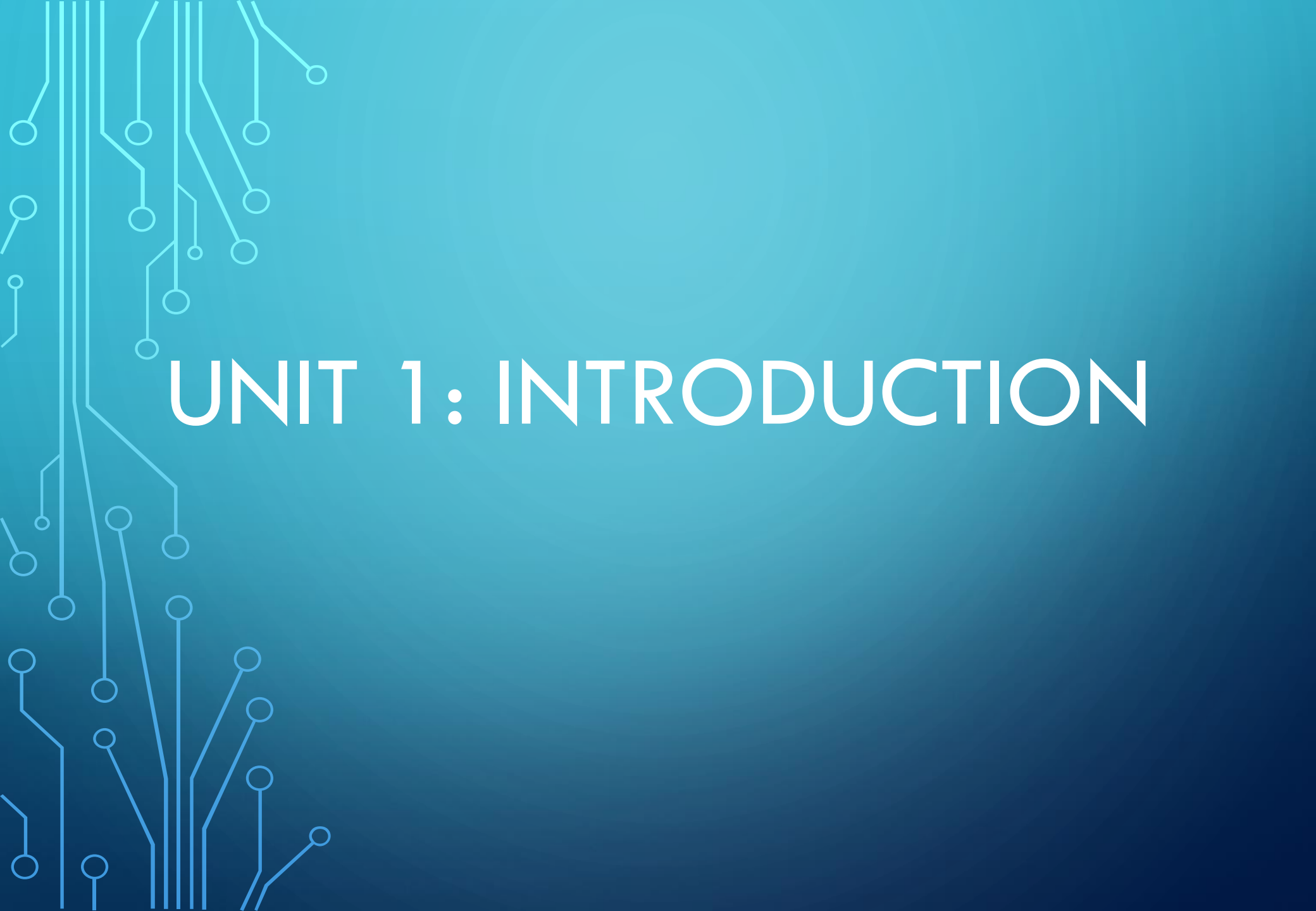
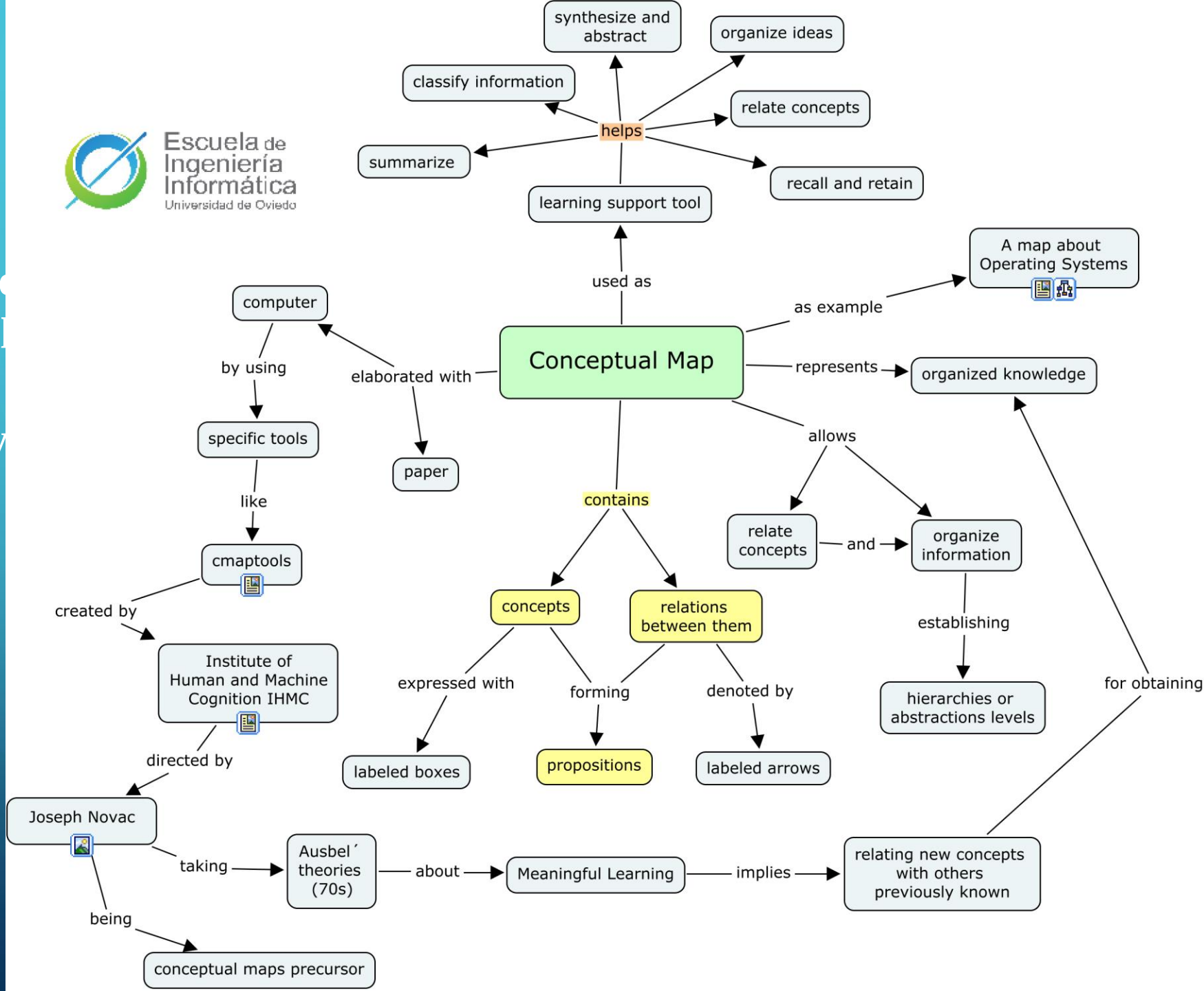

UNIT 1: INTRODUCTION

CONTENTS

We will use conceptual maps frequently



CONTENTS

0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

1. OPERATING SYSTEM CONCEPT AND FUNCTIONS

What is an Operating System?

- *What does it do?*
- *What is it for?*
- *What does it provide?*

Could we use the computer without it?

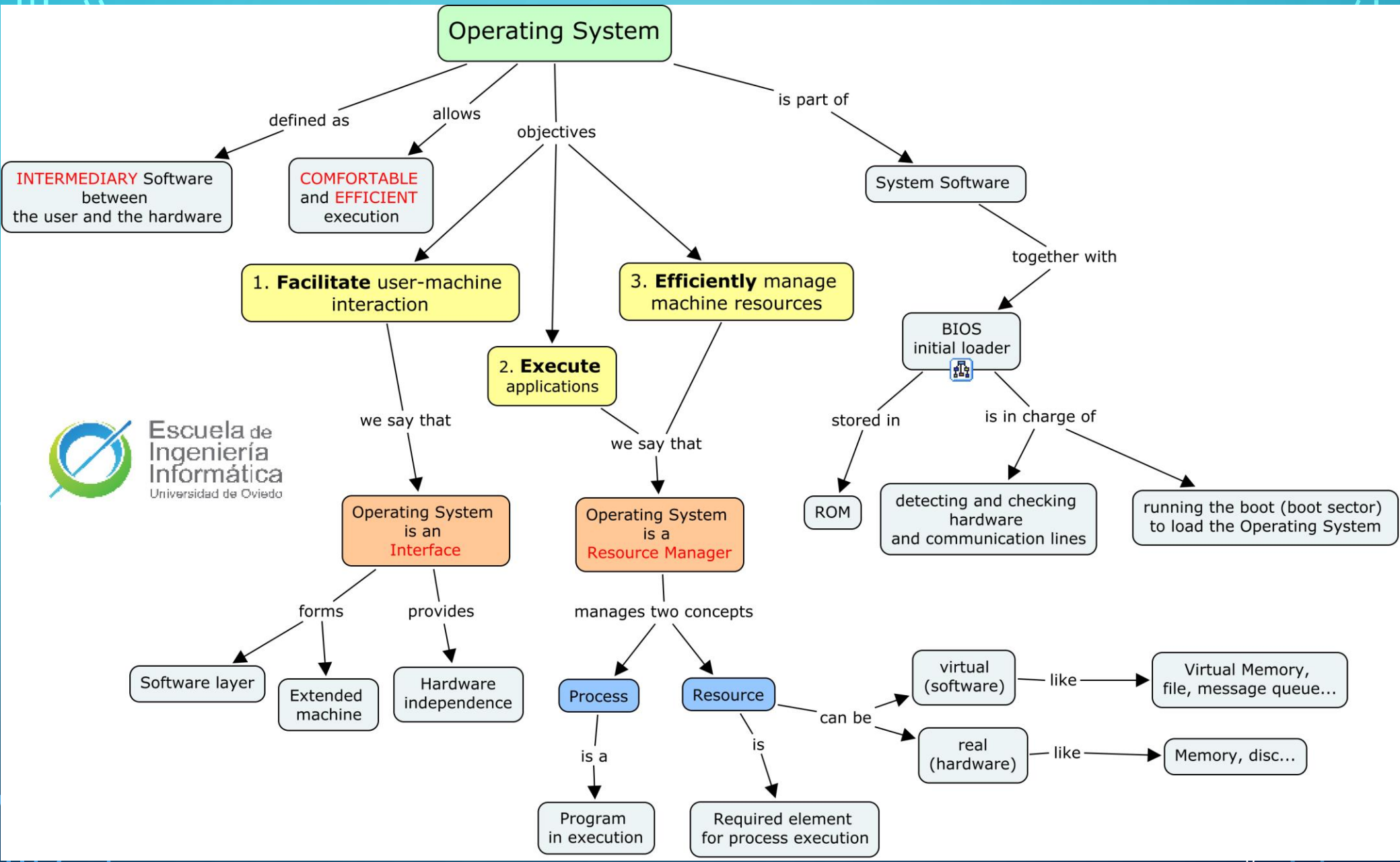
→ *Analyze these questions in groups and give an answer (59 minutes)*

What is an Operating System? It's like …

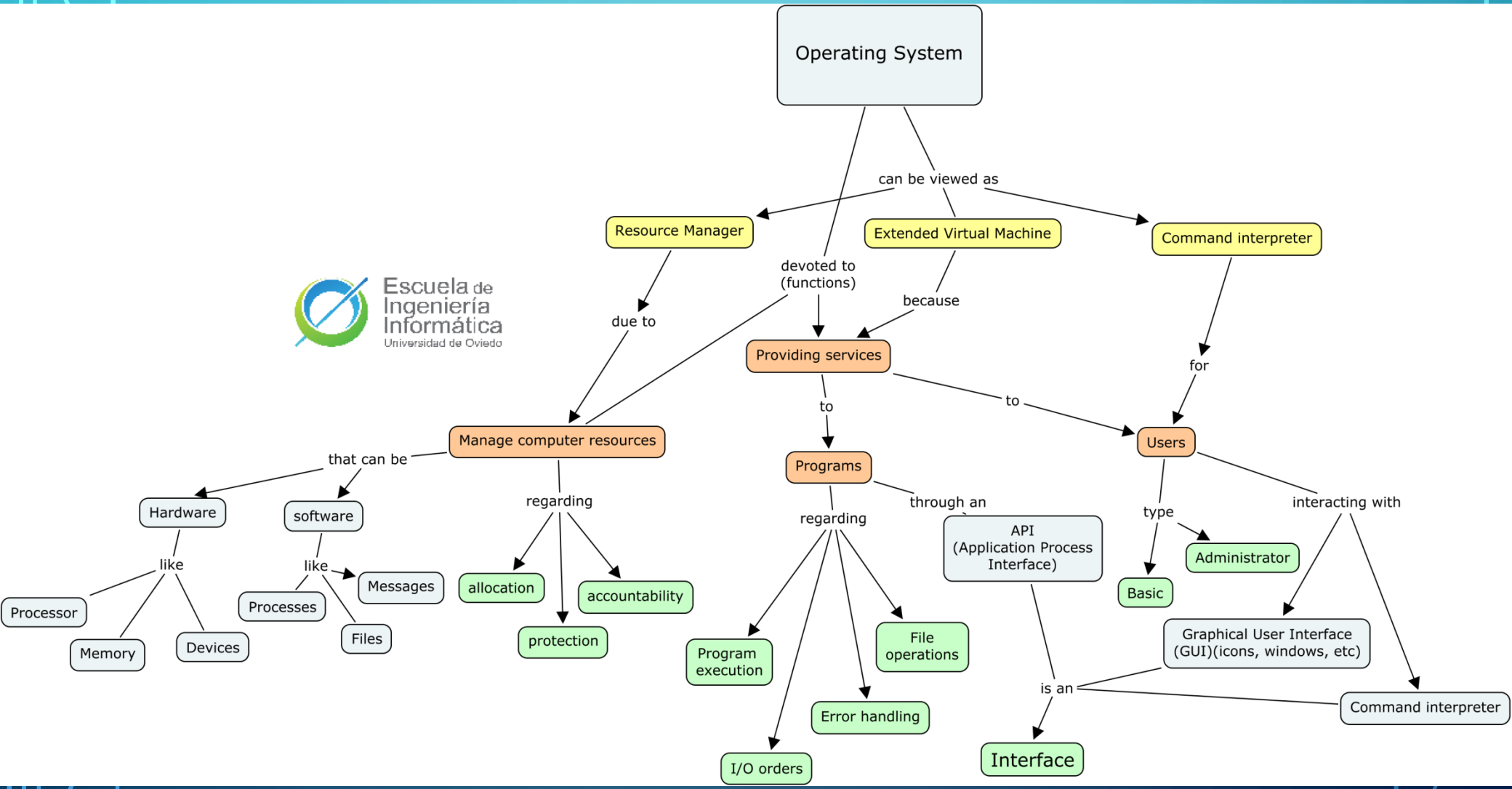
Brain *Orchestra conductor*	*Gives orders to HW* *Coordinates all requested operations optimizing results*
Shop assistant *Intermediary*	*Receives user requests (final or program) and manages them (sending orders to HW)*

1. OPERATING SYSTEM CONCEPT AND FUNCTIONS

- "An operating system is a program which aims to **simplify the management and use** of the computer, making it safe and efficient" *Jesús Carretero*
- The operating system is like the orchestra conductor: **Responsible for coordinating all the individual components of the computer,** so they operate together following a single plan. The operating system allocates computer resources to various programs, synchronizes their activities and provides appropriate mechanisms for programs running in perfect harmony. *Gary Nutt*
- An operating system is a program that controls the execution of application programs and acts as an interface between user applications and the hardware of a computer. An operating system has three objectives: **comfort, efficiency and ability to evolve.** *William Stallings.*


Operating System
defined as
allows
objectives
is part of
INTERMEDIARY Software between the user and the hardware
COMFORTABLE and EFFICIENT execution
System Software
together with
1. Facilitate user-machine interaction
3. Efficiently manage machine resources
BIOS initial loader
2. Execute applications
we say that
we say that
stored in
is in charge of
ROM
detecting and checking hardware and communication lines
running the boot (boot sector) to load the Operating System
Escuela de Ingeniería Informática
Universidad de Oviedo
Operating System is an Interface
Operating System is a Resource Manager
forms
provides
manages two concepts
Software layer
Extended machine
Hardware independence
Process
Resource
virtual (software)
like
Virtual Memory, file, message queue...
can be
real (hardware)
like
Memory, disc...
is a
is
Program in execution
Required element for process execution

1. OPERATING SYSTEM CONCEPT AND FUNCTIONS



CONTENTS

0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

2. OPERATING SYSTEM ACTIVATION

When is the operating system activated?

The computer is turned on, the OS is started up,

the user starts working ...

But, How does the operating system

takes the control again ?

Is its mission just to start the computer?

2. OPERATING SYSTEM ACTIVATION

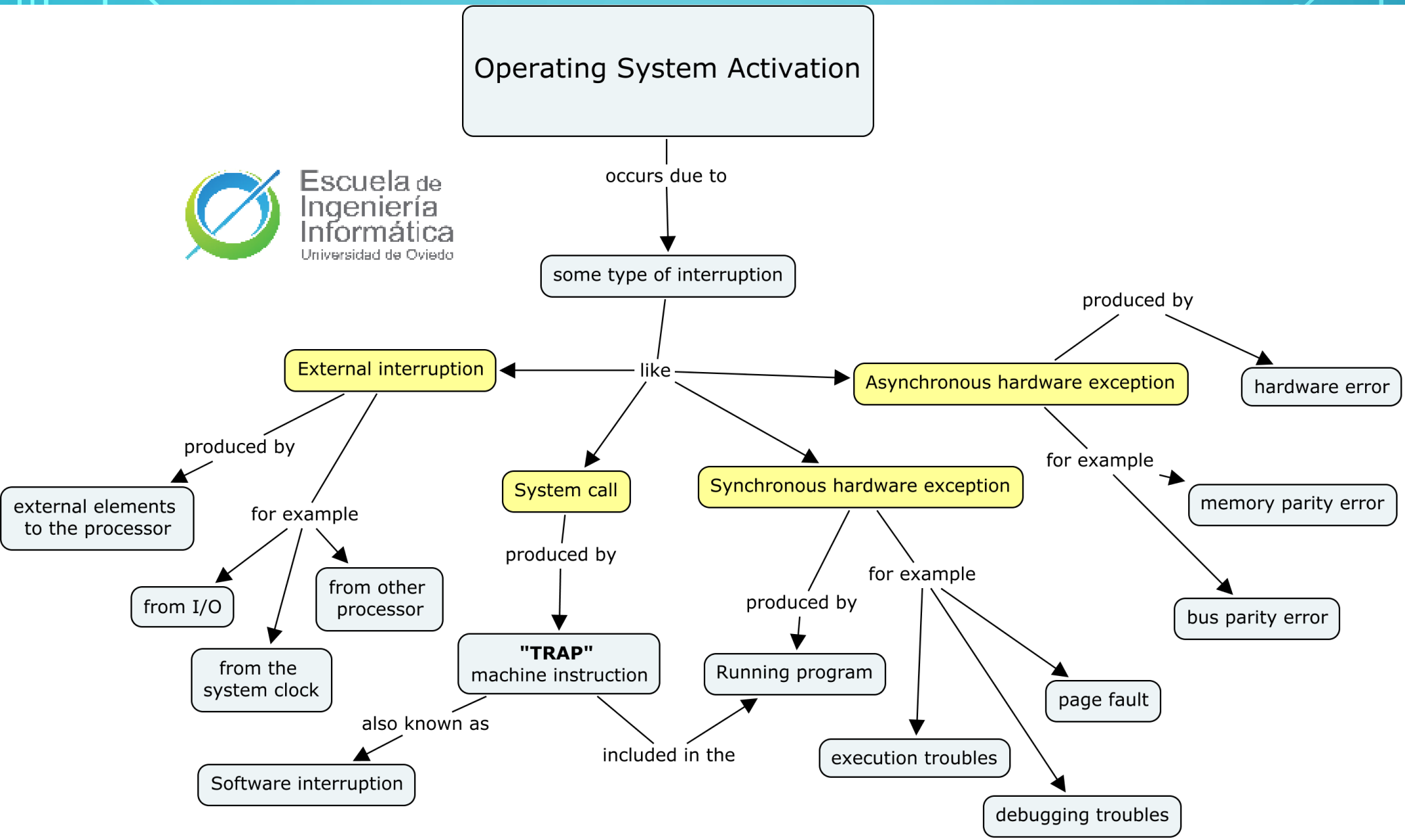
Booting Process

1. Power supply is on. The processor set initial values to its registers (PC, …)
2. The CPU execute instructions stored in ROM-BIOS.
3. BIOS performs POST test (Power On Self Test), checking which devices are connected to the computer and their state.
4. BIOS loads the first sector (MBR-Master Boot Record) of the boot device into main memory and transfers execution to it.
5. MBR code checks the MBR partition table for a bootable partition.
6. MBR loads the boot partition and transfers execution to it.

2. OPERATING SYSTEM ACTIVATION

Booting Process II

7. The boot sector loads the operating system kernel and gives the control to it.
8. In the OS initialization, several routines are loaded into memory, data structures are created and finally some programs are executed (*login program*, for instance).
9. Depending on the hardware and the operating system, this process can be slightly different (in some systems the boot sector loads a secondary, more complex, boot loader).


Operating System Activation
occurs due to
some type of interruption
like
External interruption
produced by
external elements to the processor
for example
from I/O
from other processor
from the system clock
System call
produced by
"TRAP" machine instruction
also known as
Software interruption
included in the
Synchronous hardware exception
produced by
Running program
for example
execution troubles
page fault
debugging troubles
Asynchronous hardware exception
produced by
hardware error
for example
memory parity error
bus parity error

2. OPERATING SYSTEM ACTIVATION

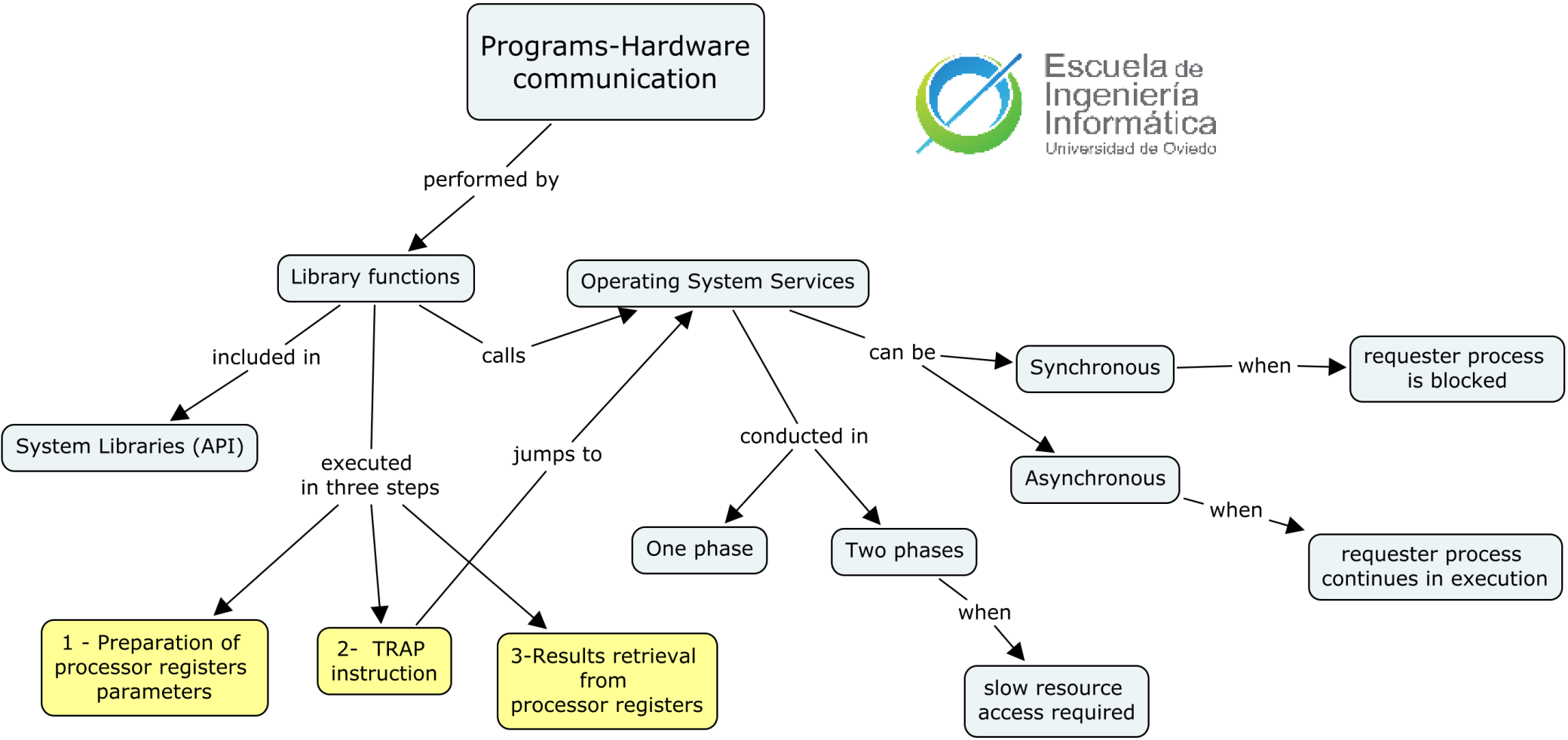
When is the operating system activated?

When an interruption occurs ...

But, How does the operating system

1. Keep the PC and Status register — *HW*
2. Uses the Interruption Vector to access the Interrupt Descriptors Table
   - The Interrupt Descriptors Table has the handling routine address
3. Jumps to the handling routine address

SO


Programs-Hardware communication
performed by
Library functions
included in
System Libraries (API)
calls
Operating System Services
executed in three steps
1 - Preparation of processor registers parameters
2- TRAP instruction
jumps to
3-Results retrieval from processor registers
conducted in
One phase
Two phases
when
slow resource access required
can be
Synchronous
when
requester process is blocked
Asynchronous
when
requester process continues in execution

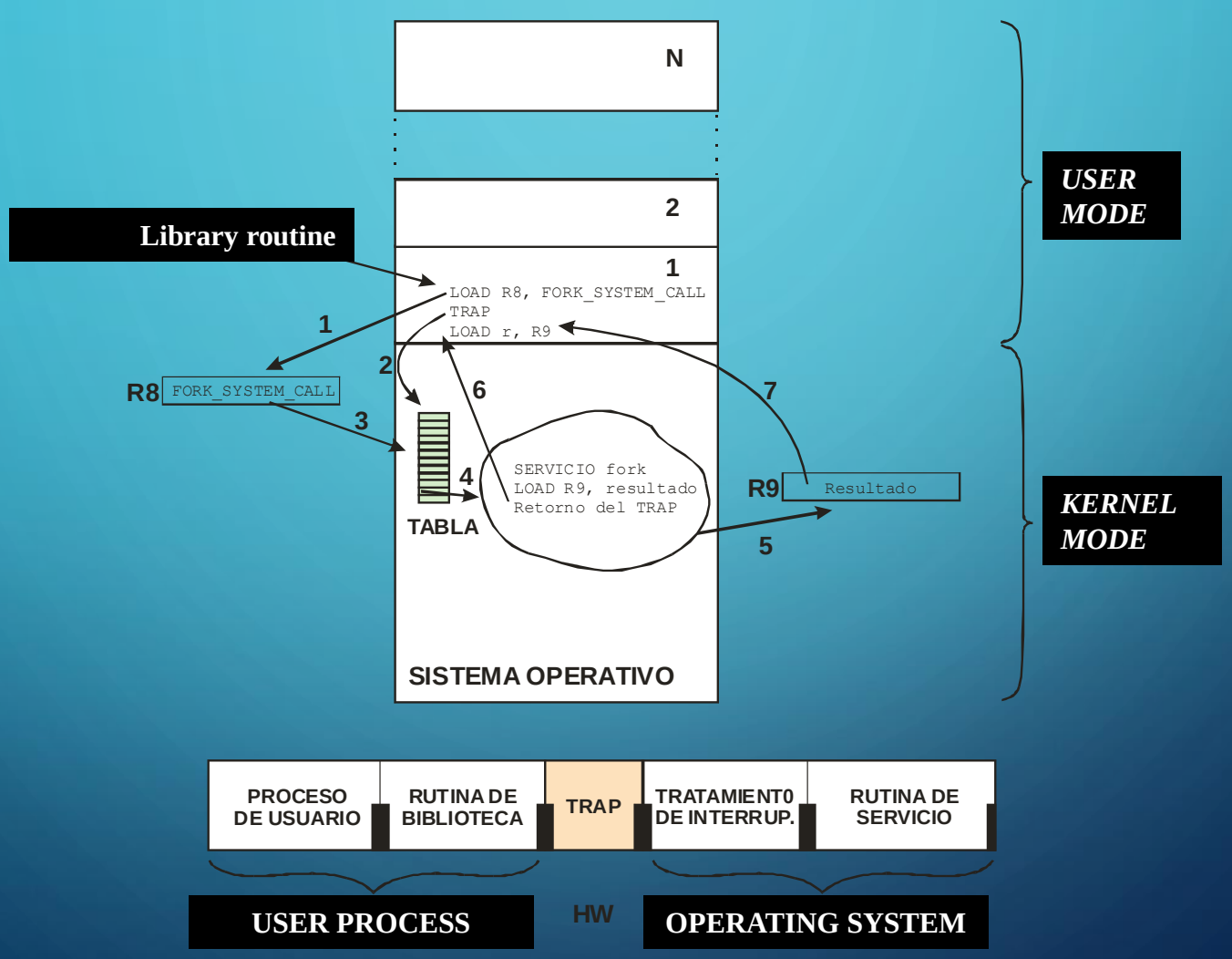
2. OPERATING SYSTEM ACTIVATION SYSTEM SERVICES AND LIBRARY FUNCTIONS

Library System calls (API)

Example: System call for creating a Unix process

```
int fork() {
  int r;
  LOAD R8, FORK_SYSTEM_CALL
  TRAP
  LOAD r, R9
  return r;
}
```

2. OPERATING SYSTEM ACTIVATION SYSTEM SERVICES AND LIBRARY FUNCTIONS



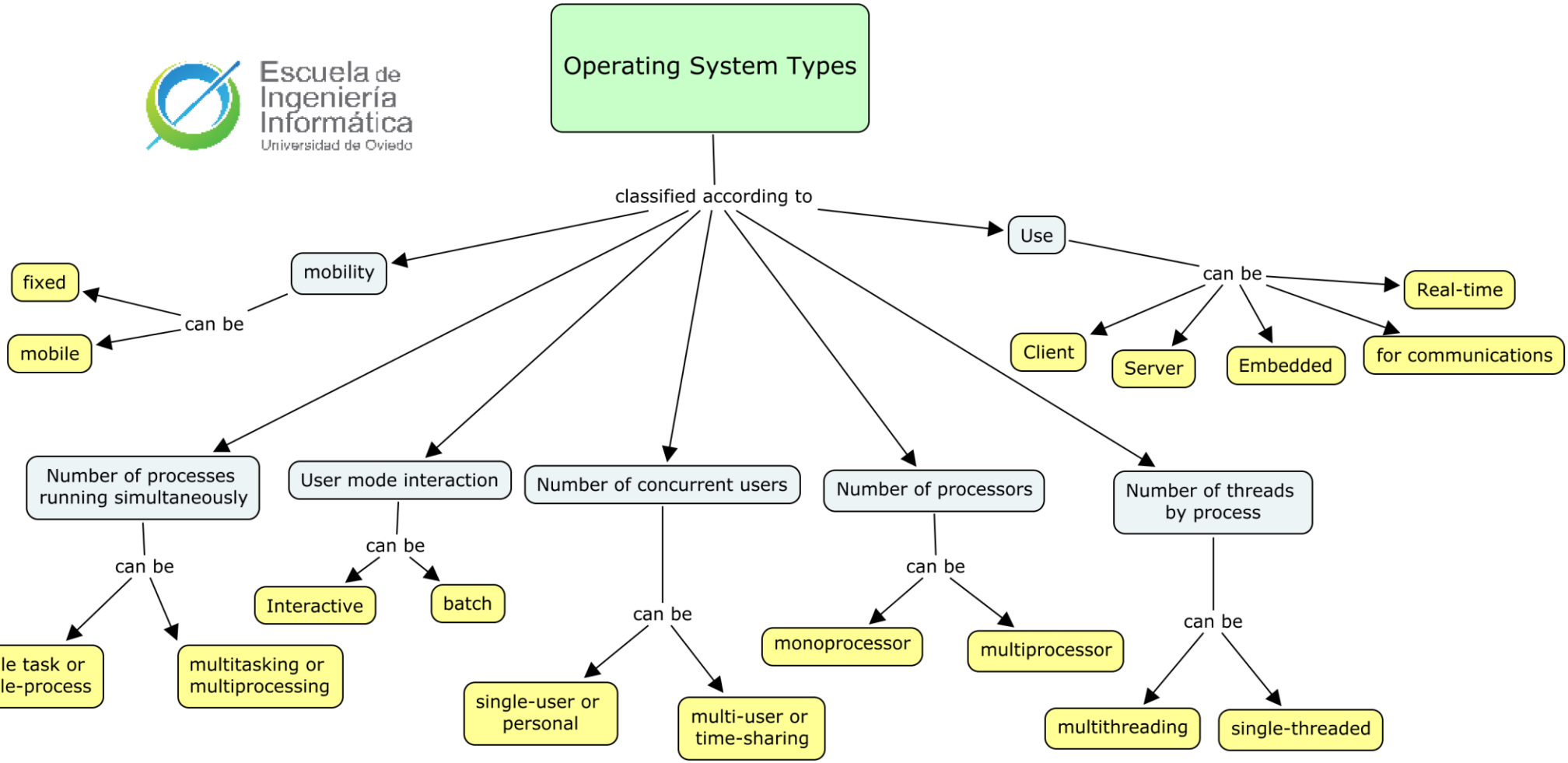
CONTENTS

0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

3. TYPES OF OPERATING SYSTEMS

- *What types of Operating Systems do you know?: Analyze the meaning of the following types:*
  - *Single task / Multitasking*
  - *Single-process / Multiprocessing*
  - *Interactive / Batch*
  - *Single User / Multiuser*
  - *Uniprocessor / Multiprocessor*
  - *Embedded*
  - *Real-time*
  - *Mobile*

3. TYPES OF OPERATING SYSTEMS



CONTENTS

0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

4. COMPONENTS OF OPERATING SYSTEMS

1. From a functional point of view the OS contains
   - Process management system.
   - Memory management system.
   - Device management system.
   - File Management System.
   - Security and Protection

What possible services may each of these systems offer?

… we will analyze them throughout the course topics

CONTENTS

0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

5. INTERFACES OFFERED BY THE OS

1. Interface to the application developer or API (Application Program Interface)
   - Standard POSIX-> Single Unix Specification
     - This is a specification, not an implementation
   - Win32 Interface
     - This isn't a standard, it is a Microsoft implementation

2. User Interface
   - Command line
   - GUIs
   - Command files or shell-scripts
     - Bourne in Unix,
     - Cmd-line, Jscript, Windows PowerShell …

CONTENTS

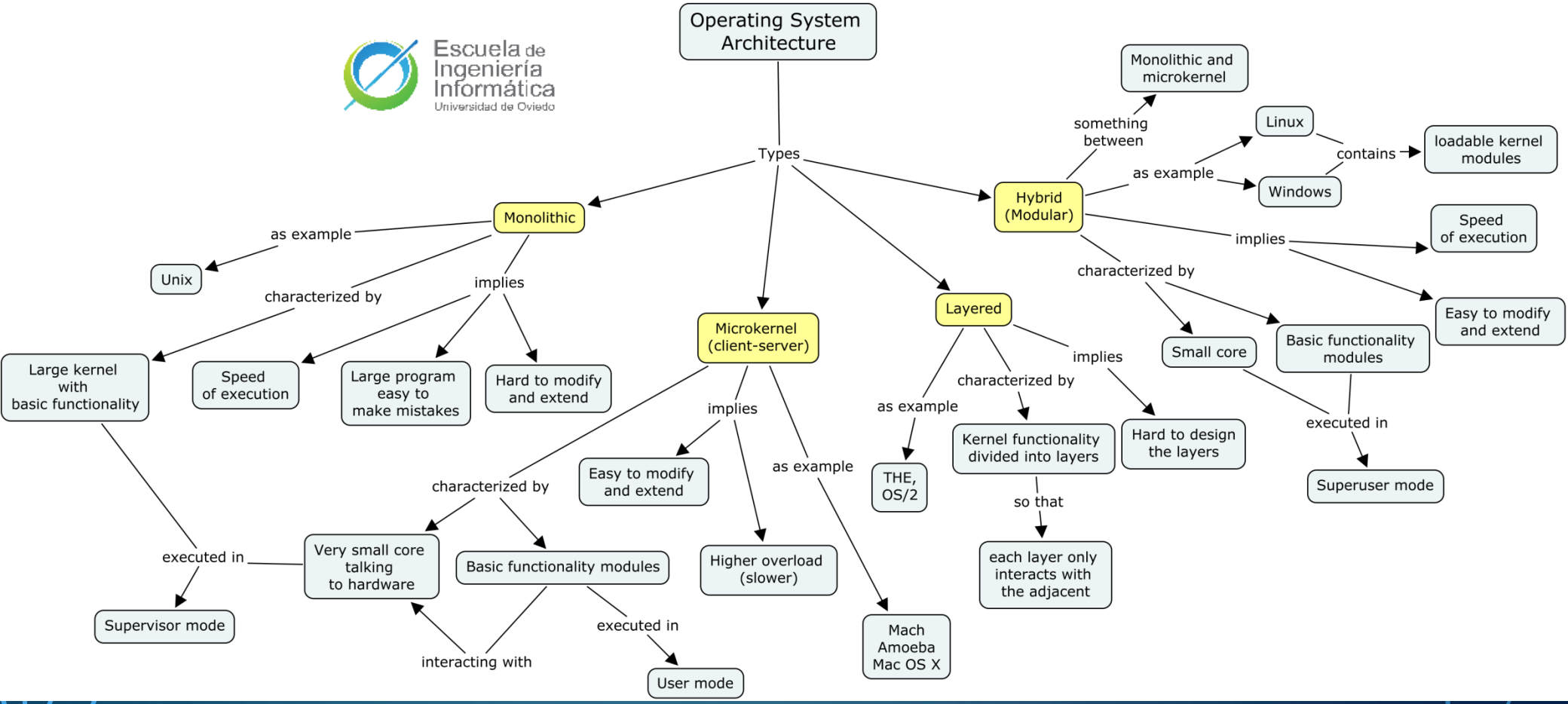
0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

6. DESIGN OF OPERATING SYSTEMS

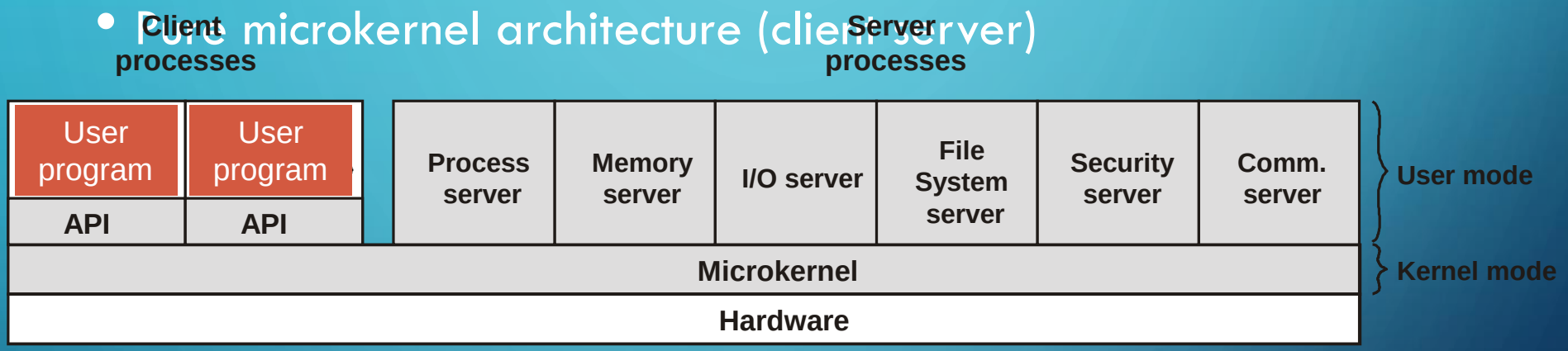
Operating System Architecture

- Monolithic Operating Systems
  - Functionality into a single program
  - Running in kernel mode
  - Hard to modify
- Structured Operating Systems
  - Organized in layers
  - Microkernel (Client / Server)
- Hybrid Operating Systems
  - Modular
  - Non pure Microkernel
- Distributed Operating Systems
  - Pure
  - Middleware

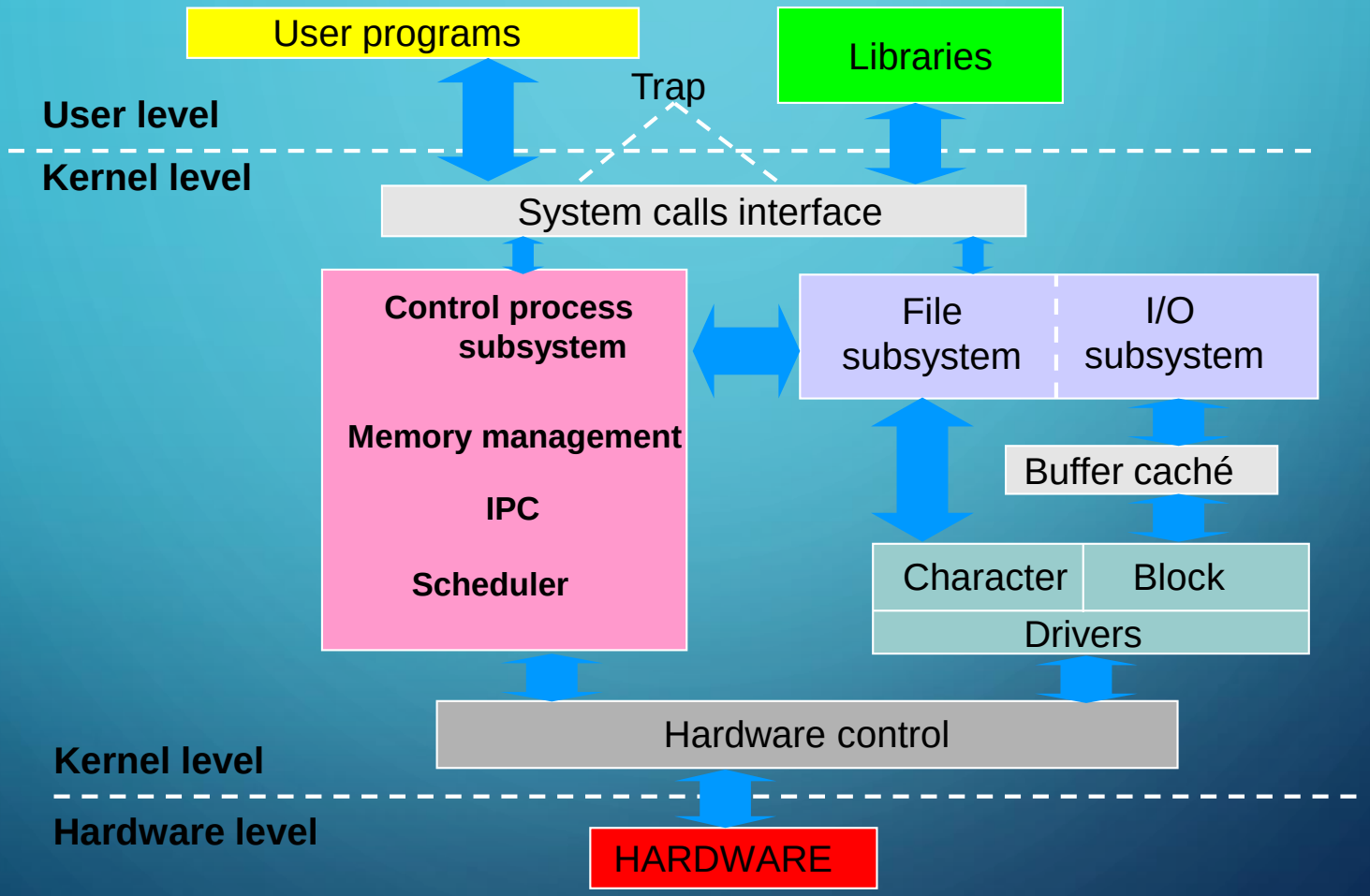
6. DESIGN OF OPERATING SYSTEMS



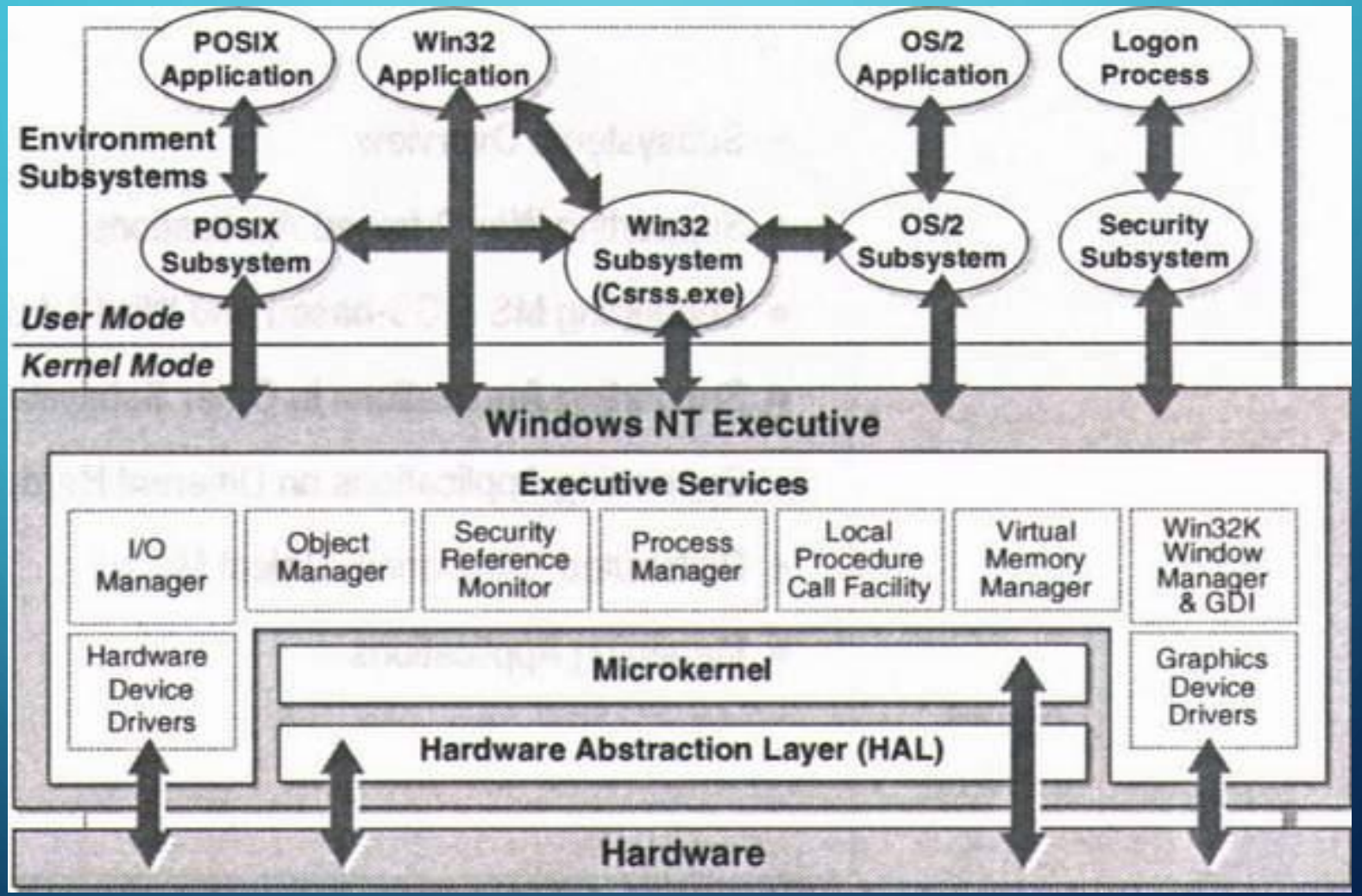
- Pure microkernel architecture (client server)



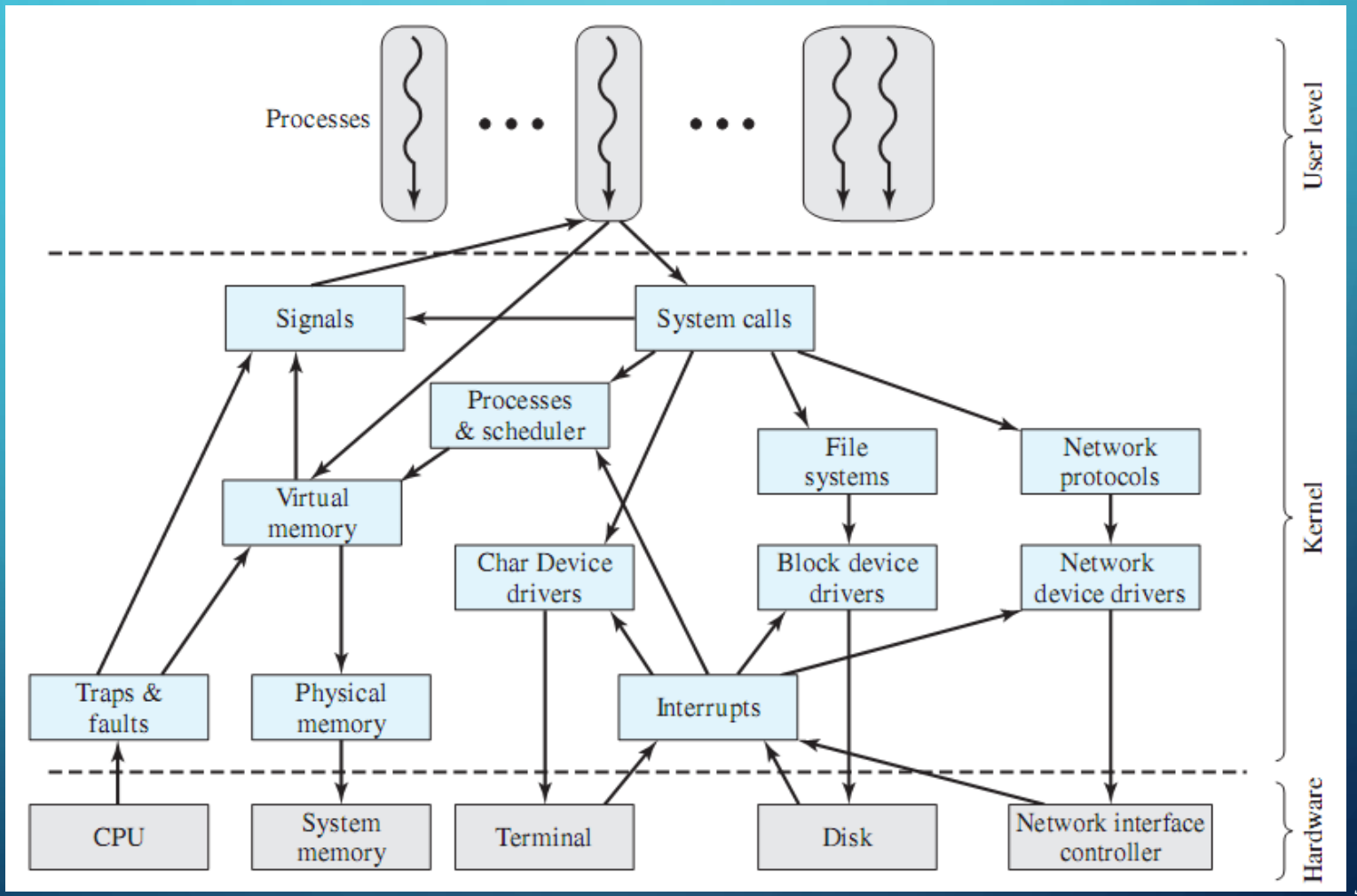
INTERNAL STRUCTURE OF THE UNIX OS



6. DESIGN OF OPERATING SYSTEMS
INTERNAL STRUCTURE OF WINDOWS NT



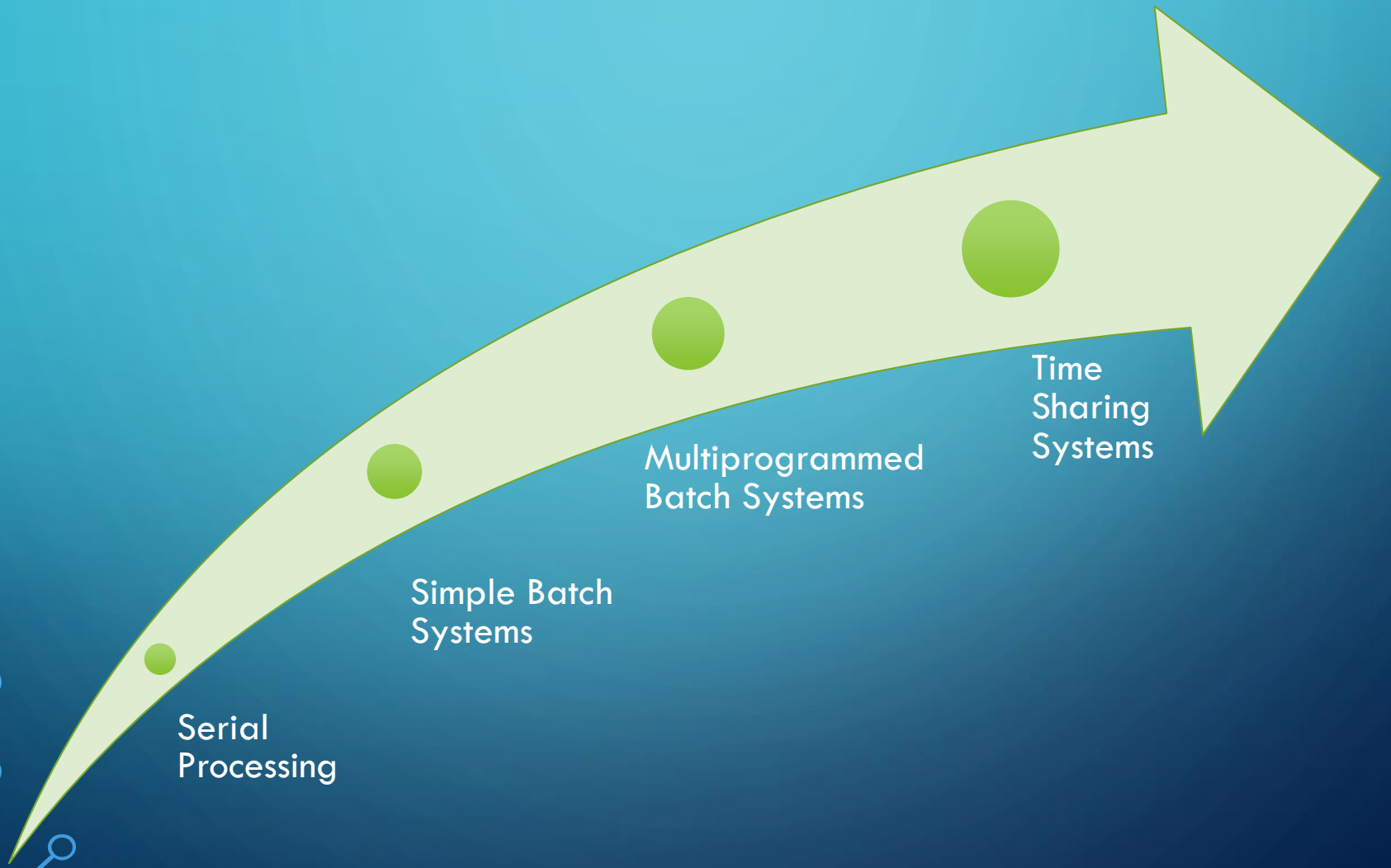
LINUX STRUCTURE



CONTENTS

0. Conceptual maps overview
- Operating System concept and functions
- Operating System Activation
- Types of Operating Systems
- Components of Operating Systems
- Interfaces offered by the OS
- Design of Operating Systems
- Operating Systems evolution

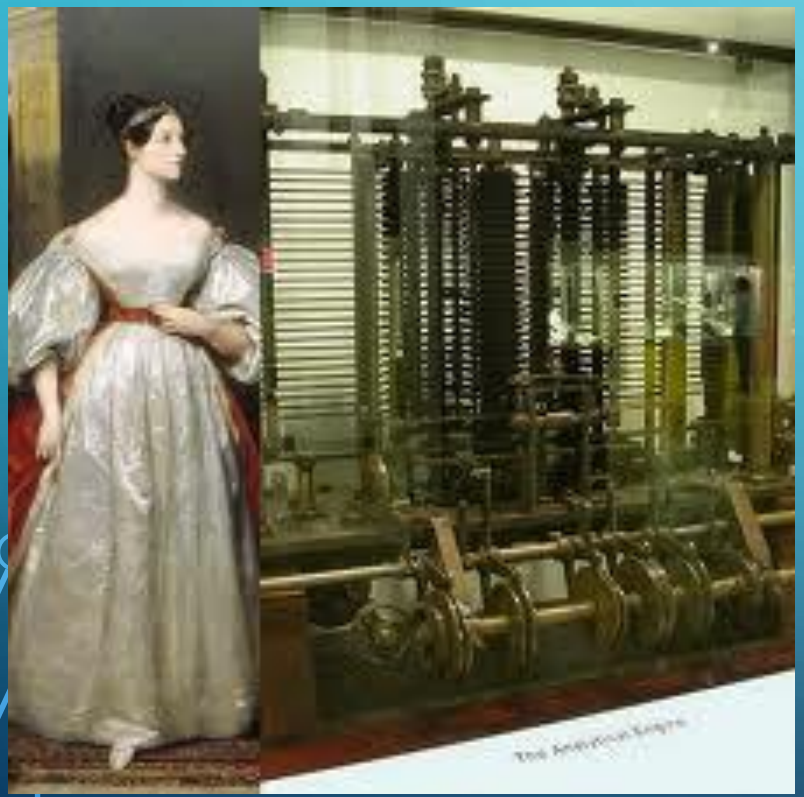
7. Operating Systems Evolution



7. OPERATING SYSTEMS EVOLUTION

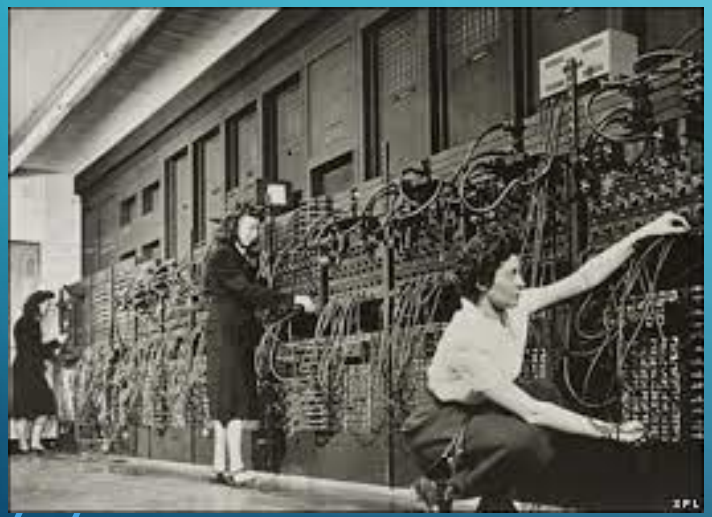
Period	Operating System Features	HW/ OS / Languages
The 40s prehistory	Nonexistent. Serial processing.	Vacuum valves Von Neumann Arch. ENIAC (1943) Assembler
The 50s 1st Generation	Resident monitor: • Loading and executing programs • Processing batch jobs • I/O Routines (device drivers) • Error Recovery • Control Language	Transistors IBSYS (IBM) FMS (IBM) FORTRAN, COBOL

In the beginning, the programming was done by the women
The first programmer was a woman: Ada Lovelace

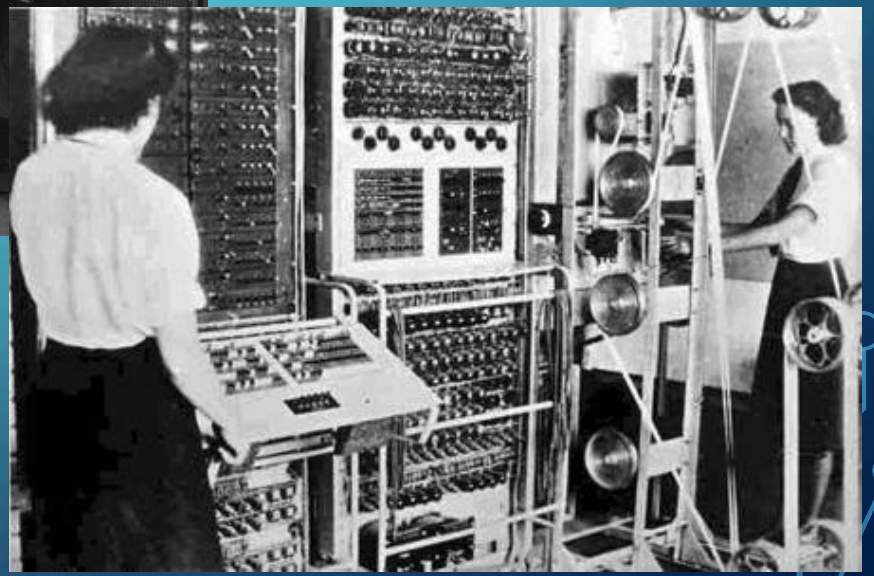


Colossus was used by Alan Turing to decipher messages from the Germans in WWII.
ENIAC was developed by the Ballistic Department of the U.S.

ENIAC



COLOSUS



7. OPERATING SYSTEMS EVOLUTION

Period	Operating System Features	HW/ OS / Languages
The 60s 2nd Generation	Performance improvements: • Multiprogramming (I/O by DMA) • Interactive Multiuser: Timesharing • Real Time • Multiprocessor Large and expensive Complex control language	Integrated Circuits PDP-8 Digital IBM 360 CTSS (IBM 7090) OS/360(IBM 360) MULTICS – UNIX BASIC, ALGOL

Period	Operating System Features	HW/ OS / Languages
The 70s	General purpose OS Dissemination of multiuser timesharing **Unix** •ATT Bell Laboratories •Ken Thompson, Dennis Ritchie •C language implementation (Dennis Ritchie) (1973) •Diffusion Lab. and universities (Source Code) •Different distributions •Appearance of BSD (Berkeley Uni) •Emergence of System V (Bell Lab) •Other manufacturers (Sun, HP, IBM) …	PDP-7,11 Apple II Intel 8008 UNIX (Bell 1976) MVS (IBM) - Mainframes VM (IBM) Minicomputers CP/M- PCs

Period	Operating System Features	SO ejemplo
The 80s	Simplification of the OS. Importance of the user Network management Network Operating Systems Graphical User Interfaces (GUIs) Internal Object Oriented Design Different OS for different PC processors Distributed Operating Systems	UNIX MS-DOS Windows, Amiga Mc OS, OS/2 Mach, Chorus, Amoeba

Period	Operating System Features	SO ejemplo
The 90s	Free Operating Systems Real-time Operating Systems Operating Systems with parallel processing Layers of middleware (Middleware) Client / Server architecture Interfaces standardization Inclusion of multiple programming interfaces Security, Cryptography Operating systems for teaching	Linux (91) FreeBSD QNX CORBA, DCOM Windows XP, NT.. POSIX MINIX, SOS, NACHOS

Period	Operating System Features	SO ejemplo
Present and future	Embedded operating systems: phones, cars, PDAs Parallelism Distributed Computing Fault Tolerance Development of new interfaces Customization and Usability Open Systems Object Oriented Design Multiple Personalities Distributed client-server architecture Incorporating many features of security and remote access Significant improvements in security (hopefully)	Linux (RedHat Slackware SuSe…) Windows Longhorn Windows CE

7. OPERATING SYSTEMS EVOLUTION