



حركة الشبيبة الفتحاوية

جامعة كلية الهندسة وتكنولوجيا المعلوما

تلخيص الطالب

Ahmed Elnajjar

حركة الشبيبة الفتحاوية  
جامعة الأزهر



| shabiba.azhar

## Class 1

### Operating Systems جیئنگ نرم افزاری

#### Definition of an Operating System انواع سیستم عامل

- 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.

OS مخفیت، نرم افزاری است که روابط بین کاربر و سخت افزار را مدیریت می کند و از کاربر دهنده های بسیاری برای اجرا کردن برنامه ها و اداره موارد رجوع می کند.

#### Main OS Responsibilities مهام اصلی سیستم عامل

- \* CPU management (Scheduling processes)
- \* Memory management
- \* File System management
- \* Device and I/O management
- \* Concurrency and synchronization
- \* Resource allocation among processes
- \* Protection and security
- \* Multiple programs
- \* Providing system calls and OS services

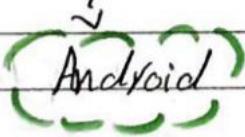
## Course Topics

Notes, Examples -

- \* . Introduction to OSs 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.

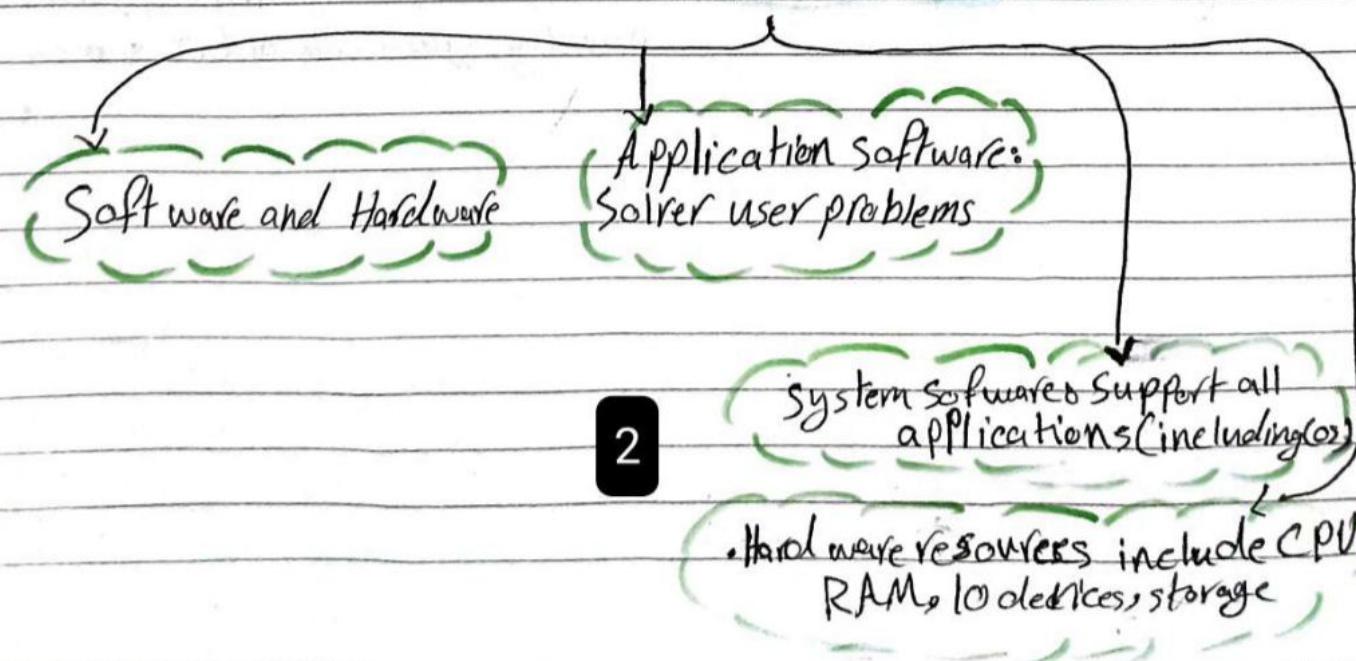
## Mobile OS Examples

Android, iOS



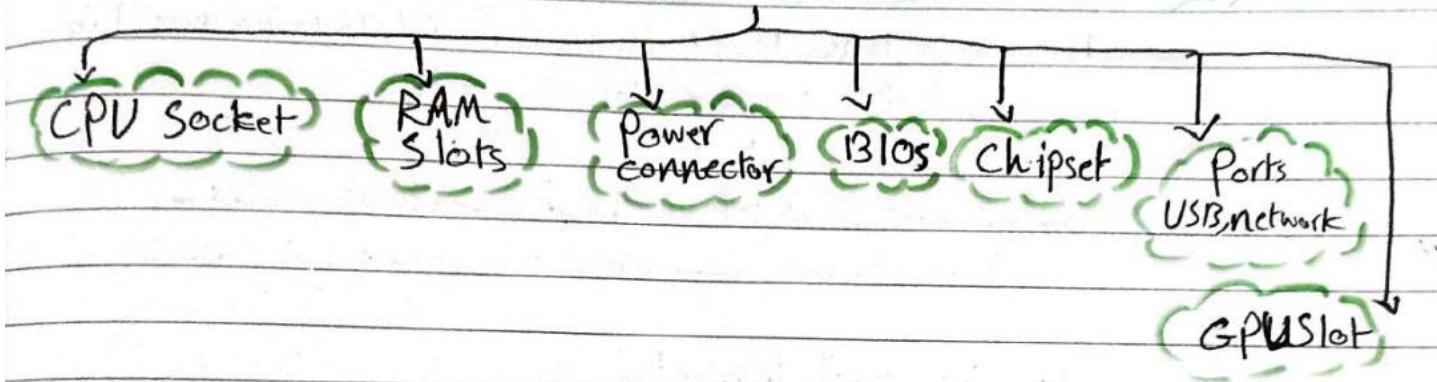
البيانات المخزنة

## Computer System Components



## Motherboard Components

مكونات اللوحة.



Q 11 OS as a mediator ؟ بس OS جي لازم

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

Class 1

frame p

## Class 2

### Early 1940s Military computer

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

جهاز حاسوب قديم جداً في الأصل، ضخم جداً، 30 طن، 100 قدم، 8 قدم، 18,000 مثبّت ضغط الهواء، بحاجة إلى تبريد كثيف.

### Program Loading and Interaction

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

يُعد البرامج في الأصل تحميل يدوياً، على ما في نظام تحكم تلقائي، لكنه في الواقع ليس كذلك.

### 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.

三代ي أو مع تطور الأدوات كانت الميزات تزداد كثافة، لكنها مكلفة، كبيرة الحجم، وتطوري أسرع في تطوير الميكروChip، مثل هذه الميزات.

عام 1940م وللأول مرة في العالم.

## Peripherals and Hardware Evolution

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

البطاقات الخفيفة والجهاز المترافق قلل من الالآلات المترافق، وساعدت على إدخال وإخراج الملفات، وأدت إلى تغيير أنماط العمل بشكل أكبر.

## Batch Vs Time-Sharing

- Batch jobs queued and executed one by one.

- Time sharing: multiple users interact the same time.

## Mainframe + Terminal

- Terminals simple input/output devices connected to mainframe.
- Mainframe executes all tasks and returns to terminal.

الهاتفيات مجرد أجهزة إدخال وإخراج، كل معاينته تم من الميمون المركزي الكبير والمتابع. ترجع الهاتفيات هنا بمعنى آخر أنه ليس في مساحة نفس الموارد.

## ● Multiprocessing & Shared vs Distributed Memory

- Shared-memory & processes share the same memory.  
جهاز واحد يحتوي على ذاكرة واحدة يشار إليها من قبل كل جهاز
- Distributed-memory & processors have private memory, communicate via network.  
جهاز كل جهاز يحتوي على ذاكرة خاصة به يشار إليها من قبل كل جهاز

## ● Tightly coupled vs Loosely coupled

- Tightly coupled & shared memory single OS image, high-performance.  
الاتصال قوي بين الأجهزة، ولنظام واحد فقط
- Loosely coupled ~~not~~ independent processors, message passing scalable.  
الاتصال ضعيف بين الأجهزة، ولأجهزة مختلفة، ولنظام متعدد

Final

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.

النظام ~~الآنجل~~ هو البرنامج الذي يدير الماردور ويجهز لك كل الأدوات  
لقد أتى بهم الجهاز ويسهل على المستخدم والباحث والملحقات الدخولة.

"Class 1" نظام التشغيل

### 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~~ the hardware safely and efficiently.

نظام التشغيل هو البرنامج الذي يدير الماردور  
ويجعله متاحاً لجميع الأدوات والملحقات والخدمات.

## OS as a Resource Allo Manager :-

- The OS controls how resources 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 الموارد، والرام، وأجهزة الادخال والادخال بحسب احتياجات كل مسحوق. وتحتاج كل مسحوق إلى مساحة معينة، كي لا تتعارض احتياجاته مع احتياجات الماسحوق الآخر.

## OS as a control program

- The OS controls 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.

نظام يراقب البرامج ويمنع اخطاء، ويتحقق من ادخال خالٍ من اخطاء.

## 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.

الجهاز يرسل إشارة إلى المعمulus لبيان انتهاء العملية  
ويتم إيقاف المعمulus وإلزامه بالانتظار

## Interrupts

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

9

الجهاز يرسل إشارة إلى المعمulus لبيان انتهاء العملية  
وهي جدول غير ثابت في الذاكرة يحتوي على إشارات إلى المركبات.

## Interrupt Service Routines ISR

The interrupt routine (ISR) is a function that handles the interrupt. It is triggered when an interrupt occurs and performs the necessary actions to handle the interrupt. The interrupt can also be referred to as an interrupt handler or interrupt service routine. Interrupt handlers are typically implemented in assembly language, as they need to be very efficient and fast.

Interrupt

class B

## Handling Interrupts

الإنترفيكت، إجراءات معالجة

- 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

الإنترفيكت يمتحن حالات البرامج  $\rightarrow$  إنترفيكت  $\rightarrow$  إنترفيكت  $\rightarrow$  إنترفيكت

## Traps (Software Interrupts)

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

الإنترفيكت هو انترفيكت البرمجي المبرمج في خطاً في الكامputer

(SystemCall) يفتح دخواً داخل البرنامج وقسمه إلى دخواً آخر، فلها يكون دخواً في المبرمج،  
والتي هي استدعاء لخطأ، لخطأ

يتم إثارة خطأ المبرمج، وهذا يتحقق بوقف تنفيذ البرنامج، ونقل التحكم لخطأ المبرمج  
بعد ما تكلم الخطأ يرسل ~~Trap Handler~~ لخطأ المبرمج،  
وينفذ ما يطلب البرنامج أو يستمر إذا لم يطلب.

## Polling vs Vectored Interrupts

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

الرجوع إلى المراجع (Polling) ~~مُنجز~~ ملخص

أسرع دلائل على معاشرة المبرمجين (Vectored) لـ  
أسرع أنوال (Vectored) لـ

## 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.

الخطوات المتبعة في إدخال/إخراج I/O

الخطوات المتبعة في إدخال/إخراج I/O

(1) إدخال/إخراج البيانات مع الاتصال

(2) إدخال/إخراج البيانات مع الإغلاق وانتهاء الاتصال

(3) إدخال/إخراج البيانات مع الإغلاق

(4) نظام التحكم يتعامل مع الاتصال ويفعل كل ما يليه

## 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 moves frequently-used data upward to faster levels.

بعض الذاكرة مترتبة في سلسلة، كل مترتبة على الأخرى، بحيث يحصل على البيانات بسرعة، وهذه هي المفهوم الذي يدعى التسلسل.

### 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.

لذا في كل إدخال، إذا وجد في (cache) أو لم يجد في (cache) في نسيخة، ~~فهي~~ لاتجده في (cache) في المرة الأولى.

## Data Movement

حركة البيانات

- Data moves from slow storage to faster storage temporarily.
- Goal & reduce slow accesses by keeping hot data in fast levels.

\* النظام ينقل البيانات المطلوبة من المراحل المتأخرة إلى المراحل المبكرة

مثال: النظام يأخذ بيانات من عالم الواقع (الهاردوير) ويسخنها في الذاكرة فإذا أتت مandaة لـ (رام) فما زلت مخزنها في الـ RAM

النتيجة هي ازاحة البيانات، ثم نستخرجها لـ RAM

## Symmetric Multiprocessing (SMP)

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

## Multicore and Logical cores

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

\* المفهوم هو أن المعمارية تجعل كل وحدة متعددة النواة كوحدة واحدة

## 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 ويسهل بينهم ~~فقط من~~ & ينزل المعاين خارجاً.

## Single-User vs Multiprogramming

- Single user  $\rightarrow$  CPU may stay idle during waits
- Multiprogramming  $\rightarrow$  CPU switches to another job to avoid idle time.

• اذا صرخ واحد في الكواكب ينفع عندما ينفع العسل ،  
لوعة برافع ، النظام يستبدل وينفع شغال .

## Job Scheduling

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

• نظام يقرر من يدخل ويخرج من المكان في نفس المكان

## Time-Sharing

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

• المكان يوزع الى وقت كل كثيل

## Sequential Instruction Execution

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

• الاتصالات تتم بترتيبها، حيث تنتهي كل امر قبل ان يبدأ اخر

## Process Management

- The OS manages processes (creat, delete, suspend, ~~resume~~, restart, and handle user / system processes).

• المديرية تتحكم في الموارد المتاحة في الجهاز (الذاكرة، القراءة، الكتابة، إيقاف، تشغيل)

## Process Sync and ~~lock~~ Deadlocks

• OS provides tools for:

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

15

• مفهوم الميتاقيمة (Deadlock)

## Memory Management

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

• النظام يراقب الذاكرة، ويغير فيها وفقاً لاحتياجات البرنامج

• 20.11.2021

Ramadan

Class 4

# Full Comparison / Key OS Concepts

| Topic                                      | Explanation  |
|--|--|
| • Single-User vs Multiprogramming          | • Single-User: Only one program runs CPU may be idle if program waits.<br>• Multiprogramming: Multiple programs in memory, CPU switches tasks to stay busy.                            |
| • Batch vs Time <del>sharing</del> Sharing | • Batch: Processes jobs in groups; no direct user interaction.<br>• Time-sharing: CPU time is split among users; users appear to work simultaneously.                                  |
| • Multicore vs Multi-CPU                   | • Multicore: One physical CPU with multiple cores that run in parallel.<br>• Multi-CPU: Multiple separate CPUs in the system, each with its own cache and registers.                   |
| • SMP vs Multicore                         | • SMP: Multiple processors work as equals, any CPU can run any task.<br>• Multicore: Multiple cores inside one CPU, cores share some resources but act as independent execution units. |
| • Polling Vs Vectored interrupts           | • Polling: CPU checks devices one by one sequentially.<br>• Vectored: Device gives ISA address directly → faster and more efficient.   |

2.5

9/11

- Cache Hit vs Cache miss
  - Hit: Data is in cache → fast access.
  - Miss: Data not in cache → fetched from slower memory → slower access.
- Registers vs Cache vs Main Memory vs Disk vs Network
  - Registers: Fastest, smallest, most expensive.
  - Cache: Fast buffer close to CPU.
  - Main memory: Slower, larger.
  - Disk: Slower, larger, cheap.
  - Network: Slowest, very large.
- SMP vs Single processors
  - SMP: multiple processors share tasks → parallel execution.
  - Single processor: only one CPU executes tasks sequentially → slower for multiple tasks.
- I/O Handling:
  - Interrupt: Device notifies CPU when ready → efficient.
  - Polling: CPU repeatedly checks device → wastes CPU time.
- System call vs Trap
  - System call: calls way for program to request one service → trap triggers handling.
  - Trap: Software-generated interrupt.
- Time-Slicing / Time Sharing
  - CPU time is divided into slices so multiple users/tasks can interact concurrently, giving the appearance of simultaneous execution.