



حركة الشبيبة الفتاوية

مادة كلية الهندسة وتكنولوجيا المعلومات

تلخيص الطالب

**Ahmed Elnajjar**

حركة الشبيبة الفتاوية  
جامعة الأزهر

   | shabiba.azhar

## Class 1

### Operating Systems 8. - أنظمة التشغيل

#### Definition of an Operating System تعريف النظام

- An OS is software that acts as an interface between users/applications and hardware.

- It provides an environment to run programs and manages all hardware resources.

- مبدأ فكرة تعريفه هو إنشاء بيئة بين المستخدم والـ OS أو التطبيقات وبينه، وإدارة جميع الموارد لتشغيل البرامج وجميع الموارد المتاحة.

#### Main OS Responsibilities

- مهام نظام التشغيل الرئيسية

- \* CPU management (scheduling) <sup>Processes</sup>
- \* File system management
- \* Concurrency and synchronization
- \* protection and security
- \* Memory management
- \* Device and I/O management
- \* Resource allocation among ~~multiple~~ multiple programs
- \* Providing system calls and OS services



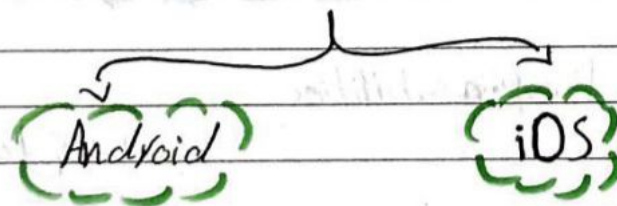
## Course Topics

المادة الأساسية

- Introduction to OS: batch systems, multiprogramming, time sharing
- System Structure: I/O, storage hierarchy, hardware protection
- Processes: concept, scheduling, operations, IPC.
- Synchronization: critical section, ~~semaphores~~ semaphores, monitors.
- CPU scheduling algorithms.
- Deadlocks: prevention, avoidance, detection, recovery.
- Memory and virtual memory management.

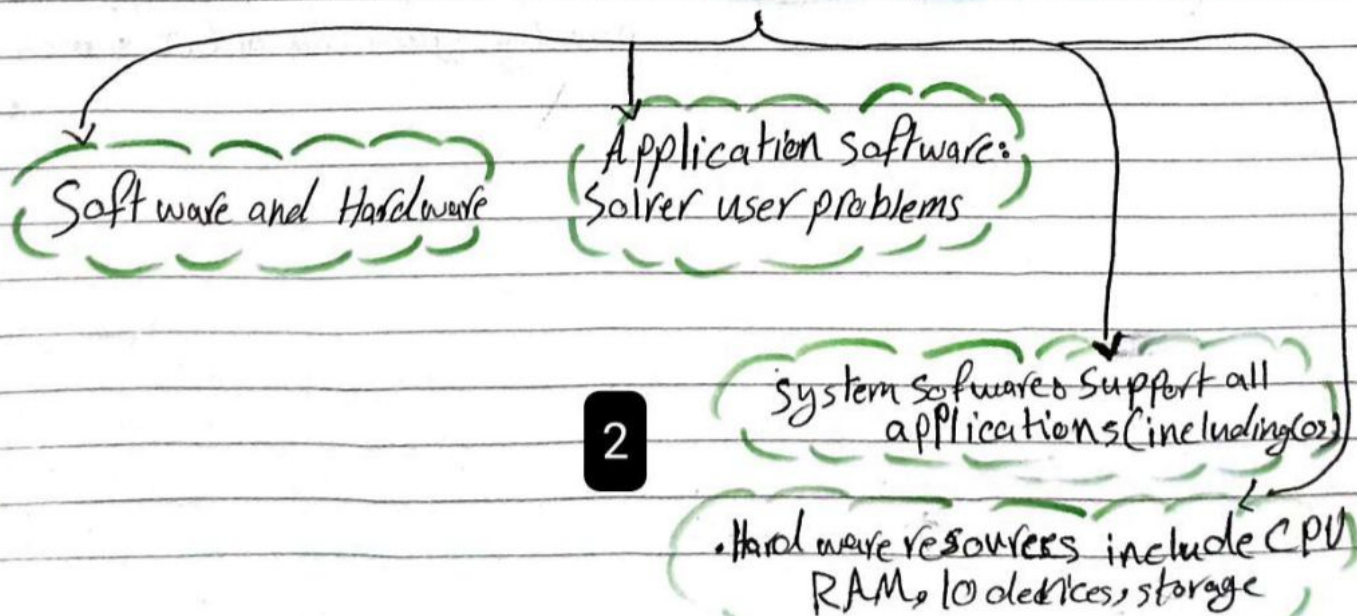
## Mobile OS Examples

أمثلة لنظم التشغيل الجوال



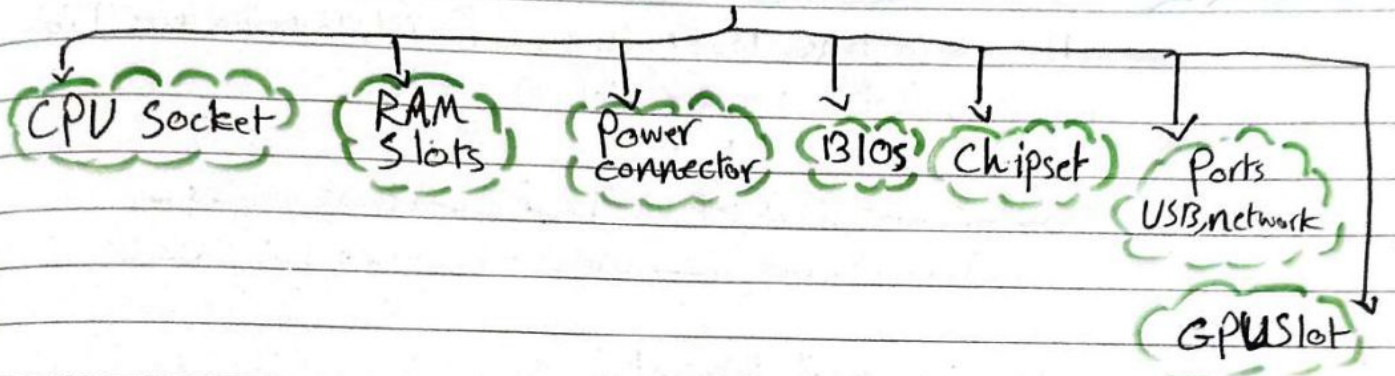
مكونات نظام الحاسوب

## Computer System components



## Motherboard Components

مكونات اللوحة الأم



Q 11 OS as a mediator؟ كيف يعمل ال OS كوسيط؟

S: Applications use system calls instead of talking directly to hardware so this ensure protection simplification, and resource sharing.

Class 1

*Ammed*



## Calss 2

### Early 1940s Military compute

- Huge & 30 tons, 100ft long, 8ft tall, 18,000 vacuum tubes, very hot, needed lots of cooling.

بمعنى أنه الحاسوب قديماً كان ضخماً جداً، حار، يحتاج إلى تبريد مكثف،  
عنايه كذا كانت كبيرة استهلاكه في الجيش والفضاء.

### Program loading and Interactio

- Programs loaded manually using switches or punched card  
no ~~inter~~ interactive OS humans set up and run ~~jobs~~ job  
step by step.

يعني البرامج كانت تدخل يدوياً، ومالاً في نظام تشغيل تقاسم، لا شيء  
كأنه بيئة التشغيل البشري خطوة بخطوة.

### Computer/OS Generation

- First Gen (1940s) & vacuum tubes, batch processing.
- Second Gen (1950s) & transistors, smaller, faster, less heat
- Third Gen & integrated circuits (ICs), more advanced OS features
- Fourth Gen & microprocessors, PCs, miniaturization

كما نرى أنه مع تطور الأجيال كانت الميزات تزداد كالجم، ففقد الحرارة  
زمن، الموثوقية، وتطور أيضاً في نظم التشغيل، مثل تعدد  
المهام والمشاركة مع المستخدمين

## Peripherals and Hardware Evolution

تطور الأجهزة الطرفية

punch cards and add-on boards improved I/O Helped automate tasks and made OS more complex (device drivers, Spooling).

المعدات الطرفية مثل البطاقات المثقبة والوحدة الإضافية حسنت الإدخال والإخراج، ساعدت على أتمتة المهام، وأدت إلى تعقيد أنظمة التشغيل بشكل أكبر.

## Batch Vs Time-Sharing

- Batch jobs queued and executed one by one  
بعض في الباتش المهام تنتظر دورها وتنفذ واحدة واحدة

- Time sharing: multiple users interact the same time  
بعض عدة مستخدمين يتفاعلون مع الحاسوب بنفس الوقت، حيث يُشغل النظام بشكل أفضل.

## Mainframe + Terminal

- Terminals: simple input/output devices connected to mainframe.

- Mainframe: executed all tasks and returns to terminal.

الطرفيات مجرد أجهزة ادخال واخراج، كل المعالجة تتم في الكمبيوتر المركزي الكبير والسريع ترجع الطرفيات بهذا يسمح للأجهزة كثيرة بمسألة نفس الموارد.



## Multi processing & Shared vs Distributed Memory

Shared-memory & Processes share the same memory.  
يعني كل العمليات تستخدم نفس الذاكرة، لا يوجد نظام تشغيل، لكن جميع البرامج.

Distributed-memory & Processors have private memory, communicate via network.

يعني لكل معالج ذاكرته الخاصة، التواصل عبر شبكة، أسهل للتوسع، لكن يحتاج تنسيق أكثر بين المعالجات.

## Tightly coupled vs Loosely coupled

Tightly coupled & shared memory single OS image, high-performance.  
النظام متكامل، نظام واحد، نظام واحد، نظام واحد، نظام واحد، نظام واحد.

Loosely coupled & independent processors, message passing scalable.

أكثر ارتباط المعالجات، معقدة، أسهل للتوسع، لكن تنسيق بين المعالجات أصعب.

Amr  
class 2

### Class 3

#### 1) What is an operating system?

An operating system (OS) is the main software that manages the computer hardware and provides services for applications. It acts as a bridge between users/programs and the hardware. The OS makes running programs easier by handling low-level tasks like memory management, CPU scheduling, file handling and device control.

يعني نظام التشغيل هو البرنامج الأساسي الذي يدير العتاد ويرتب العمل على البرامج بشكل  
وهو الوسيط بينك وبين الجهاز ويدير الذاكرة والمعالج والملفات والأجهزة.  
"نظام شرحه في Class 1"

#### 2. Components of a computer system

A computer is made of four main components:

1. Users
2. Applications (browsers, games, editors, etc..)
3. Operating System
4. Hardware (CPU, memory, disk, I/O devices)

The OS connects the applications to the hardware safely and efficiently.

الجهاز يتكون من المستخدم، البرامج، نظام التشغيل، العتاد.  
والنظام هو الذي يوصل البرامج مع العتاد ويرتب العمل.



## OS as a Resource Manager:-

- The OS controls how resource are shared among programs. It decides

- Which program gets CPU time

- how memory is allocated

- how I/O devices are used.

- This ensures efficiency and fairness when multiple programs run at the same time.

• النظام يوزع المعالج، الذاكرة، وأجهزة الإدخال والإخراج بين البرامج بشكل عادل وفعال.  
ويجوز النظام أيضاً أن يسيطر على وقت وحدة التخزين المركزي، كيفية تخصيص الذاكرة  
كيفية استخدام أجهزة الإدخال والإخراج.

## ~~OS as a control program~~ OS as a control program

- The OS Control program execution to prevent errors and illegal actions. It protects the system from faults and keeps everything running smoothly. It is the only program that must always be running.

• لا يمكن للبرامج أن تتداخل مع عمل البرنامج المسيطر، ويجب أن يكون البرنامج المسيطر دائماً قيد التشغيل.  
يجب أن يكون البرنامج المسيطر دائماً قيد التشغيل.

- النظام يراقب البرامج ويمنع الأخطاء، وهو الوحيد الذي يجب أن يكون قيد التشغيل دائماً.

## Device Controllers and I/O Buffering

- Each hardware device is managed by a controller, which has a small buffer to hold data during transfer. The CPU communicates with these controllers to move data between main memory and devices.

كل جهاز له كترولر، وأفرقته للتحريك للوقت وبذلك فالجهاز يقوم بنقل البيانات بين الذاكرة والكترولر.

## Interrupts

Interrupts allow devices to notify the CPU when an operation finishes. This avoids wasting CPU time in waiting. When an interrupter occurs, the CPU runs a specific routine called an ISR. The addresses of ISRs are stored in the interrupt vector table.

9

الفرق هو، ان كل من الجهاز بجدة انه ابصر للجهاز لما يتصله شغل الجهاز يعود على ISR او في جدول مخزنه فيه عناوينه والروتينات.

### Interrupt Service Routines (ISR)

هو قطعة كود يستخدمها الجهاز لاجابة Interrupt. فالجهاز حينه كذا (كيورد كود) يرسل interrupt للجهاز. هذا interrupt هو ISR. ISR مسؤول عن معالجة سبب Interrupt، قراءة البيانات او التعامل مع الخطأ، تجهيز النظام لرجوع البرنامج الى مكانه السابق.

met

ب class



## Handling Interrupts

مبدأ التعامل مع المقاطعات

When an interrupt happens?

1. OS saves the CPU state "registers, program counter"
2. CPU runs the ISR
3. OS restores the saved state.
4. program continues normally

مبدأ الخطة البرنامج بحفظ حاله الحالي ← يستدعى ISR ← يرفع الكاله ← البرنامج لم يكمل.

## Traps (Software Interrupts)

- A trap is a software-generated interrupt caused by an error (like divide by zero), or a system call made by a program

التراب هو انترابت من البرامج بسبب خطأ زوطلب خدمه النظام

مثال على خطأ داخل البرنامج و قسمه على صفر، فلها يكونه طريقه (system call) والى هو استدعاء النظام

يتم الخليه لخدمه انواع المعالج يوقف تنفيذ البرنامج وينقل التحكم لنظام التشغيل بعد ما تكامل التشغيل يستدعى Trap Handler المناسب ونفسه ما يتكامل البرنامج او يستمر اذا لزمه خطأ كسب.

## Polling vs Vectored Interrupts

- Polling: CPU checks devices one by one - slower
- Vectored Interrupts: hardware gives the exact ISR address. Faster and more efficient.

بكل بـ لـ ~~البرنامج~~ (Polling) ~~البرنامج~~ لأنه يحتاج يفحص الأجهزة

أما (Vectored) أسرع لأنه يبرمج مباشرة للرئيسة المطلوب  
بجهاز ينسج الوال (Vectored) أسرع بالأداء.

## I/O Operation Timeline

- Program requests an I/O operation.
- Controller transfers data to/from device.
- Device sends an interrupt when done.
- OS handles the interrupt and resumes the program.

عبر خطوات I/O ~~البرنامج~~ طلب - نقل - استجابة - معالجة - ~~البرنامج~~

الخطوات الرئيسية التي تتم في الجهاز البنية

1. البرنامج يطلب عمل إدخال / إخراج
2. المبرمج هو الذي يتعامل مع الجهاز وينقل البيانات
3. كما يولد البرنامج Interrupt
4. نظام التشغيل يتعامل مع الإشارة وينقلها للبرنامج

Amal



## Class 4

### Storage Hierarchy

Storage is organized in levels from fastest to slowest:

Registers → cache → Main → memory → Disk → Network storage

- Fast levels are small and expensive. slow levels are large and cheap.

- System move frequently-used data upward to faster levels.

بمعنى أنه إذا كانت مرتبة، البيانات في الأسرع فوه، الأنظمة تحت، أي ما قبل، تكبيره  
و ينقل السري، يعني أن النظام يحاول يخلو البيانات فوه فوه خسارة الوصول يكون أسرع.

### Caching

The CPU checks the cache first for data.

∴ IF found → Cache Hit (fast)

∴ IF not → cache Miss, data is fetched from slower memory and stored in cache for next time.

يعني الخايع فوه فوه فوه، (cache)، إذا وجد فوه فوه، إذا لم يجد  
ينقل إلى الذاكرة السري، (cache)، ويخزنه فيه.

## Data Movement

## حركة البيانات

- Data moves from slow storage to faster storage temporarily.

∴ Could reduce slow accesses by keeping hot data in fast levels.

\* النظام ينقل البيانات المطلوبة للكمبيوتر بسرعة عالية لتتمتع بالسرعة كما أنها تكون جارية

مثال: النظام يواظب على بيانات المستخدم (الهارد) وينقلها للكمبيوتر (الرام) وإذا كانت مهمة كثير للبرامج، ينقل نسختها منها للـ (cache)

النتيجة من Data movement هي أن نقل البيانات إلى المستويات القريبة من المعالج يكون أسرع.

## Symmetric Multiprocessing (SMP)

المعالجة المتماثلة

- ∴ Multiple CPUs/cores work as equals. Each core has its own registers and cache. Any core can run any task.

يعني أن المعالجات متساوية بالحقول والذاكرة الخاصة بها، النظام يوزع المهام بينهم.

## Multicore and Logical cores

- One physical CPU can contain multiple cores. Each core acts like an independent processor with its own small caches.

\* المعالج الواحد يمكنه أن يحتوي على أكثر من وحدة معالجة، وكل وحدة لها معالجها الخاص.



## Batch Systems and Multiprogramming

Batch systems process grouped jobs. Multiprogramming keeps CPU busy by switching to another job when one is waiting.

\* النظام يجمع شغل ويبديل بينهم حتى يستعمل الجهاز فارغاً.

## Single-User vs Multiprogramming

- Single user → CPU may stay idle during waits
- Multiprogramming → CPU switches to another job to avoid idle time.
- إذا استخدم واحد الجهاز يقف عندها يقف الشغل،  
لوعدة لراجع النظام فيستبدل ويبقى شغال.

## Job Scheduling

- The OS chooses which job to run and when another job should wait ensuring high CPU utilization.

النظام يقرر فيه يشغل فيه يصير ينظر على ريفل الطالب ريفل مشغول حالياً.

## Time-Sharing

• CPU time is split into very small slices so many users feel like the system works for them simultaneously.

الطالب بيوزع وقته أجزاء صغيرة على كل واحد منهم انه خلال 20 نفس الثانية

\* إمكانية توزيع الوقت على كل طالب

## Sequential Instruction Execution

Instructions run one after another in order; each completes before the next starts.

يعني إذا امرت تشغيل التسلسلي، أمر الخلية ويجب أن ينتهي قبل أن يبدأ التالي.

## Process Management

The OS manages processes create, delete, suspend, ~~and~~ resume restart, and handle user/system processes.

يعني النظام مسؤول عن العمليات داخل الجهاز، إنشاء، حذف، إيقاف، تشغيل العمليات.

## Process Sync and ~~Deadlock~~ Deadlocks

OS provides tools for:

- Synchronization
- Communication between processes (IPC)
- Deadlock handling

15

يعني النظام يوفر أدوات للعمليات للتواصل والتزامن.

(Deadlock)

## Memory Management

OS ~~tracks~~ tracks memory usage, decides what stays in memory, swaps data when needed, and allocates/free memory to processes.

النظام يتتبع الذاكرة، يقرر متى ينقل البيانات أو يوزع الذاكرة بين البرامج.

Amr

class 4



# Full Comparison / Key OS Concepts

Topic	Explanation
<ul style="list-style-type: none"> <li>Single-User vs Multiprogramming</li> </ul>	<ul style="list-style-type: none"> <li>Single-User: Only one program runs, CPU may be idle if program waits.</li> <li>Multiprogramming: Multi programs in memory, CPU switches tasks to stay busy.</li> </ul>
<ul style="list-style-type: none"> <li>Batch vs Time <del>Sharing</del> Sharing</li> </ul>	<ul style="list-style-type: none"> <li>Batch: Processes jobs in groups; no direct user interaction.</li> <li>Time-sharing: CPU time is split among users, users appear to work simultaneously.</li> </ul>
<ul style="list-style-type: none"> <li>Multicore vs Multi-CPU</li> </ul>	<ul style="list-style-type: none"> <li>Multicore: One physical CPU with multiple cores that run in parallel.</li> <li>Multi-CPU: Multiple separate CPUs in the system, each with its own cache and registers.</li> </ul>
<ul style="list-style-type: none"> <li>SMP vs Multicore</li> </ul>	<ul style="list-style-type: none"> <li>SMP: Multiple processors work as equals, any CPU can run any task.</li> <li>Multicore: Multiple cores inside one CPU, cores share some resources but act as independent execution units.</li> </ul>
<ul style="list-style-type: none"> <li>Polling vs Vectored interrupts</li> </ul>	<ul style="list-style-type: none"> <li>Polling: CPU checks devices one by one slowly.</li> <li>Vectored: Device gives ISR address directly → faster and more efficient.</li> </ul>

← تابع

~~1/1~~

<ul style="list-style-type: none"> <li>• <del>Cache</del> Cache Hit Vs Cache miss</li> </ul>	<ul style="list-style-type: none"> <li>• Hit: Data is in cache → Fast access.</li> <li>• Miss: Data not in cache → Fetched from slower memory slower access.</li> </ul>
<ul style="list-style-type: none"> <li>• Registers Vs Cache Vs Main Memory Vs Disk Vs Network</li> </ul>	<ul style="list-style-type: none"> <li>• Registers: Fastest, smallest, most expensive.</li> <li>• Cache: Fast buffer close to CPU.</li> <li>• Main memory: Slower, larger.</li> <li>• Disk: slower, larger, cheap.</li> <li>• Network: Slowest, very large.</li> </ul>
<ul style="list-style-type: none"> <li>• SMP Vs Single processors</li> </ul>	<ul style="list-style-type: none"> <li>• SMP: multiple processors share tasks → parallel execution.</li> <li>• Single processor: only one CPU executes tasks sequentially → slower for multiple tasks.</li> </ul>
<ul style="list-style-type: none"> <li>• I/O Handling: Interrupt Vs Polling</li> </ul>	<ul style="list-style-type: none"> <li>• Interrupt: Device notifies CPU when ready → efficient.</li> <li>• Polling: CPU repeatedly checks device → wastes CPU time.</li> </ul>
<ul style="list-style-type: none"> <li>• System call Vs Trap</li> </ul>	<ul style="list-style-type: none"> <li>• Call: way for program to request one service → trap triggers handling.</li> <li>• Trap: Software-generated interrupts.</li> </ul>
<ul style="list-style-type: none"> <li>• Time-slicing / Time sharing</li> </ul>	<p>CPU time is divided into slices so multiple users/tasks can interact concurrently, giving the appearance of simultaneous execution.</p>

*Amir*