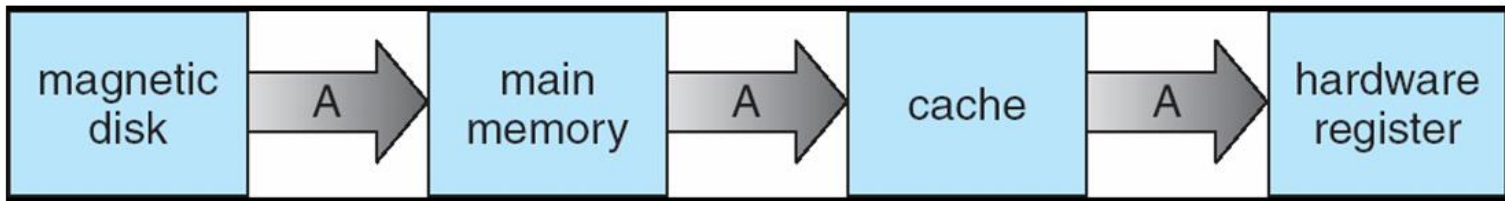
- Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Chapter 17





## من القرص إلى التسجيل A ترحيل عدد صحيح

- يجب أن تكون البيئات متعددة المهام حريصة على استخدام أحدث قيمة ، بغض النظر عن مكان تخزينها في التسلسل الهرمي للتخزين



- يجب أن توفر بيئة المعالجات المتعددة اتساق ذاكرة التخزين المؤقت في الأجهزة بحيث يكون لجميع وحدات المعالجة المركزية أحدث قيمة في ذاكرة التخزين المؤقت الخاصة بها
- حالة البيئة الموزعة أكثر تعقيدًا

\* يمكن أن توجد عدة نسخ من مرجع  
\* تم تناول حلول مختلفة في الفصل 17





# I/O Subsystem

---

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices







# نظام الإدخال / الإخراج الفرعي

- يتمثل أحد أغراض نظام التشغيل في إخفاء خصائص الأجهزة عن المستخدم
- الإدخال / الإخراج الفرعي المسؤول عن:

\* إدارة ذاكرة الإدخال / الإخراج بما في ذلك التخزين المؤقت (تخزين البيانات مؤقتًا أثناء نقلها) والتخزين المؤقت (تخزين أجزاء من البيانات في تخزين أسرع للأداء) والتخزين المؤقت (تداخل إخراج وظيفة واحدة مع إدخال وظائف أخرى)

\* واجهة برنامج تشغيل الجهاز العامة

\* برامج تشغيل لأجهزة معينة



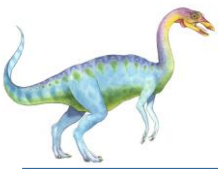


# Protection and Security

---

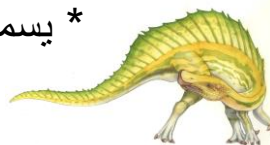
- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights





# الحماية والأمن

- **الحماية** - أي آلية للتحكم في وصول العمليات أو المستخدمين إلى الموارد المحددة بواسطة نظام التشغيل
- **الأمن** - الدفاع عن النظام ضد الهجمات الداخلية والخارجية
- \* نطاق ضخم ، بما في ذلك رفض الخدمة ، والديدان ، والفيروسات ، وسرقة الهوية ، وسرقة الخدمة
- تميز الأنظمة بشكل عام أولاً بين المستخدمين ، لتحديد من يمكنه فعل ماذا
- \* تتضمن هويات المستخدم (**معرفات المستخدم** ومعرفات الأمان) الاسم والرقم المرتبط به ، واحد لكل مستخدم
- \* يتم ربط معرف المستخدم بعد ذلك بجميع الملفات والعمليات الخاصة بهذا المستخدم لتحديد التحكم في الوصول
- \* معرف المجموعة (**معرف المجموعة**) مجموعة من المستخدمين يتم تحديدها والتحكم في إدارتها ، ثم ربطها أيضاً بكل عملية ، ملف
- \* يسمح **تصعيد الامتياز** للمستخدم بالتغيير إلى معرف فعال بمزيد من الحقوق





# Computing Environments

---

- Traditional computer
  - Blurring over time
  - Office environment
    - 4 PCs connected to a network, terminals attached to mainframe or minicomputers providing batch and timesharing
    - 4 Now portals allowing networked and remote systems access to same resources
  - Home networks
    - 4 Used to be single system, then modems
    - 4 Now firewalled, networked





# بيئات الحوسبة

- كمبيوتر تقليدي

\* التعقيم بمرور الوقت

\* مكتب البيئة

**4** أجهزة كمبيوتر متصلة بشبكة أو أطراف توصيل متصلة بإطار مركزي أو أجهزة كمبيوتر صغيرة توفر الدفعات ومشاركة الوقت

**4** الآن بوابات تتيح للأنظمة المتصلة بالشبكة والبعيدة الوصول إلى نفس الموارد

\* الشبكات المنزلية

**4** يستخدم ليكون نظامًا واحدًا ، ثم أجهزة المودم

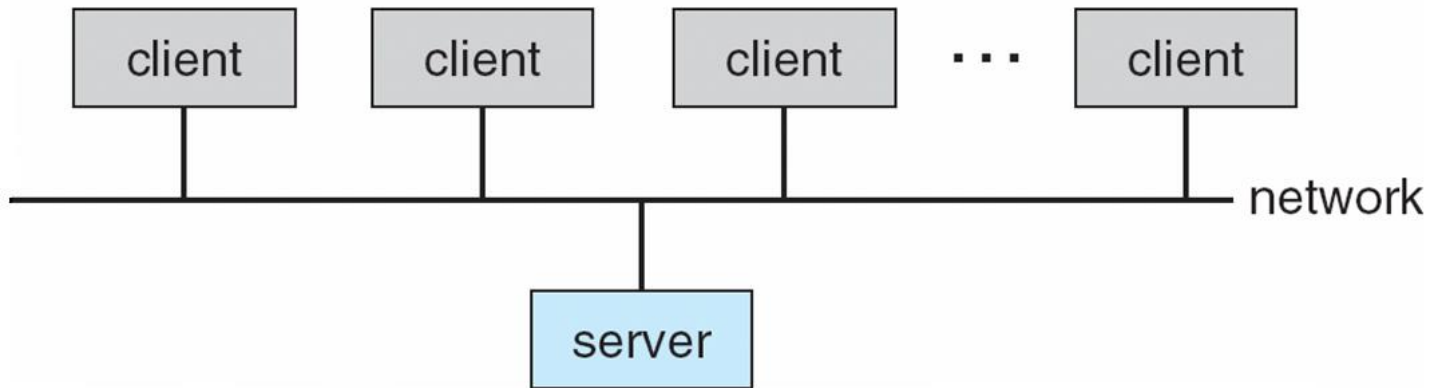
**4** الآن بجدار حماية ، متصل بالشبكة





# Computing Environments (Cont)

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - 4 **Compute-server** provides an interface to client to request services (i.e. database)
    - 4 **File-server** provides interface for clients to store and retrieve files





# بيئات الحوسبة (تابع)

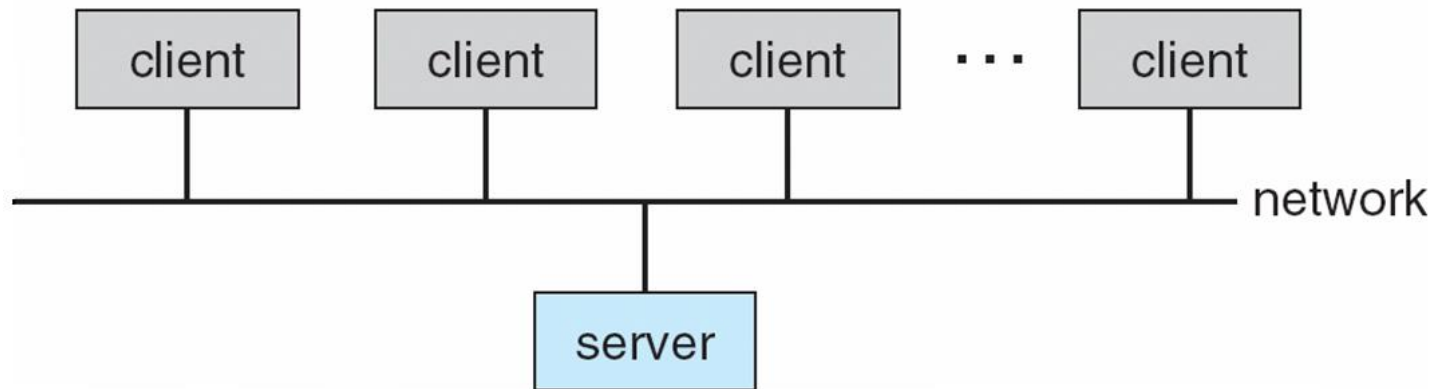
- حوسبة خادم العميل

\* المحطات البكمية تحل محلها أجهزة الكمبيوتر الذكية

\* العديد من الأنظمة الآن خوادم تستجيب لطلبات العملاء

4 يوفر خادم الحوسبة واجهة للعميل لطلب الخدمات (أي قاعدة البيانات)

4 يوفر خادم الملفات واجهة للعملاء لتخزين واسترداد الملفات





# Peer-to-Peer Computing

---

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - 4 Registers its service with central lookup service on network, or
    - 4 Broadcast request for service and respond to requests for service via **discovery protocol**
- Examples include *Napster* and *Gnutella*







# الحوسبة من نظير إلى نظير

- نموذج آخر للنظام الموزع

- لا يميز P2P العملاء والخوادم

\* بدلاً من ذلك ، تعتبر جميع العقد نظيرة

\* يجوز لكل منهما أن يعمل كعميل أو خادم أو كليهما

\* يجب أن تنضم العقدة إلى شبكة P2P

4 يسجل خدمته مع خدمة البحث المركزي على الشبكة ، أو

4 طلب البث للخدمة والاستجابة لطلبات الخدمة عبر بروتوكول الاكتشاف

4 تشمل الأمثلة نابستر ونوتيل





# Web-Based Computing

---

- Web has become ubiquitous
- PCs most prevalent devices
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers:  
**load balancers**
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers





# الحوسبة على شبكة الإنترنت

- أصبح الويب في كل مكان
- أجهزة الكمبيوتر أكثر الأجهزة انتشارا
- يتم ربط المزيد من الأجهزة بالشبكة للسماح بالوصول إلى الويب
- فئة جديدة من الأجهزة لإدارة حركة مرور الويب بين الخوادم المماثلة: موازين التحميل
- تطور استخدام أنظمة التشغيل مثل Windows 95 ، من جانب العميل ، إلى Linux و Windows XP ، والتي يمكن أن تكون عملاء وخوادم





# Open-Source Operating Systems

---

- Operating systems made available in source-code format rather than just binary [closed-source](#)
- Counter to the [copy protection](#) and [Digital Rights Management \(DRM\)](#) movement
- Started by [Free Software Foundation \(FSF\)](#), which has “copyleft” [GNU Public License \(GPL\)](#)
- Examples include [GNU/Linux](#), [BSD UNIX](#) (including core of [Mac OS X](#)), and [Sun Solaris](#)





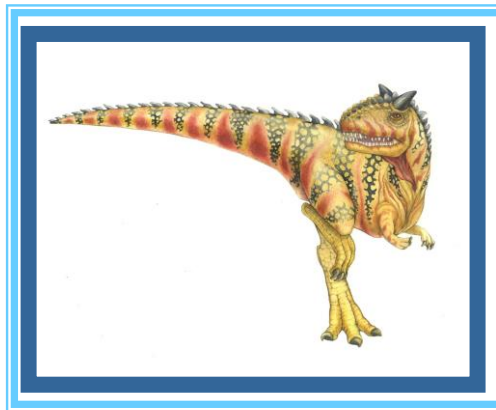
# أنظمة تشغيل مفتوحة المصدر

- إتاحة أنظمة التشغيل بتنسيق شفرة المصدر بدلاً من مجرد مصدر ثنائي مغلق
- مواجهة لحركة حماية النسخ وإدارة الحقوق الرقمية (DRM)
- بدأت من قبل مؤسسة البرمجيات الحرة (FSF)، التي لديها "حقوق متروكة" رخصة GNU العمومية (GPL)
- تتضمن الأمثلة GNU / Linux و BSD UNIX بما في ذلك جوهر نظام (Sun Solaris و Mac OS X)



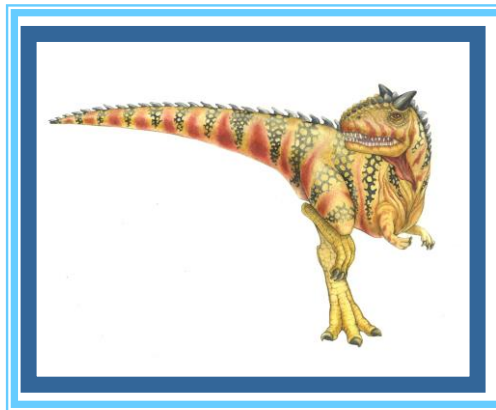
# End of Chapter 1

---

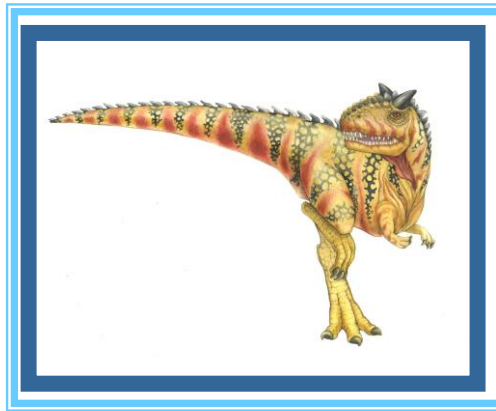


# Chapter 3: Processes operating System

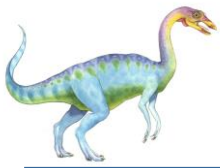
---



# الفصل 3: العمليات نظم تشغيل







# Chapter 3: Processes

---

- Process Concept
- Process Scheduling
- Operations on Processes
- Interposes Communication
- Examples of IPC Systems
- Communication in Client-Server Systems





## الفصل 3: العمليات

---

- مفهوم العملية
- جدولة العملية
- العمليات على العمليات
- يفسر التواصل
- أمثلة على أنظمة IPC
- الاتصالات في أنظمة خادم العميل





# Process Concept

---

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms *job* and *process* almost interchangeably
- Process – a program in execution; process execution must progress in sequential fashion
- A process includes:
  - program counter
  - stack
  - data section





# مفهوم العملية

- \* ينفذ نظام التشغيل مجموعة متنوعة من البرامج:
- \* نظام الدفوعات - الوظائف
- \* أنظمة الوقت المشترك - برامج المستخدم أو المهام
- \* يستخدم الكتاب المدرسي مصطلحي الوظيفة والعملية بالتبادل تقريباً
- \* عملية - برنامج في التنفيذ. يجب أن يتقدم تنفيذ العملية بطريقة متسلسلة
- العملية تشمل:

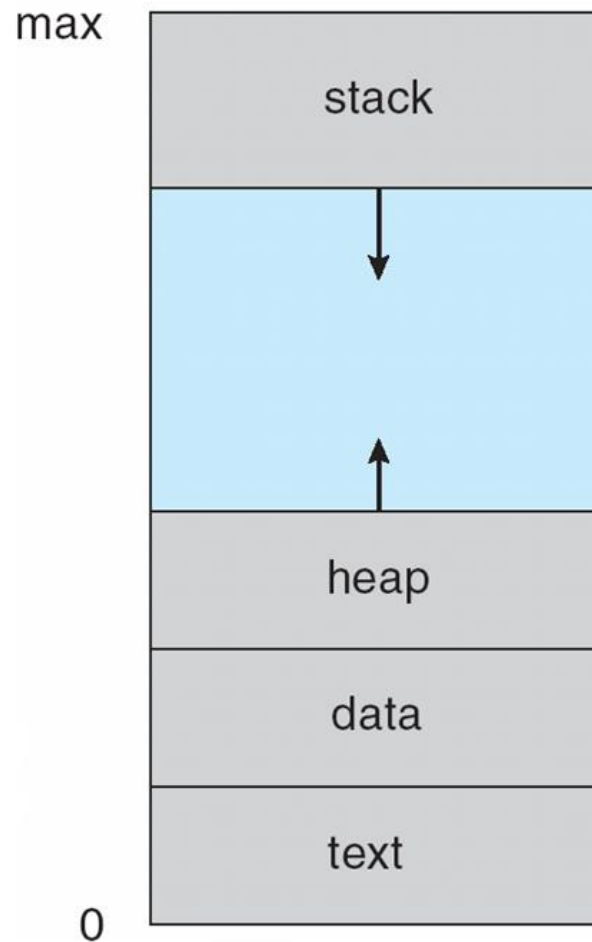
- \* برنامج العداد
- \* مكدس.
- \* قسم البيانات





# Process in Memory

العملية في الذاكرة





# Process State

---

- As a process executes, it changes *state*
  - **new**: The process is being created
  - **running**: Instructions are being executed
  - **waiting**: The process is waiting for some event to occur
  - **ready**: The process is waiting to be assigned to a processor
  - **terminated**: The process has finished execution





# حالة العملية

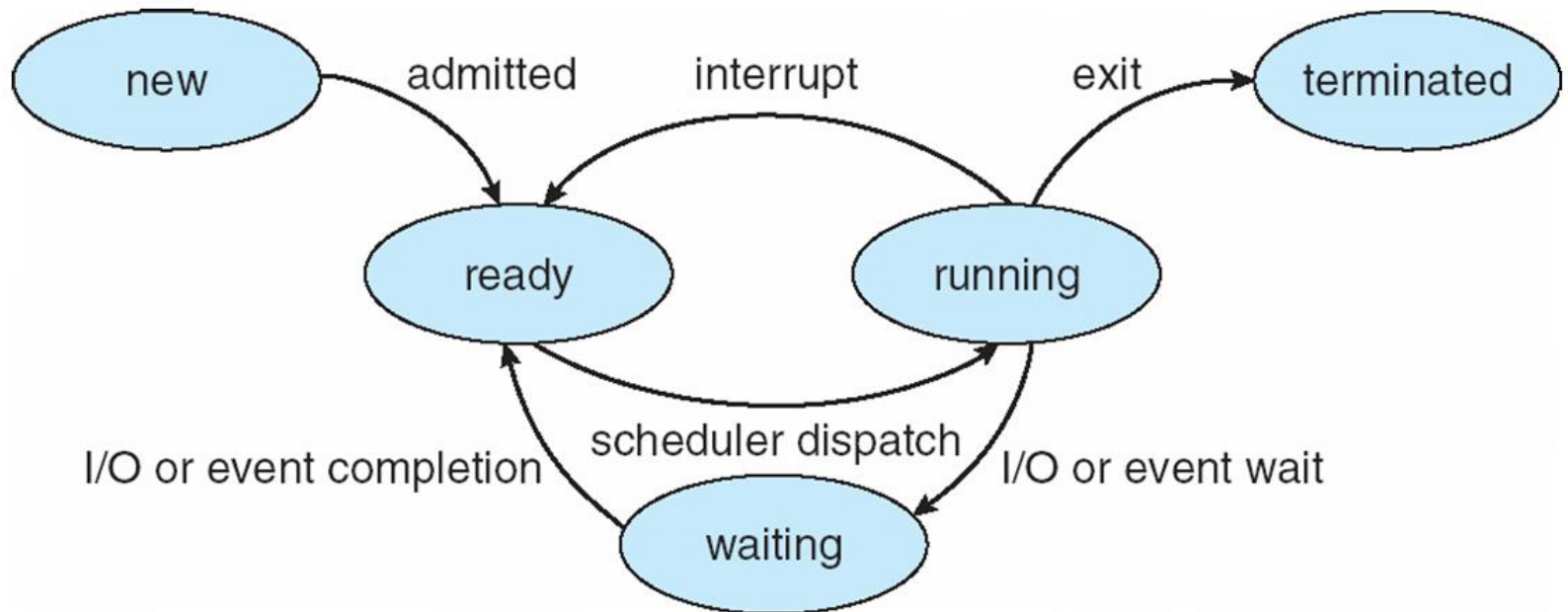
- عندما يتم تنفيذ العملية ، فإنها تغير الحالة
- جديد: العملية قيد الإنشاء
- قيد التشغيل: يتم تنفيذ التعليمات
- الانتظار: العملية تنتظر حدوث بعض الأحداث
- جاهز: العملية تنتظر تعيينها إلى معالج
- منتهية: انتهت العملية من التنفيذ





# Diagram of Process State

## رسم تخطيطي لحالة العملية







# Process Control Block (PCB)

---

Information associated with each process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information





# كتلة التحكم في العمليات (PCB)

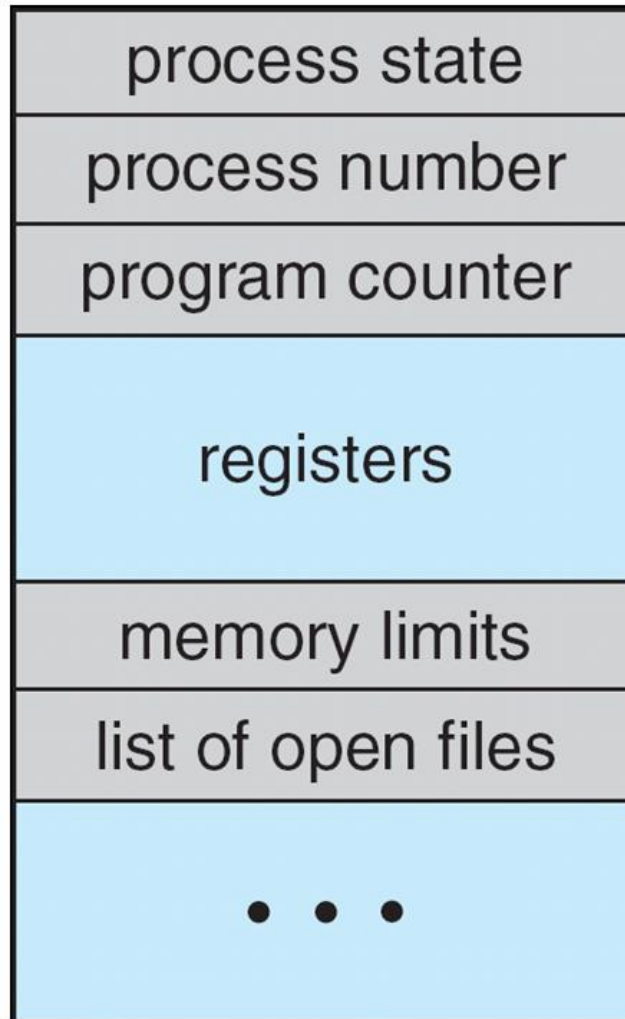
المعلومات المرتبطة بكل عملية

- حالة العملية
- عداد البرنامج
- سجلات وحدة المعالجة المركزية
- معلومات جدول وحدة المعالجة المركزية
- معلومات إدارة الذاكرة
- المعلومات المحاسبية
- معلومات حالة الإدخال / الإخراج





# Process Control Block (PCB)



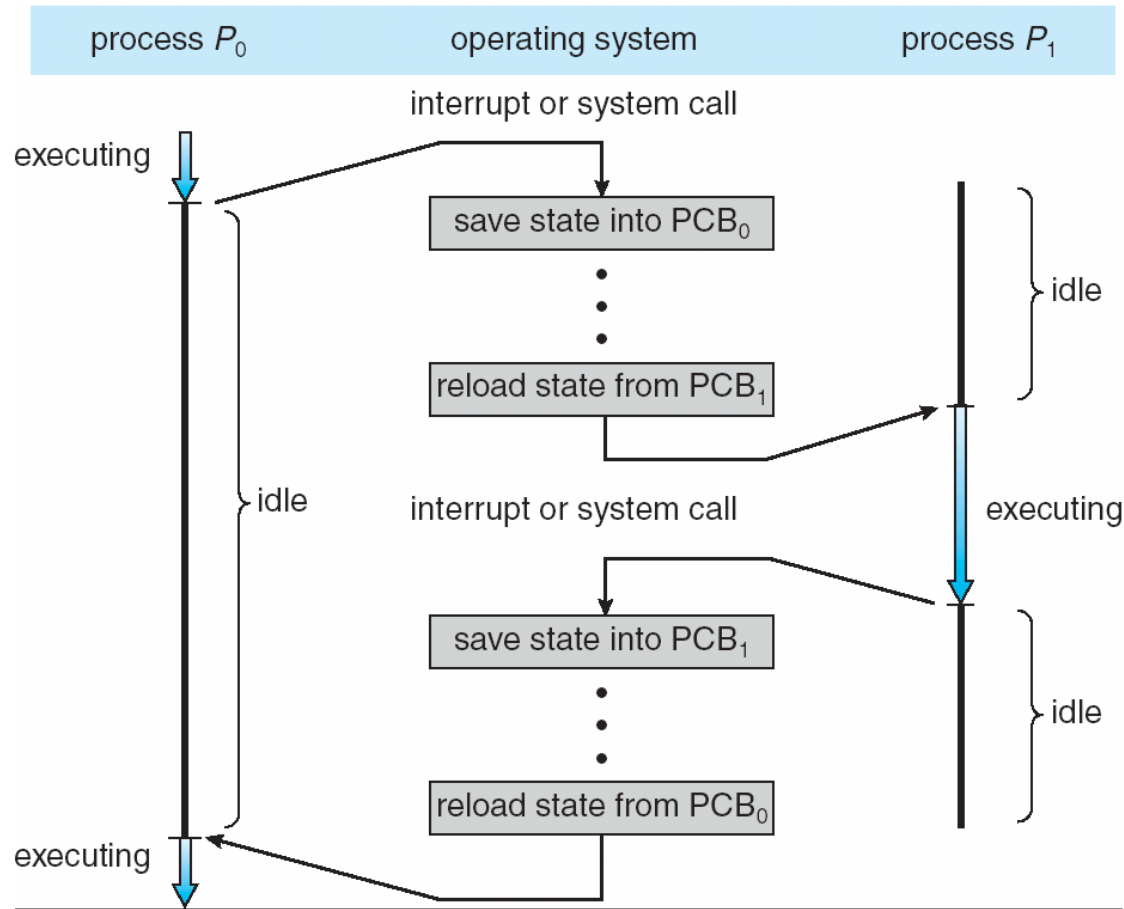
كتلة التحكم في العمليات





# CPU Switch From Process to Process

## تبدیل وحدة المعالجة المركزية من عملية إلى عملية





# Process Scheduling Queues

---

- **Job queue** – set of all processes in the system
- **Ready queue** – set of all processes residing in main memory, ready and waiting to execute
- **Device queues** – set of processes waiting for an I/O device
- Processes migrate among the various queues





# قوائم انتظار جدولة العملية

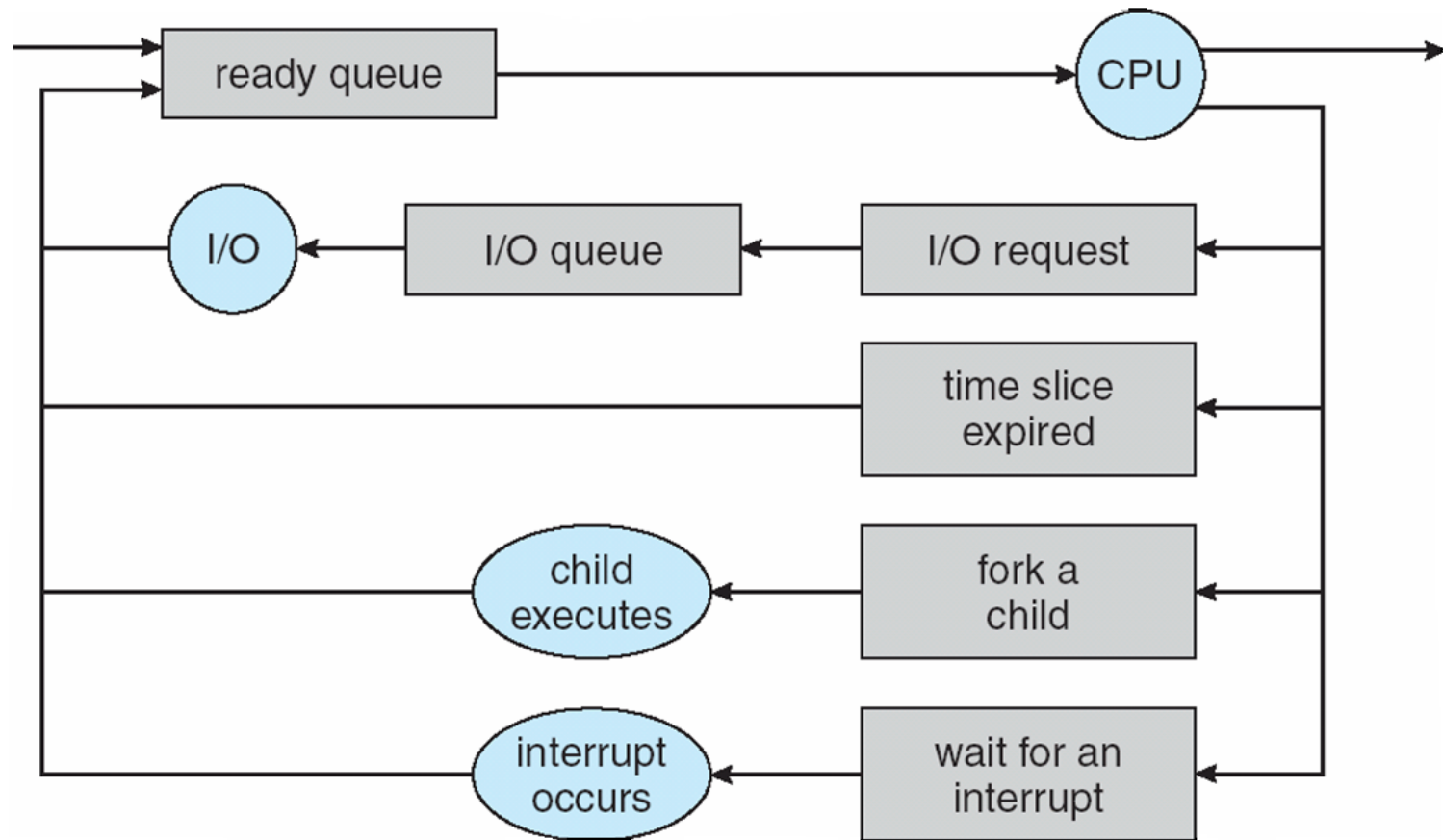
- قائمة انتظار الوظائف - مجموعة من جميع العمليات في النظام
- قائمة انتظار جاهزة - مجموعة من جميع العمليات الموجودة في الذاكرة الرئيسية ، جاهزة وتنتظر التنفيذ
- قوائم انتظار الجهاز - مجموعة من العمليات في انتظار جهاز الإدخال / الإخراج
- ترحيل العمليات بين قوائم الانتظار المختلفة





# Representation of Process Scheduling

## تمثيل عملية الجدولة





# Schedulers

---

- **Long-term scheduler** (or job scheduler) – selects which processes should be brought into the ready queue
- **Short-term scheduler** (or CPU scheduler) – selects which process should be executed next and allocates CPU







# المنظمون

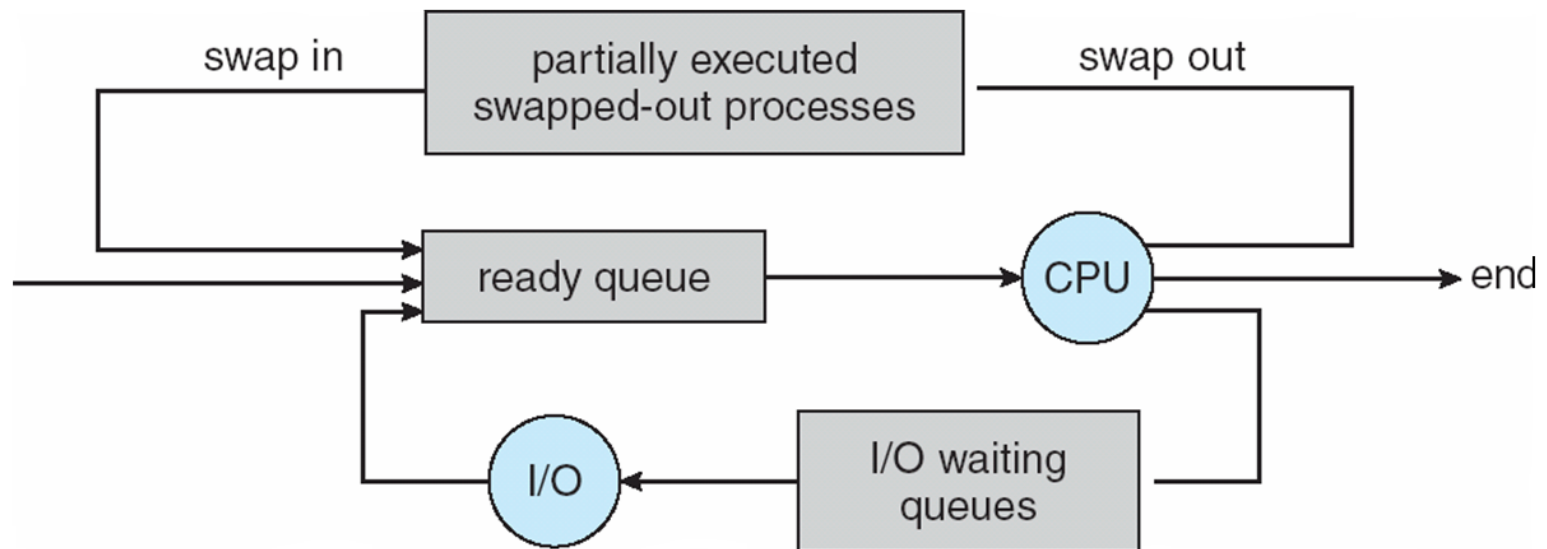
- جدولة طويلة الأجل (أو مجدول عمل) - يحدد العمليات التي يجب إحضارها إلى قائمة الانتظار الجاهزة
- جدولة قصيرة المدى (أو جدولة وحدة المعالجة المركزية) - يختار العملية التي يجب تنفيذها بعد ذلك ويخصص وحدة المعالجة المركزية





# Addition of Medium Term Scheduling

## إضافة جدولة متوسطة المدى





# Schedulers (Cont)

---

- Short-term scheduler is invoked very frequently (milliseconds)  $\Rightarrow$  (must be fast)
- Long-term scheduler is invoked very infrequently (seconds, minutes)  $\Rightarrow$  (may be slow)
- The long-term scheduler controls the *degree of multiprogramming*
- Processes can be described as either:
  - **I/O-bound process** – spends more time doing I/O than computations, many short CPU bursts
  - **CPU-bound process** – spends more time doing computations; few very long CPU bursts





## المنظمون (تابع)

- يتم استدعاء جدولة المدى القصير بشكل متكرر (مللي ثانية)  $\Rightarrow$  (يجب أن يكون سريعًا)
- يتم استدعاء جدولة طويلة المدى بشكل غير متكرر (ثوانٍ ، دقائق)  $\Rightarrow$  (قد يكون بطيئًا)
- يتحكم المجدول طويل المدى في درجة البرمجة المتعددة
- يمكن وصف العمليات على أنها إما:
  - \* عملية مرتبطة بالإدخال / الإخراج - تقضي وقتًا أطول في إجراء الإدخال / الإخراج مقارنة بالعمليات الحسابية ، والعديد من دفعات وحدة المعالجة المركزية القصيرة
  - \* عملية مرتبطة بوحدة المعالجة المركزية - تقضي وقتًا أطول في إجراء الحسابات ؛ بضع دفعات طويلة جدًا من وحدة المعالجة المركزية





# Context Switch

---

- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a **context switch**
- **Context** of a process represented in the PCB
- Context-switch time is overhead; the system does no useful work while switching
- Time dependent on hardware support





# تبديل السياق

- عندما تتحول وحدة المعالجة المركزية إلى عملية أخرى ، يجب على النظام حفظ حالة العملية القديمة وتحميل الحالة المحفوظة للعملية الجديدة عبر مفتاح السياق
- سياق عملية ممثلة في pcb
- وقت تبديل السياق هو النفقات العامة ؛ لا يقوم النظام بعمل مفيد أثناء التبديل
- يعتمد الوقت على دعم الأجهزة





# Process Creation

---

- **Parent** process create **children** processes, which, in turn create other processes, forming a tree of processes
- Generally, process identified and managed via a **process identifier (pid)**
- Resource sharing
  - Parent and children share all resources
  - Children share subset of parent's resources
  - Parent and child share no resources
- Execution
  - Parent and children execute concurrently
  - Parent waits until children terminate



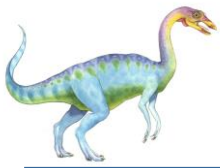


# إنشاء العملية

- **Parent process** تخلق **children processes**، والتي بدورها تخلق عمليات أخرى ، وتشكل شجرة من العمليات
- بشكل عام ، يتم تحديد العملية وإدارتها عبر معرف العملية (PID)
- تقاسم الموارد
- \* يتشارك **Parent** و **children** في جميع الموارد
- \* يشارك **children** مجموعة فرعية من موارد **Parent**
- \* لا يتشارك **Parent** و **children** في الموارد
- التنفيذ
- \* يقوم **Parent** و **children** بالتنفيذ المتزامن
- \* **Parent** ينتظر حتى ينتهي **children**







# Process Termination

---

- Process executes last statement and asks the operating system to delete it (**exit**)
  - Output data from child to parent (via **wait**)
  - Process' resources are deallocated by operating system
- Parent may terminate execution of children processes (**abort**)
  - Child has exceeded allocated resources
  - Task assigned to child is no longer required
  - If parent is exiting
    - 4 Some operating system do not allow child to continue if its parent terminates
      - All children terminated - **cascading termination**





# إنهاء العملية

- العملية تنفذ البيان الأخير وتطلب من نظام التشغيل حذفها (خروج)
- \* بيانات الإخراج من **children** إلى **Parent** (عن طريق الانتظار)
- \* يتم إلغاء تخصيص موارد العملية بواسطة نظام التشغيل
- يجوز ل **Parent** إنهاء تنفيذ عمليات **children** (إيقاف)
  - \* تجاوز **children** للموارد المخصصة
  - \* لم تعد المهمة المعينة ل **children** مطلوبة
  - \* في حالة إنهاء **Parent**
  - \* بعض أنظمة التشغيل لا تسمح ل **children** بالمتابعة إذا تم إنهاء الأصل
  - تم إنهاء جميع **children** - إنهاء متتالي





# Interprocess Communication

---

- Processes within a system may be **independent** or **cooperating**
- Cooperating process can affect or be affected by other processes, including sharing data
- Reasons for cooperating processes:
  - Information sharing
  - Computation speedup
  - Modularity
  - Convenience
- Cooperating processes need **interprocess communication (IPC)**
- Two models of IPC
  - Shared memory
  - Message passing





# الاتصال بين العمليات

- قد تكون العمليات داخل النظام مستقلة أو متعاونة
- يمكن أن تؤثر العملية التعاونية أو تتأثر بعمليات أخرى ، بما في ذلك مشاركة البيانات
- أسباب التعاون:

\* مشاركة المعلومات

\* تسريع الحساب

\* نمطية

\* السهولة أو الراحة

- تحتاج العمليات المتعاونة إلى اتصال بين العمليات (IPC)

- نموذجان من IPC

\* مشاركة الذكريات

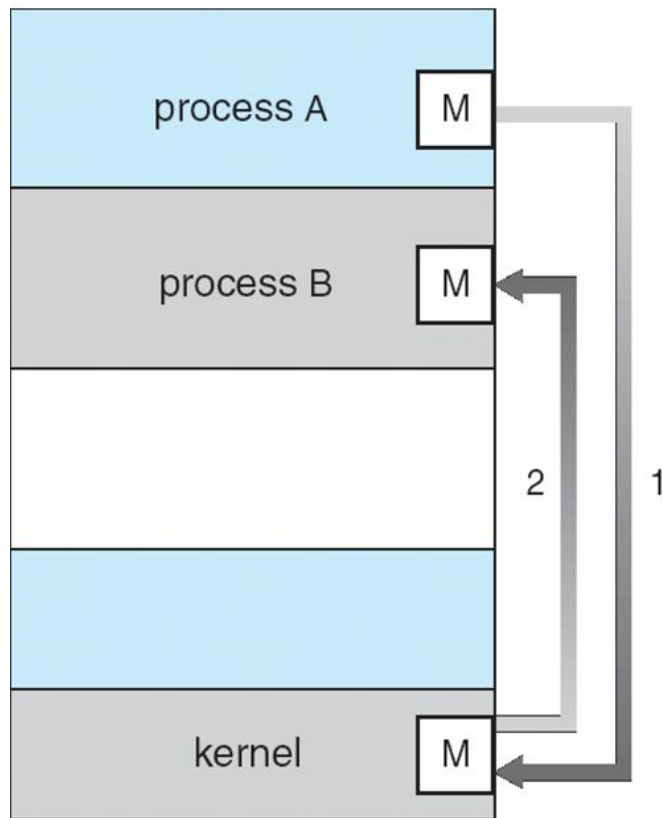
\* تمرير الرسالة



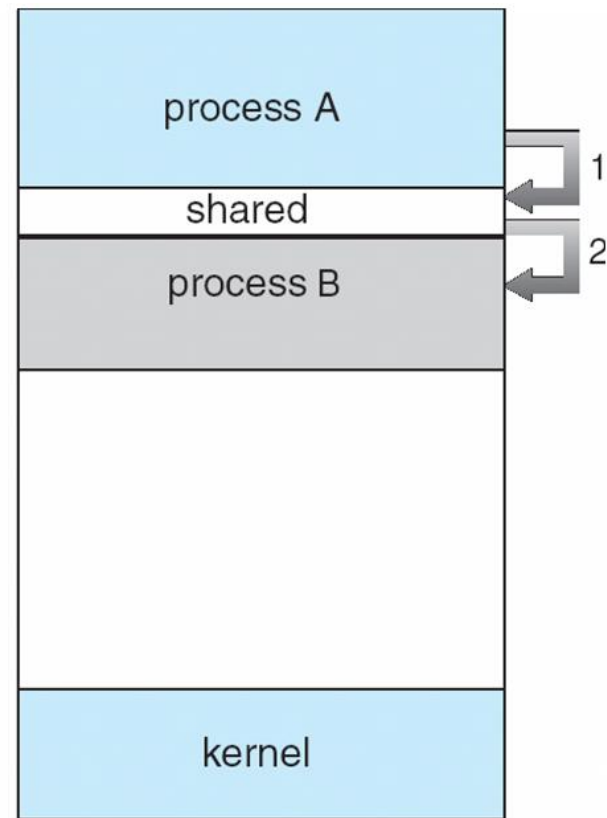


# Communications Models

## نماذج الاتصالات



(a)



(b)





# Cooperating Processes

---

- **Independent** process cannot affect or be affected by the execution of another process
- **Cooperating** process can affect or be affected by the execution of another process
- Advantages of process cooperation
  - Information sharing
  - Computation speed-up
  - Modularity
  - Convenience





# العمليات المتعاونة

- لا يمكن أن تؤثر العملية المستقلة أو تتأثر بتنفيذ عملية أخرى
- يمكن أن تؤثر عملية التعاون أو تتأثر بتنفيذ عملية أخرى
- مزايا عملية التعاون
  - \* مشاركة المعلومات
  - \* تسريع الحساب
  - \* نمطية
  - \* السهولة أو الراحة





# Producer-Consumer Problem

---

- Paradigm for cooperating processes, *producer* process produces information that is consumed by a *consumer* process
  - *unbounded-buffer* places no practical limit on the size of the buffer
  - *bounded-buffer* assumes that there is a fixed buffer size







# مشكلة المنتج والمستهلك

- نموذج للعمليات المتعاونة ، تنتج عملية المنتج المعلومات التي تستهلكها عملية المستهلك

\* لا يضع المخزن المؤقت غير المحدود أي حد عملي على حجم المخزن المؤقت

\* يفترض المخزن المؤقت المحدد وجود حجم مخزن مؤقت ثابت





# Interprocess Communication – Message Passing

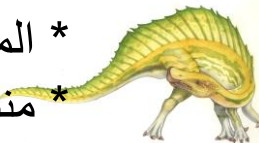
- Mechanism for processes to communicate and to synchronize their actions
- Message system – processes communicate with each other without resorting to shared variables
- IPC facility provides two operations:
  - **send**(*message*) – message size fixed or variable
  - **receive**(*message*)
- If  $P$  and  $Q$  wish to communicate, they need to:
  - establish a *communication link* between them
  - exchange messages via send/receive
- Implementation of communication link
  - physical (e.g., shared memory, hardware bus)
  - logical (e.g., logical properties)





# الاتصال بين العمليات - تمرير الرسائل

- آلية لعمليات التواصل ومزامنة أفعالهم
- نظام الرسائل - تتواصل العمليات مع بعضها البعض دون اللجوء إلى المتغيرات المشتركة
- مرفق IPC يوفر عمليتين:
  - \* إرسال (رسالة) - حجم الرسالة ثابت أو متغير
  - \* تلقي (رسالة)
- إذا رغب P و Q في التواصل ، فهما بحاجة إلى:
  - \* إقامة رابط اتصال بينهما
  - \* تبادل الرسائل عبر إرسال / استقبال
- تنفيذ رابط الاتصال
  - \* المادية (على سبيل المثال ، الذاكرة المشتركة ، ناقل الأجهزة)
  - \* منطقي (مثل الخصائص المنطقية)





# Implementation Questions

---

- How are links established?
- Can a link be associated with more than two processes?
- How many links can there be between every pair of communicating processes?
- What is the capacity of a link?
- Is the size of a message that the link can accommodate fixed or variable?
- Is a link unidirectional or bi-directional?





# أسئلة التنفيذ

- كيف يتم إنشاء الروابط؟
- هل يمكن ربط ارتباط بأكثر من عمليتين؟
- كم عدد الروابط التي يمكن أن توجد بين كل زوج من عمليات الاتصال؟
- ما هي سعة الارتباط؟
- هل حجم الرسالة التي يمكن للرباط أن يتسع لها ثابت أم متغير؟
- هل الرباط أحادي الاتجاه أم ثنائي الاتجاه؟





# Direct Communication

---

- Processes must name each other explicitly:
  - **send** ( $P$ ,  $message$ ) – send a message to process  $P$
  - **receive**( $Q$ ,  $message$ ) – receive a message from process  $Q$
- Properties of communication link
  - Links are established automatically
  - A link is associated with exactly one pair of communicating processes
  - Between each pair there exists exactly one link
  - The link may be unidirectional, but is usually bi-directional





# الاتصال المباشر

- يجب أن تسمى العمليات بعضها البعض صراحة:

\* إرسال ( P ، رسالة ) - إرسال رسالة لعملية P  
\* تلقي ( Q ، رسالة ) - تلقي رسالة من العملية Q

- خصائص ارتباط الاتصال

\* يتم إنشاء الروابط تلقائيًا  
\* يرتبط الارتباط بزوج واحد من عمليات الاتصال  
\* يوجد رابط واحد بالضبط بين كل زوج  
\* قد يكون الارتباط أحادي الاتجاه ، ولكنه عادة ما يكون ثنائي الاتجاه





# Indirect Communication

---

- Messages are directed and received from mailboxes (also referred to as ports)
  - Each mailbox has a unique id
  - Processes can communicate only if they share a mailbox
- Properties of communication link
  - Link established only if processes share a common mailbox
  - A link may be associated with many processes
  - Each pair of processes may share several communication links
  - Link may be unidirectional or bi-directional







# التواصل غير المباشر

- يتم توجيه الرسائل واستلامها من صناديق البريد (يشار إليها أيضًا باسم المنافذ)

\* لكل صندوق بريد معرف فريد  
\* يمكن للعمليات التواصل فقط إذا كانت تشترك في صندوق بريد

- خصائص ارتباط الاتصال

\* تم إنشاء الارتباط فقط إذا كانت العمليات تشترك في صندوق بريد مشترك  
\* قد يرتبط ارتباط بالعديد من العمليات  
\* قد تشترك كل زوج من العمليات في عدة روابط اتصال  
\* قد يكون الارتباط أحادي الاتجاه أو ثنائي الاتجاه





# Indirect Communication

---

- Operations
  - create a new mailbox
  - send and receive messages through mailbox
  - destroy a mailbox
- Primitives are defined as:  
**send**(*A, message*) – send a message to mailbox *A*  
**receive**(*A, message*) – receive a message from mailbox *A*





# التواصل غير المباشر

- عمليات

- \* إنشاء صندوق بريد جديد
- \* إرسال واستقبال الرسائل من خلال صندوق البريد تدمير صندوق البريد

- يتم تعريف الأوليات على النحو التالي:

إرسال (A، رسالة) - إرسال رسالة إلى صندوق البريد A  
استقبال (A، رسالة) - تلقي رسالة من صندوق البريد A





# Indirect Communication

---

- Mailbox sharing
  - $P_1$ ,  $P_2$ , and  $P_3$  share mailbox A
  - $P_1$ , sends;  $P_2$  and  $P_3$  receive
  - Who gets the message?
- Solutions
  - Allow a link to be associated with at most two processes
  - Allow only one process at a time to execute a receive operation
  - Allow the system to select arbitrarily the receiver. Sender is notified who the receiver was.





# التواصل غير المباشر

- مشاركة صندوق البريد:

\* P1, P2, and P3 share mailbox A

\* P1, sends; P2 and P3 receive

\* من يتلقى الرسالة؟

- حلول

\* السماح بربط ارتباط بعمليتين على الأكثر

\* السماح بعملية واحدة فقط في كل مرة لتنفيذ عملية استلام

\* اسمح للنظام بتحديد جهاز الاستقبال بشكل تعسفي. يتم إخطار المرسل من كان المتلقي.





# Synchronization

---

- Message passing may be either blocking or non-blocking
- **Blocking** is considered **synchronous**
  - **Blocking send** has the sender block until the message is received
  - **Blocking receive** has the receiver block until a message is available
- **Non-blocking** is considered **asynchronous**
  - **Non-blocking send** has the sender send the message and continue
  - **Non-blocking receive** has the receiver receive a valid message or null





# التزامن

- قد يكون تمرير الرسالة إما محظورًا أو غير محظور
- يعتبر الحظر متزامنًا
- \* يؤدي حظر الإرسال إلى حظر المرسل حتى يتم استلام الرسالة
- \* يؤدي حظر الاستلام إلى حظر المتلقي حتى تتوفر الرسالة
- يعتبر عدم الحظر غير متزامن.
- \* عدم حجب الإرسال يجعل المرسل يرسل الرسالة والمتابعة.
- \* عدم حجب الاستلام يتلقى المتلقي رسالة صالحة أو فارغة.





# Buffering

---

- Queue of messages attached to the link; implemented in one of three ways
  1. Zero capacity – 0 messages  
Sender must wait for receiver (rendezvous)
  2. Bounded capacity – finite length of  $n$  messages  
Sender must wait if link full
  3. Unbounded capacity – infinite length  
Sender never waits







# التخزين المؤقت

• قائمة انتظار الرسائل المرفقة بالرابط ؛ تم تنفيذها بإحدى الطرق الثلاث

1. السعة الصفريّة - 0 رسائل يجب على المرسل انتظار المتلقي (موعد)

2- السعة المحدودة - الطول المحدد لعدد  $n$  من الرسائل يجب على المرسل الانتظار إذا كان الارتباط ممتلئاً

3. السعة غير المحدودة - الطول اللانهائي لا ينتظر المرسل أبداً





# Communications in Client-Server Systems

- Sockets
- Remote Procedure Calls
- Remote Method Invocation (Java)

## الاتصالات في أنظمة خادم العميل

- المقابس.
- مكالمات الإجراءات عن بعد.
- استدعاء الطريقة عن بعد (جافا)





# Sockets

- A socket is defined as an *endpoint for communication*
- Concatenation of IP address and port
- The socket **161.25.19.8:1625** refers to port **1625** on host **161.25.19.8**
- Communication consists between a pair of sockets

## المقابس

\* يتم تعريف المقبس على أنه نقطة نهاية للاتصال

\* تسلسل عنوان IP والمنفذ

\* المقبس 161.25.19.8:1625 يشير إلى المنفذ 1625 على المضيف 161.25.19.8

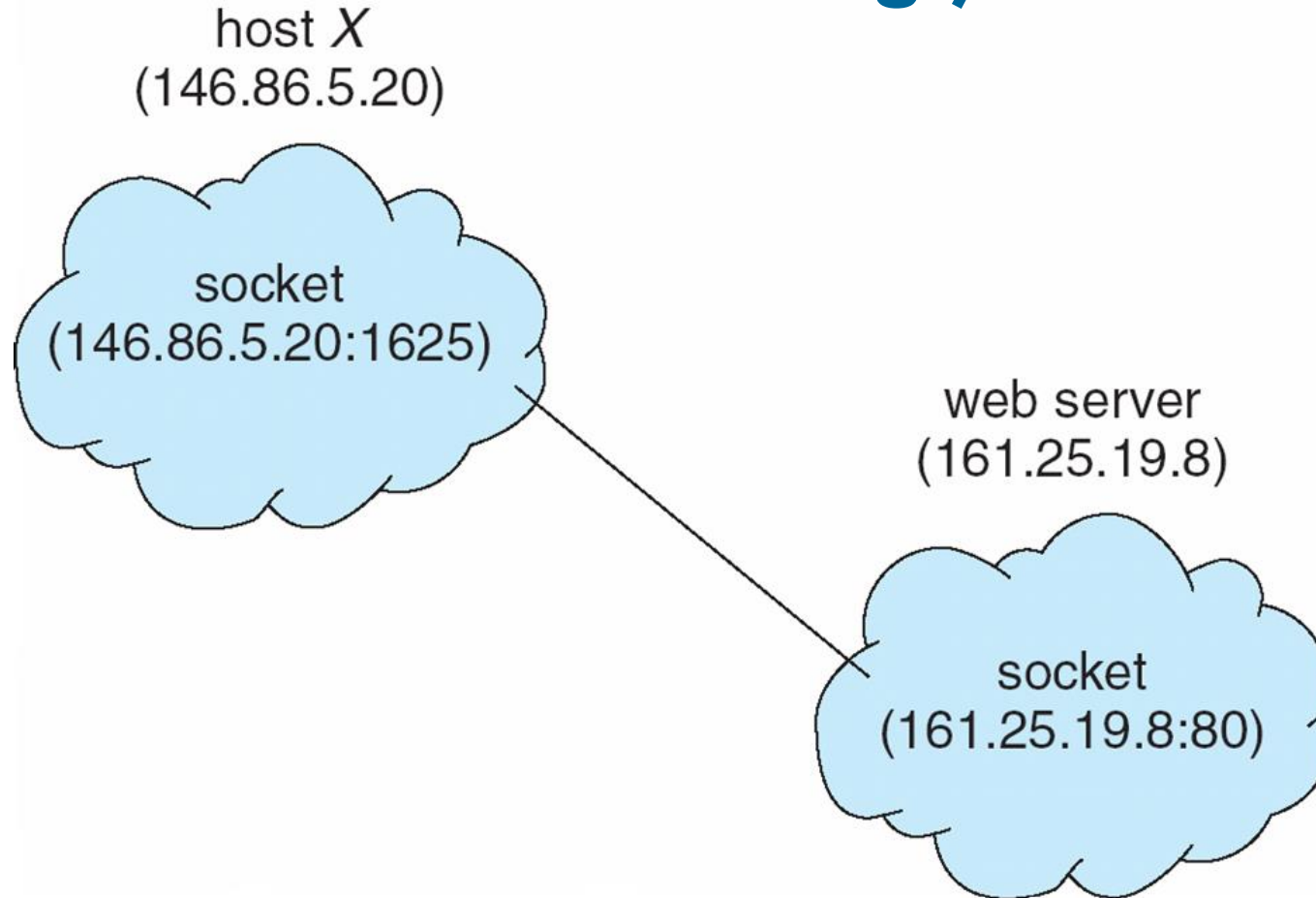
\* يتكون الاتصال بين زوج من المقابس





# Socket Communication

## اتصالات المقبس





# Remote Procedure Calls

---

- Remote procedure call (RPC) abstracts procedure calls between processes on networked systems
- **Stubs** – client-side proxy for the actual procedure on the server
- The client-side stub locates the server and *marshalls* the parameters
- The server-side stub receives this message, unpacks the marshalled parameters, and performs the procedure on the server





# استدعاءات الإجراءات عن بعد

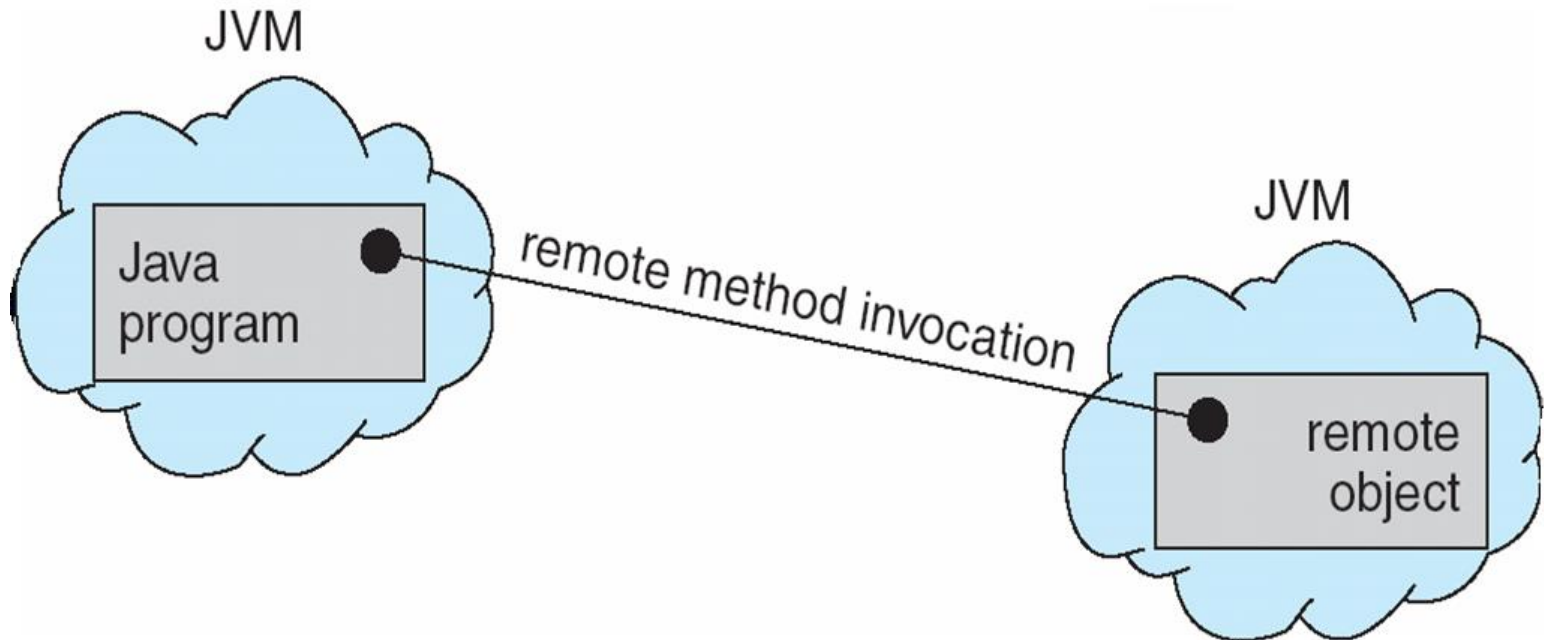
- استدعاء الإجراءات البعيد (RPC) يلخص استدعاءات الإجراءات بين العمليات على الأنظمة المتصلة بالشبكة
- stubs - الوكيل من جانب العميل للإجراء الفعلي على الخادم
- يحدد كعب العميل موقع الخادم وينظم المعلومات
- يتلقى كعب الخادم من جانب الخادم هذه الرسالة ، ويفكك المعلومات المنظمة ، وينفذ الإجراء على الخادم





# Remote Method Invocation

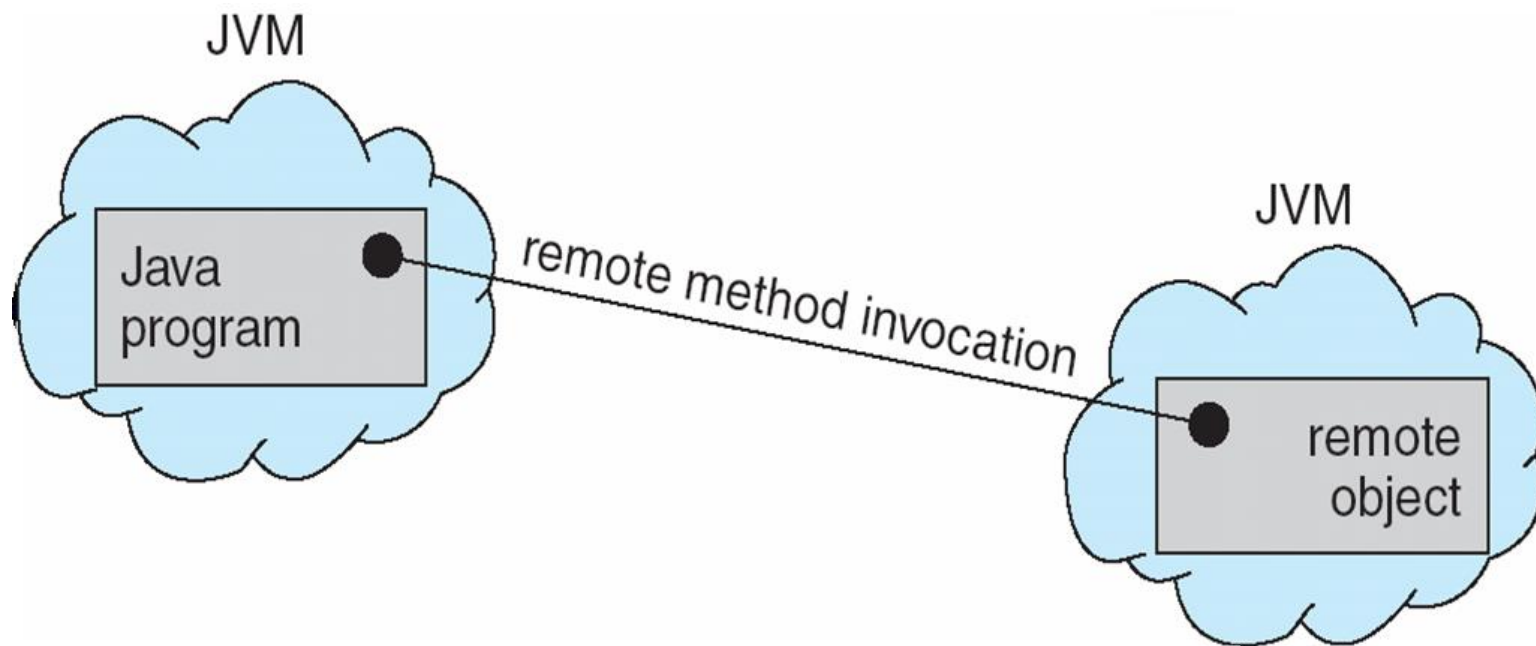
- Remote Method Invocation (RMI) is a Java mechanism similar to RPCs
- RMI allows a Java program on one machine to invoke a method on a remote object





# استدعاء الطريقة البعيدة

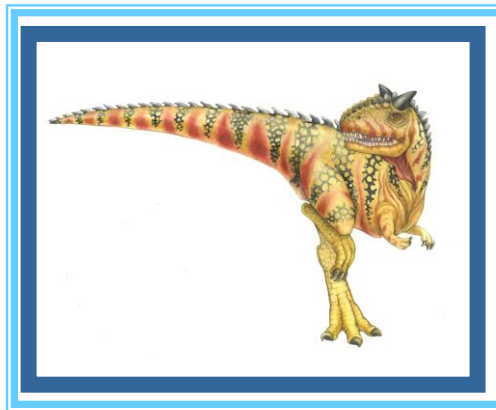
- استدعاء الطريقة عن بعد ( RMI ) هي آلية Java مشابهة لـ RPCs
- يسمح RMI لبرنامج Java على جهاز واحد باستدعاء طريقة على كائن بعيد





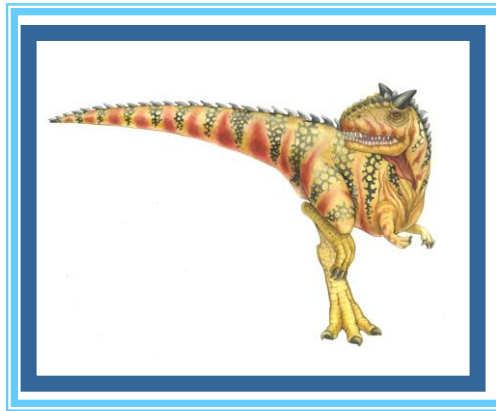
# End of Chapter 3

---

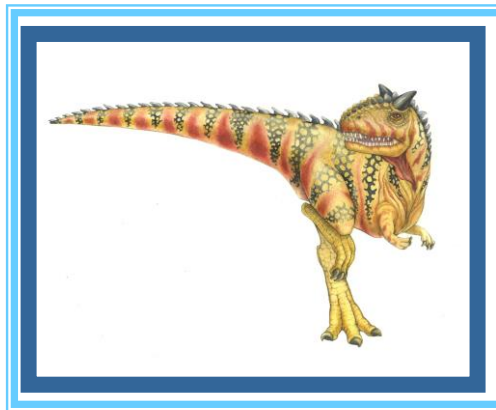


# Chapter 5: CPU Scheduling

---



# الفصل 5: جدولة وحدة المعالجة المركزية نظم التشغيل





# Chapter 5: CPU Scheduling

---

- Basic Concepts
- Scheduling Criteria
- Scheduling Algorithms
- Thread Scheduling
- Multiple-Processor Scheduling
- Operating Systems Examples
- Algorithm Evaluation





## الفصل 5: جدولة وحدة المعالجة المركزية

---

- مفاهيم أساسية
- معايير الجدولة
- جدولة الخوارزميات
- جدولة الموضوع
- جدولة متعددة المعالجات
- أمثلة على أنظمة التشغيل
- تقييم الخوارزمية





# Objectives

---

- To introduce CPU scheduling, which is the basis for multiprogrammed operating systems
- To describe various CPU-scheduling algorithms
- To discuss evaluation criteria for selecting a CPU-scheduling algorithm for a particular system

## أهداف:

- لتقديم جدولة وحدة المعالجة المركزية ، والتي هي أساس أنظمة التشغيل متعددة البرامج.
- لوصف مختلف خوارزميات جدولة وحدة المعالجة المركزية.
- لمناقشة معايير التقييم لاختيار خوارزمية جدولة وحدة المعالجة المركزية لنظام معين.





# Basic Concepts

- Maximum CPU utilization obtained with multiprogramming
- CPU–I/O Burst Cycle – Process execution consists of a *cycle* of CPU execution and I/O wait
- **CPU burst** distribution

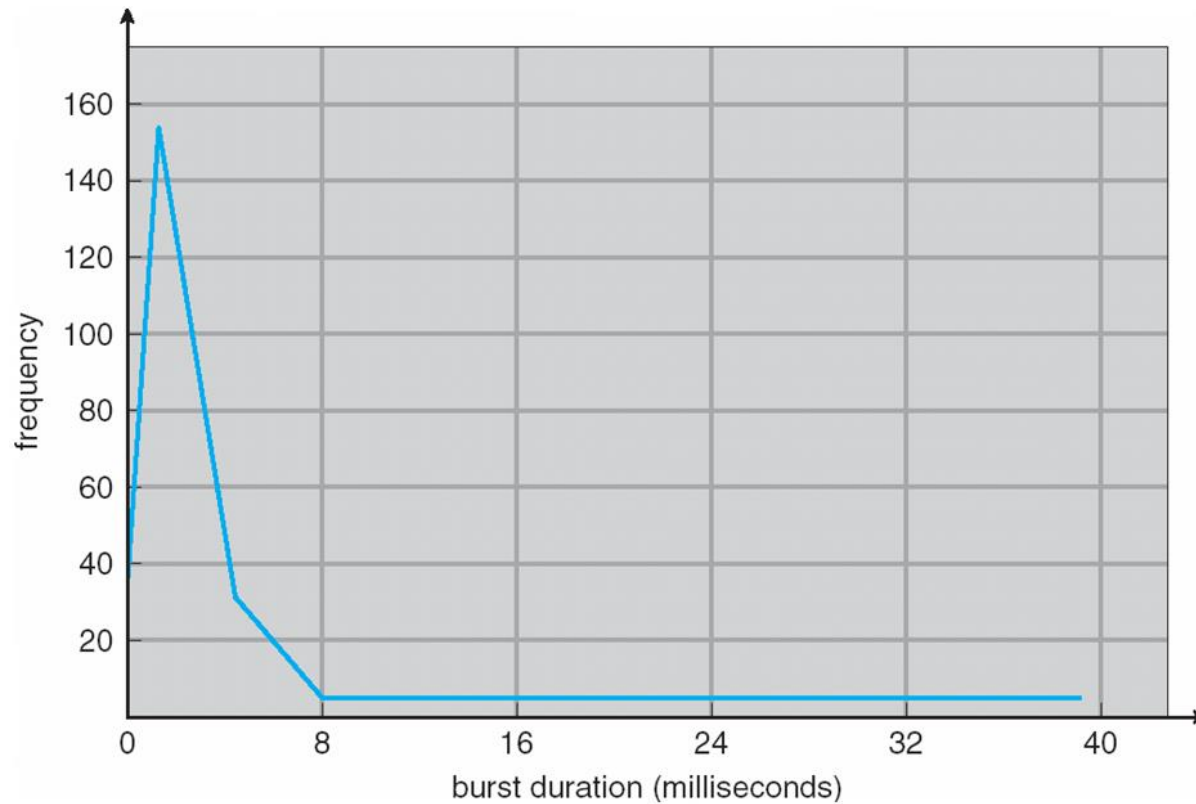
## مفاهيم أساسية

- الاستفادة القصوى من وحدة المعالجة المركزية التي تم الحصول عليها من خلال البرمجة المتعددة.
- وحدة المعالجة المركزية - دورة انفجار الإدخال / الإخراج - يتكون تنفيذ العملية من دورة تنفيذ وحدة المعالجة المركزية وانتظار الإدخال / الإخراج.
- توزيع انفجار وحدة المعالجة المركزية





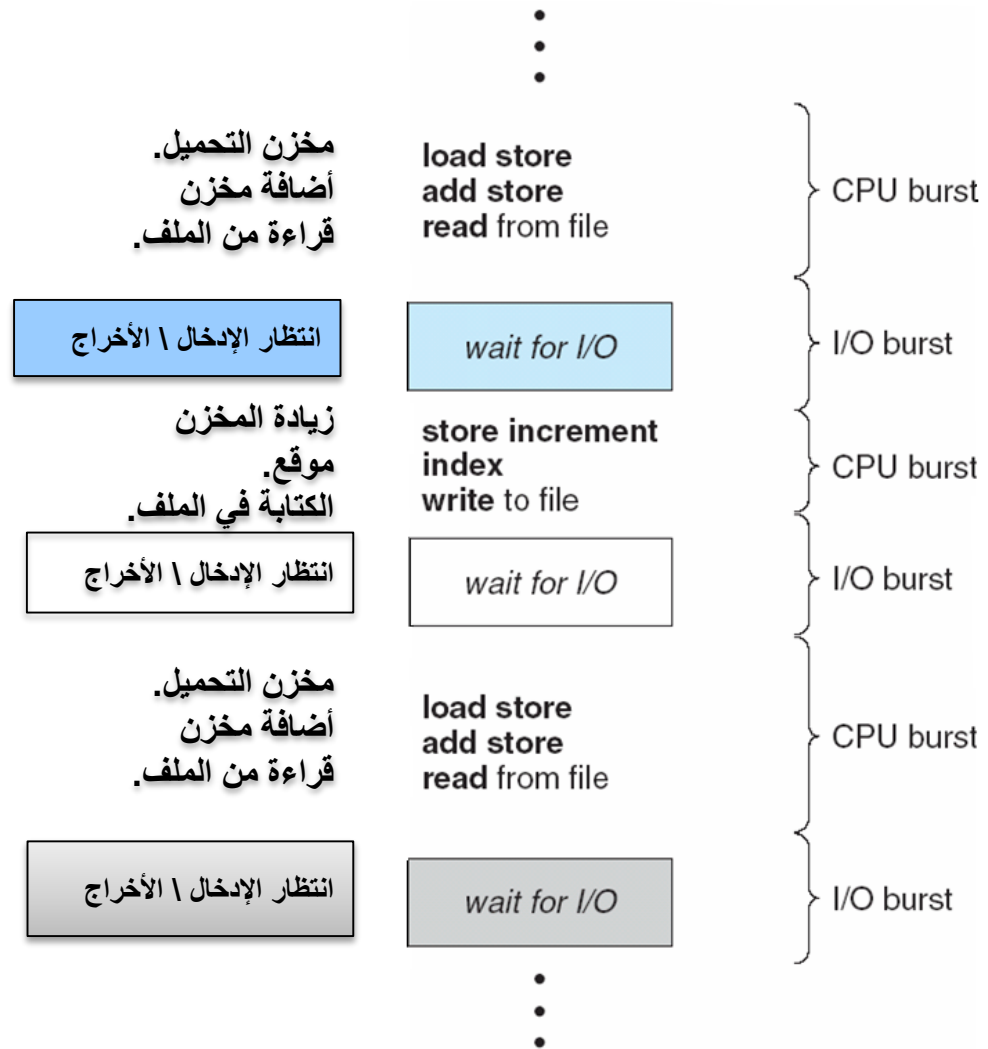
# Histogram of CPU-burst Times







# Alternating Sequence of CPU And I/O Bursts





# CPU Scheduler

---

- Selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them
- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates
- Scheduling under 1 and 4 is **nonpreemptive**
- All other scheduling is **preemptive**





# جدولة وحدة المعالجة المركزية

- يختار من بين العمليات الموجودة في الذاكرة الجاهزة للتنفيذ ، ويخصص وحدة المعالجة المركزية لواحدة منها
- قد يتم اتخاذ قرارات جدولة وحدة المعالجة المركزية عندما تكون العملية:
  1. التبديل من حالة التشغيل إلى حالة الانتظار
  2. التبديل من حالة التشغيل إلى حالة الاستعداد
  3. التبديل من الانتظار إلى الاستعداد
  4. ينتهي.
- الجدولة تحت 1 و 4 غير استباقية
- جميع المواعيد الأخرى استباقية





# Dispatcher

---

- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves:
  - switching context
  - switching to user mode
  - jumping to the proper location in the user program to restart that program
- **Dispatch latency** – time it takes for the dispatcher to stop one process and start another running





- وحدة المرسل تعطي التحكم في وحدة المعالجة المركزية للعملية المحددة من قبل المجدول على المدى القصير ؛ وهذا ينطوي:
  - \* تبديل السياق.
  - \* التحول إلى وضع المستخدم.
  - \* القفز إلى المكان المناسب في برنامج المستخدم لإعادة تشغيل هذا البرنامج
- زمن انتقال الإرسال - الوقت الذي يستغرقه المرسل لإيقاف إحدى العمليات وبدء تشغيل أخرى.





# Scheduling Criteria

---

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – # of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process
- **Waiting time** – amount of time a process has been waiting in the ready queue
- **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)





# معايير الجدولة

- استخدام وحدة المعالجة المركزية - اجعل وحدة المعالجة المركزية مشغولة قدر الإمكان
- الإنتاجية - عدد العمليات التي تكمل تنفيذها لكل وحدة زمنية
- وقت الاستجابة - مقدار الوقت اللازم لتنفيذ عملية معينة.
- وقت الانتظار - مقدار الوقت الذي كانت تنتظره العملية في قائمة الانتظار الجاهزة
- وقت الاستجابة - مقدار الوقت المستغرق من وقت تقديم الطلب حتى إنتاج الاستجابة الأولى ، وليس الناتج (لبيئة مشاركة الوقت)





# Scheduling Algorithm Optimization Criteria

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

## جدولة معايير تحسين الخوارزمية

- أقصى استخدام لوحدة المعالجة المركزية
- أقصى سرعة
- الحد الأدنى من الوقت المستغرق
- وقت الانتظار دقيقة
- وقت الاستجابة الأدنى







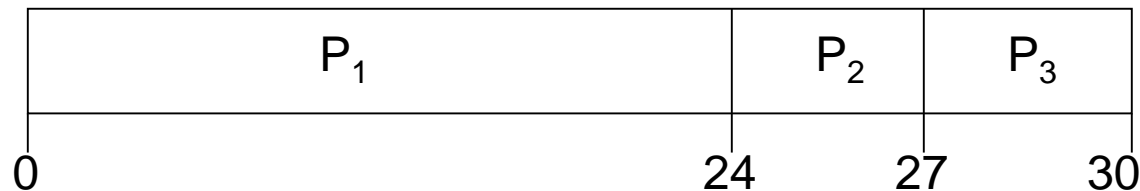
# First-Come, First-Served (FCFS) Scheduling

## جدولة من يأتي أولاً ، يخدم أولاً (FCFS)

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:

افترض أن العمليات تصل بالترتيب:  $P_1$  ,  $P_2$  ,  $P_3$  مخطط جانث للجدول هو:



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



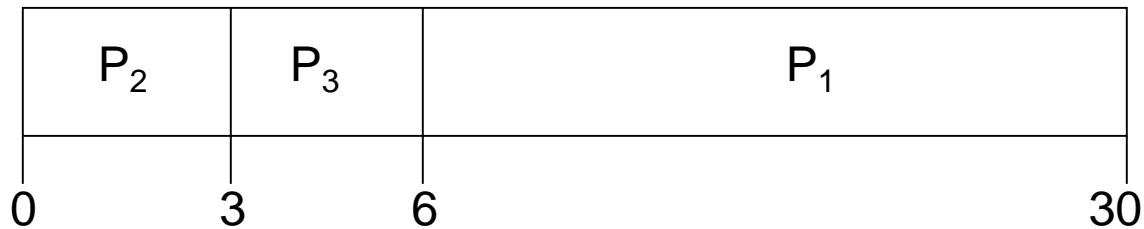


# FCFS Scheduling (Cont)

Suppose that the processes arrive in the order

$$P_2, P_3, P_1$$

- 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





# Shortest-Job-First (SJF) Scheduling

- Associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time
- SJF is optimal – gives minimum average waiting time for a given set of processes
  - The difficulty is knowing the length of the next CPU request

جدولة أقصر مهمة أولاً ( SJF )

- ارتبط مع كل عملية بطول انفجار وحدة المعالجة المركزية التالي. استخدم هذه الأطوال لجدولة العملية بأقصر وقت
- SJF هو الأمثل - يعطي الحد الأدنى لمتوسط وقت الانتظار لمجموعة معينة من العمليات  
\* تكمن الصعوبة في معرفة طول طلب وحدة المعالجة المركزية التالي.

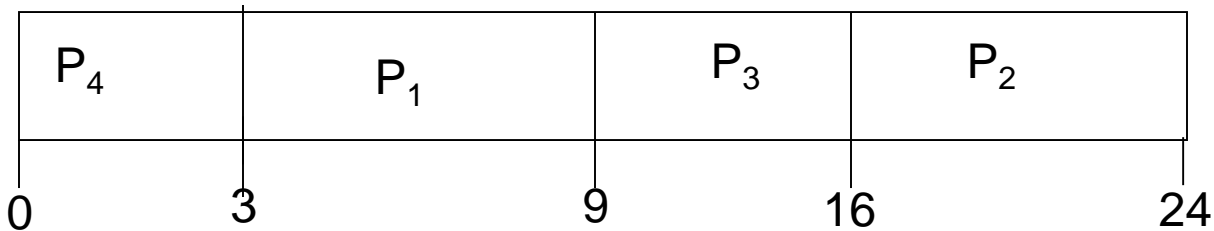




# Example of SJF

<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
$P_1$	0.0	6
$P_2$	2.0	8
$P_3$	4.0	7
$P_4$	5.0	3

- SJF scheduling chart



- Average waiting time =  $(3 + 16 + 9 + 0) / 4 = 7$

