Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informática de Sistemas y Computadoras (DISCA)
Universitat Politècnica de València

Part 1: Introduction

Unit 1

Operating System Concept

fSO

Goals and bibliography

- Goals
  - Introducing the Operating System (OS) **concept**
  - Describing the **functions** an OS perform
  - Reviewing the evolution of operating systems to help understanding what services an OS provides and how they are provided
- Bibliography
  - A. Silberschatz, P. B. Galvin. Chapters 1 and 2
  - Wikipedia and other Internet sources:
    - A Brief History of Computing - Operating Systems https://trillian.randomstuff.org.uk/~stephen/history/timeline-OS.html
    - Timeline: 40 years of OS milestones http://www.computerworld.com/article/2531905/operating-systems/timeline--40-years-of-os-milestones.html

- **Operating system concept**
- Operating system structure
- CPU utilization
- Historical evolution

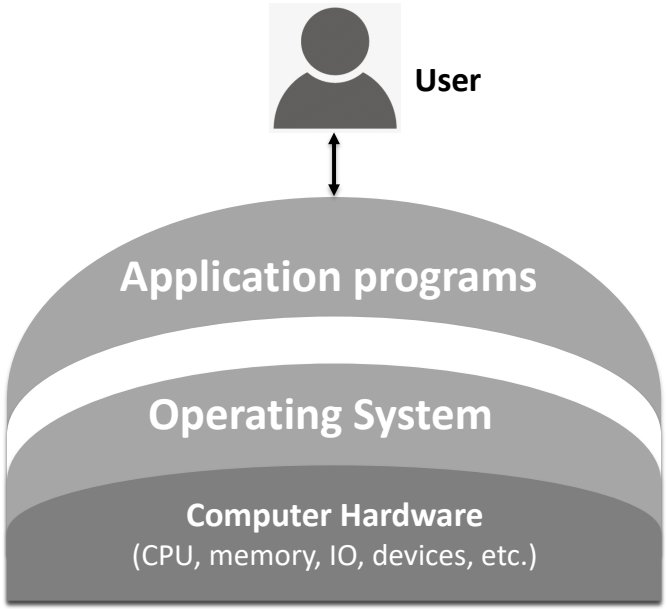
Terms:	
OS	Operating system
I/O	Input / Output
CPU	Central processing unit (processor)
API	Application Programming Interface
HAL	Hardware abstraction layer
UNIX	Portable, multitask and multiuser operating system
Linux	Free and open SO kernel based on UNIX

Operating system concept

- A computer system is the set of hardware elements, organized according to a specific "Architecture", that define a computing unit
  - The direct management of these hardware elements is complex and depends on the devices features
  - These hardware elements have limited capabilities and this leads to the need to establish operating criteria that optimize their use

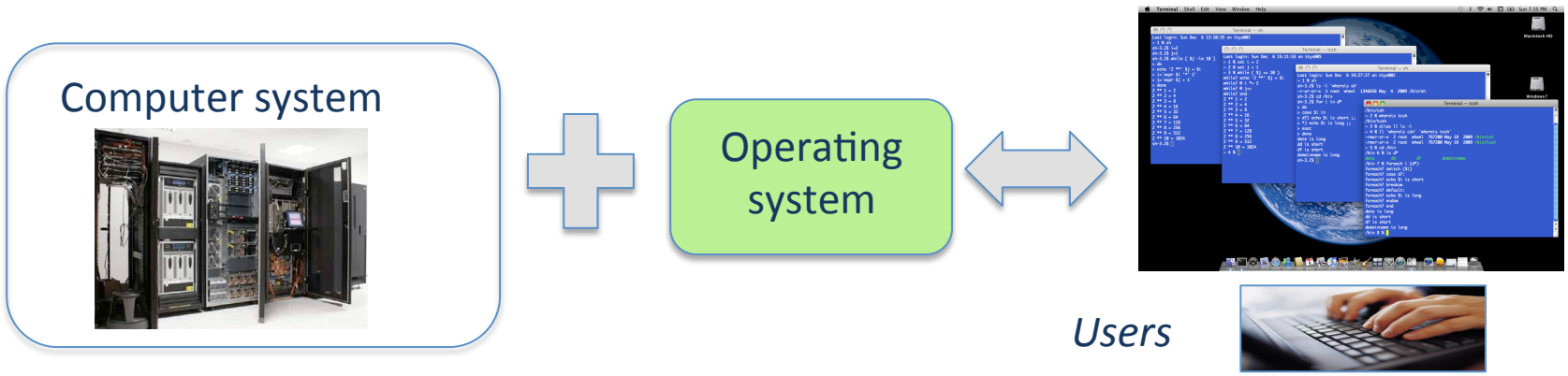
Computer system concept

- **Computer system structure** can be divided into four components:
  - **Hardware** – provides basic computing resources: CPU, memory, I/O devices
  - **Operating system**: 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
  - **Users**
    - People, machines, other computers

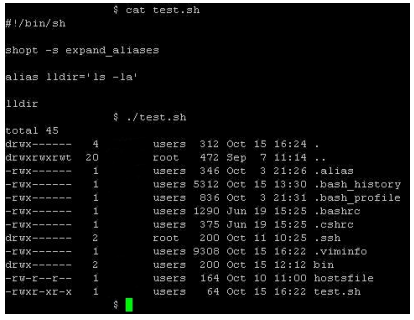


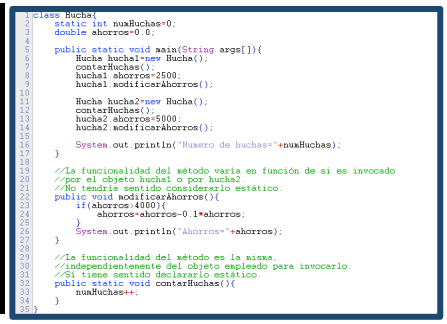
Operating system concept

- Definition
  - **An OS is the set of software** that allows the operation of computer systems and offers a friendly interface to users
- Purpose
  - Creating a **comfortable an efficient environment** to run programs
- OS goals: accessibility, commodity, efficiency, security, portability, etc
  - It acts as the **intermediate** between the user and the system
  - It guaranties the **correct computer operation**
  - It **easies the application creation** task for programmers
  - It manages **efficiently** the hardware resources available



- An OS should provide **services** to the several kinds of computer **users**
- User types:
  - Application user
  - Application programmer
  - System programmer
  - System administrator





Operating system concept

- Goal
  - The **operating system abstracts and manages** the operation of all system hardware/software components that make up the computer system
    - **System view**: Resource Manager and protection
      - SO components and their interrelation
    - **User view**: Abstraction of resources aimed at facilitating its use -> Extended machine
      - Services provided
      - Interfaces provided to programmers and final users

Operating system concept

- Functions
  - Providing user, programmer and administrator interfaces
    - **Hardware Abstraction**
  - Offering a range of services in the form of "**system calls**"
  - **Manage resources.** Is responsible for deciding which program may use a hardware device and for how long
    - Process, memory, files and I/O management
  - **Security and protection:** Controlling and supervising resource access to avoid conflicts and unauthorized access

- Systems with OS

Server Pool

Computer Server

Personal Computer

Router

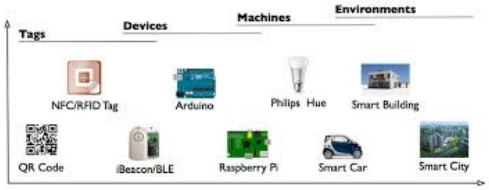
Tablet

Smart phone

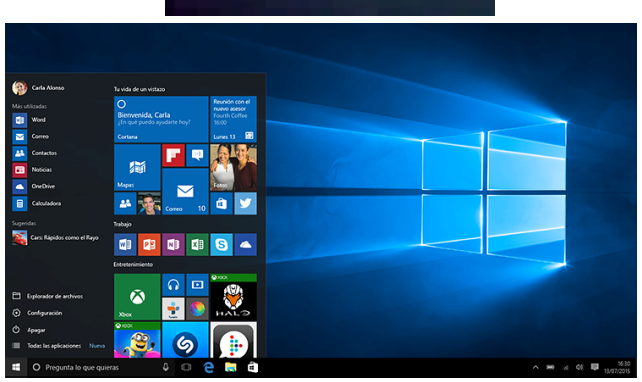
Video Console

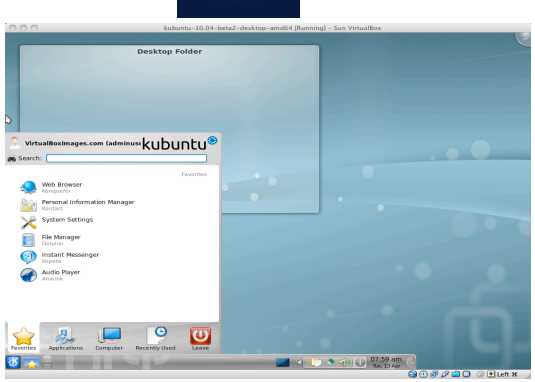
Smart TV

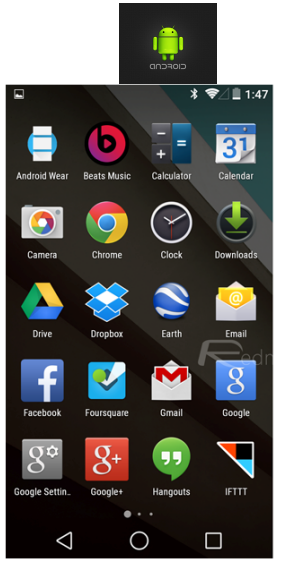
Smart Devices & IoT



- Nowadays OSs



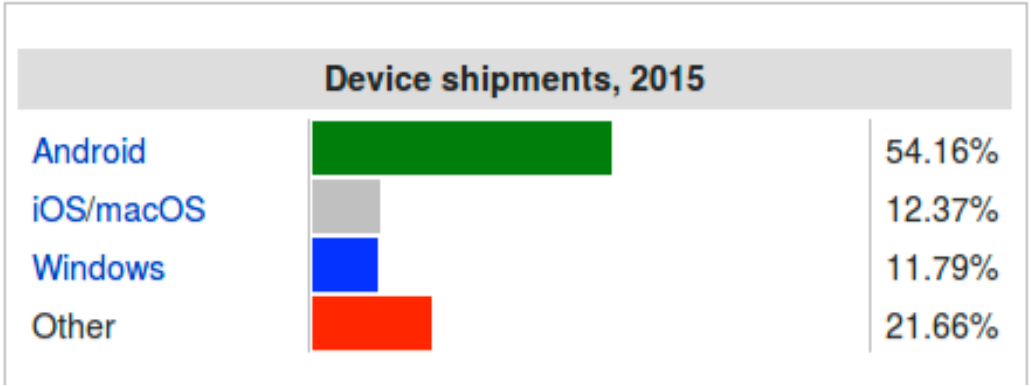




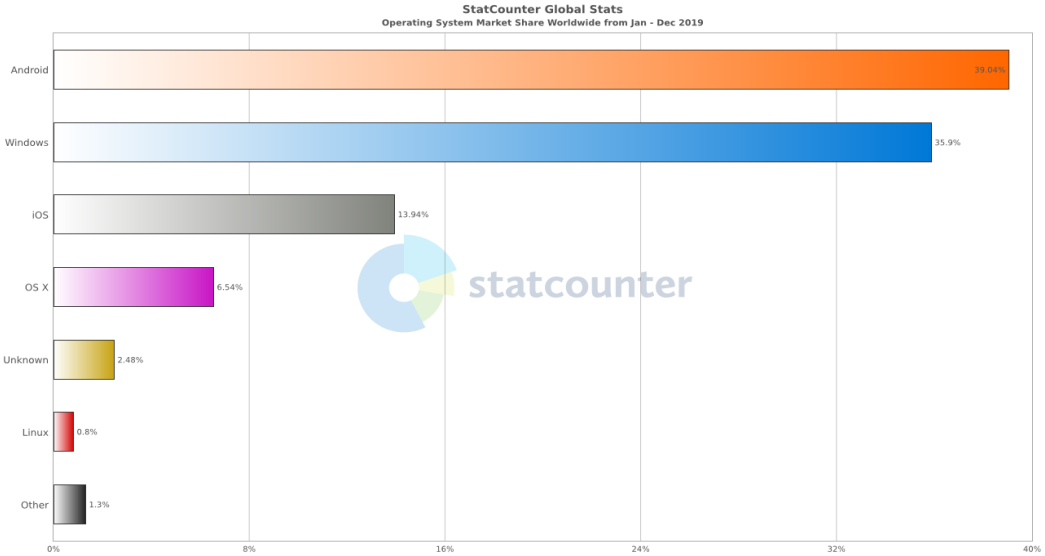
Operating system concept

- Embedded OSs

- OS use statistics
  - *Gartner* [1] (2015)
    - *Smartphones, tablets, laptops and PCs together*



  - *Statcounter* [2] (2019)



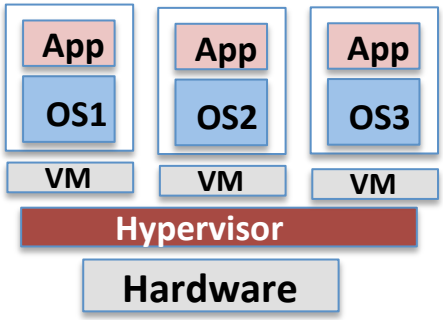
Fuente:
[1] Wikipedia <https://en.m.wikipedia.org/wiki/Usage_share_of_operating_systems>
[2] Statcounter. Operating System Market Share Worldwide. Junio 2017. <http://gs.statcounter.com/#desktop-os-ww-monthly-201508-201508-bar>

Operating system concept: trends

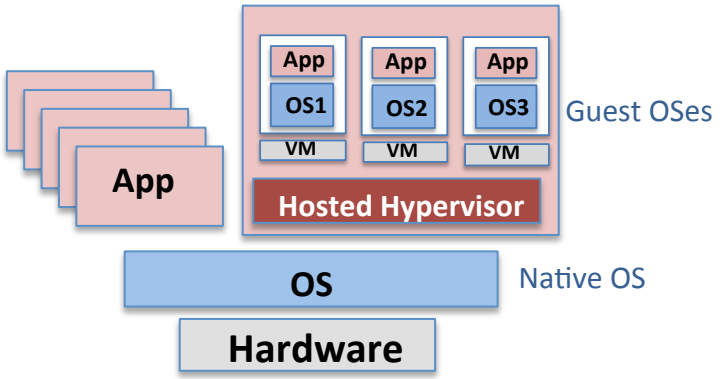
- **Virtualisation**
  - Allows offers **hardware abstraction** to run several **Execution Environments** (OS + applications) on the same **hardware platform**
  - It can be performed by:
    - **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
    - **Virtualisation** OS natively compiled for CPU, running **guest** OSes also natively compiled
  - **Hypervisors** or **VMM** (virtual machine Manager) provides virtualization

Operating system concept:trends

- **Virtualisation** based on hypervisor
  - **Type 1**: **Bare-metal hypervisor** is a layer of software **on top of a physical hardware** that provides virtual machines close to the native one. Example: Xen



  - **Type 2**: **Hosted hypervisor** is an **application on top of a native OS** that provides virtual machines. Example: VirtualBox

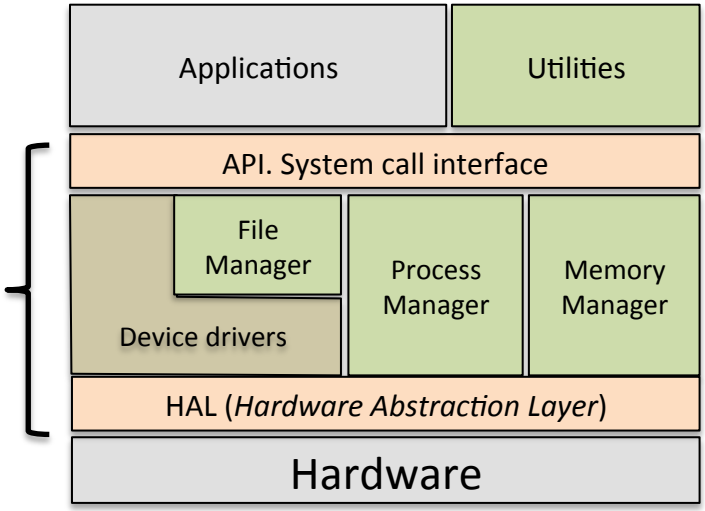


- Operating system concept
- **Operating system structure**
- CPU utilization
- Historical evolution

Terms:	
OS	Operating system
I/O	Input / Output
CPU	Centra processing unit (processor)
API	Application Programming Interface
HAL	Hardware abstraction layer
UNIX	Portable, multitask and multiuser operating system
Linux	Free and open SO kernel based on UNIX

Operating system structure

- **Kernel**
  - File Systems
  - Memory Manager
  - Process Manager
  - Device drivers



- **System Utilities**:
  - They extend the OS providing key utilities not included in the OS kernel
    - Shell, GUI, Monitoring, Maintenance, Administration ...

Operating system structure

- Kernel architectures
  - **Microkernel**: provides only the basic hardware abstractions and minimal services.
    - The resource usage policies are implemented as "servers" that run in user space. It has been much debate about their efficiency problems.
    - Examples: Mach, QNX
  - **Monolithic**: All kernel components are in the same address space.
    - Only one program contains the whole kernel functionality (it has to be recompiled after every change).
    - Example: Linux
  - **Hybrid**: This is a modified microkernel that includes not essential components whose execution speed is critical.
    - Examples : Windows NT, XNU (Mac OSX)
  - ….

- Operating system concept
- Operating system structure
- **CPU utilization**
- Historical evolution

Terms:	
OS	Operating system
I/O	Input / Output
CPU	Centra processing unit (processor)
API	Application Programming Interface
HAL	Hardware abstraction layer
UNIX	Portable, multitask and multiuser operating system
Linux	Free and open SO kernel based on UNIX

CPU utilization

- **System workload**
  - The workload of a computer system consists of a set of programs to be executed
  - In a simplified description a program execution can be seen as a sequence of CPU and I/O bursts
    - CPU burst → time interval required to perform consecutive CPU operations by a program
    - I/O burst → idem with I/O operations

- CPU bound and I/O bound programs
  - A program may be **bound by the CPU speed**
  - A program may be **bound by the I/O speed**

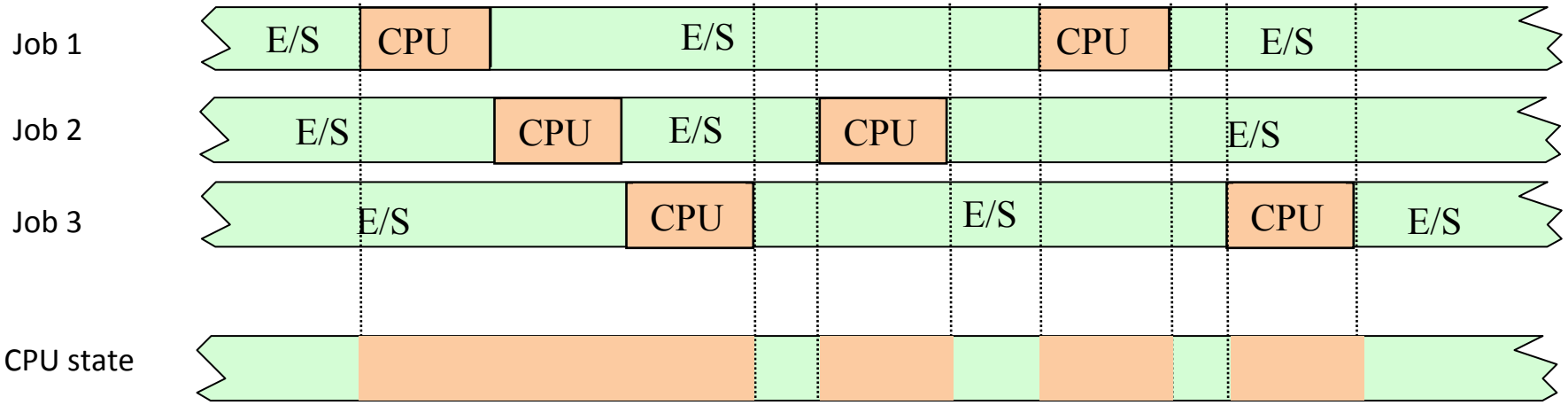
CPU utilization

- **CPU utilization concept**
  - The CPU is the main computer component
  - OSs have to achieve that the CPU be active as much as possible
  - **CPU utilization**: Fraction of time when the CPU is active in relation to the whole required to end the system tasks

$$\text{CPU\_utilización} = \frac{\text{CPU busy time}}{\text{Total time}}$$

CPU utilization

- Multiprogramming
  - Alternative use of the CPU by running programs
    - When a process is blocked waiting for a pendent I/O operation, the CPU executes instructions from another ready process
    - A "context switch" is performed when an I/O operation is demanded
  - CPU utilization increases
  - The system performance increases: more jobs end with less time



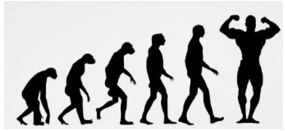
- Operating System Concept
- Operating System Structure
- CPU utilization
- **Historical evolution**

Terms:	
OS	Operating system
I/O	Input / Output
CPU	Centra processing unit (processor)
API	Application Programming Interface
HAL	Hardware abstraction layer
UNIX	Portable, multitask and multiuser operating system
Linux	Free and open SO kernel based on UNIX

Historical evolution

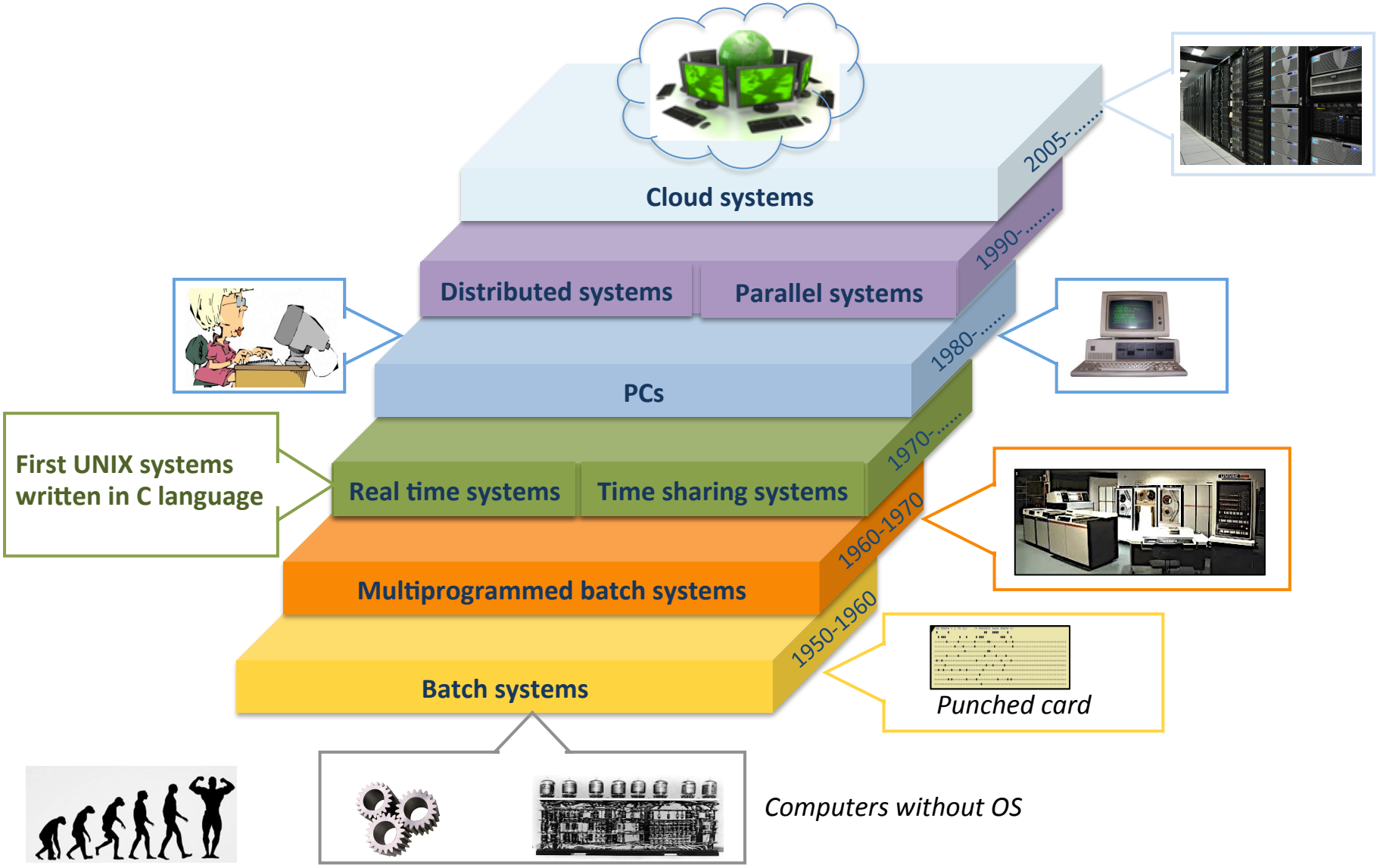
- OS capabilities

Single user: Only one system can be active at a given time	Active users support	**Multiuser**: Several users can be active at the same time
No direct user-machine interaction: Only dedicated machine operators can directly deal with the system	Direct user-machine operation support	**Direct user-machine interaction:** Users can dialog with the system posting commands and waiting form immediate response
Text mode: Users interact with the system with text commands and receiving text response	User interface (UI)	**Graphic mode**: The system offers a graphic user interface (GUI) made of windows, icons and menus
Single task: The OS performs tasks sequentially, a task must wait to the previous one to finish before starting	Tasks support	**Multitask:** Several tasks can be active in the system simultaneously
Single processor: The OS can only work with one CPU	Processors support	**Multiprocessor**: The OS can work with several CPUs simultaneously



Historical evolution

- The OS evolution is conducted by the technological innovations on computer architecture, communications and storage media



Historical evolution

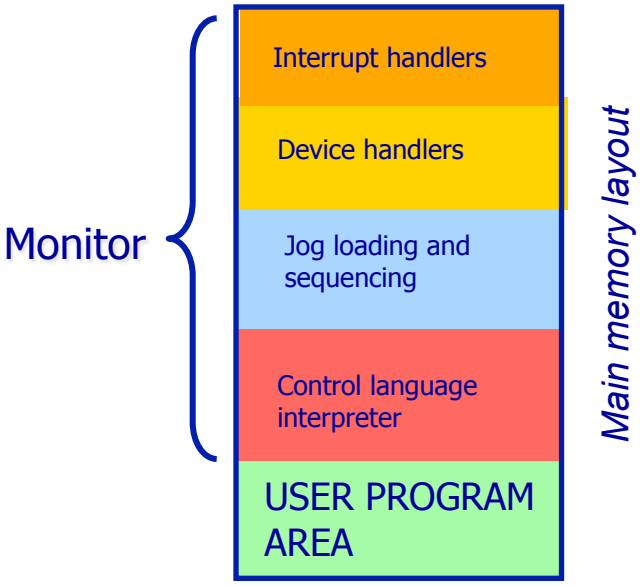
- First OSs: Batch systems

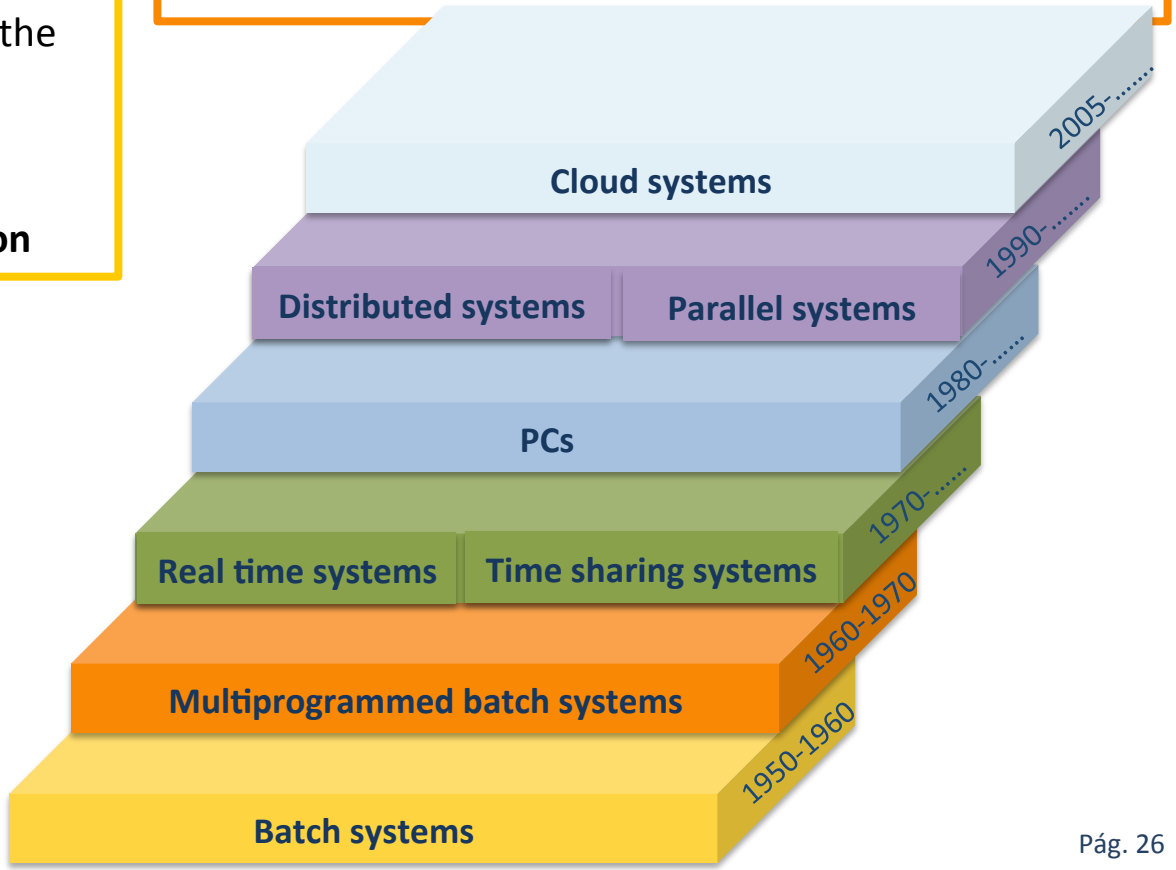
Basic batch systems

- Jobs are processed sequentially: the CPU is idle when the active jobs is performing I/O
- Low CPU utilization
- Resident monitor that automatizes some tasks: job ending, error treatment, loading and executing the next job
- Batch processing
- I/O Access
- **No direct user-machine interaction**

Multiprogrammed batch systems

- Job/CPU scheduling
- Multiprogramming
- Memory management and protection based on fixed memory partitions
- Disk Management
- **No user-machine interaction**





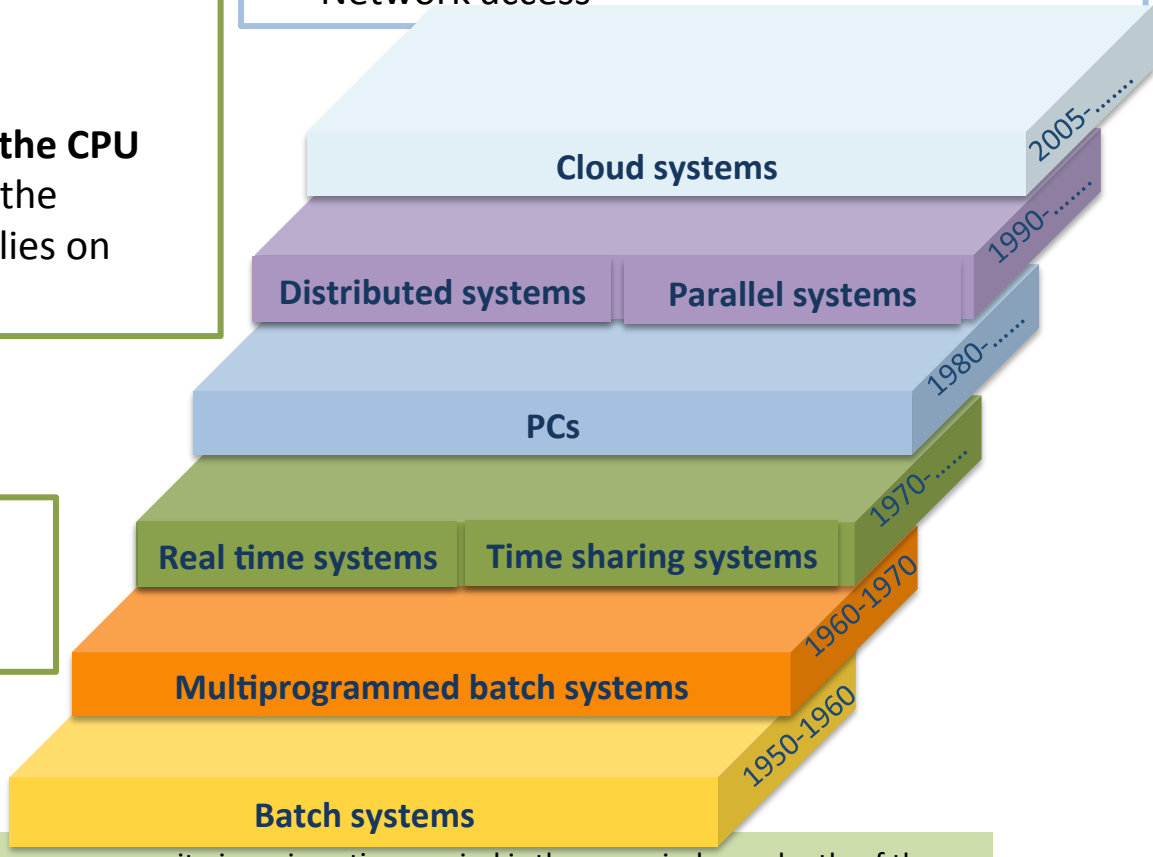
Historical evolution

- Modern OSs

Time sharing systems
- **Direct user-macnine interaction with multiprogramming**
- Jobs synchronization and communication
- File systems that manage files
- Protection
- Virtual memory
- Process scheduler: The OS **limits ethe CPU occupancy** by a process by means the **context switch** mechanism that relies on timer interrupts

PC systems
- Personal use
- Friendly user interfaces based on windows and mouse
- Multimedia capabilities
- Plug-and-play support
- Network access

Real time systems
- For executing tasks with a fixed deadline



Corbato's law: The number of lines that a programmer can write in a given time period is the same independently of the programming language used -> increasing the programming language capabilities the programmers throughput will increase.

Historical evolution

- Modern OSs (cont.)

Parallel systems
- Multiprocessor (Multicore):
  - Several processors/cores coupled by shared memory
- Reliability

Distributed systems
- The whole computation is distributed among several computers connected with a network
- Internode communication
- Resource sharing
- Workload sharing

First UNIX written in high level language (C)

Cloud systems
- Storage and computation as a service
- Based on virtualisation

Historical evolution

- UNIX and C origin

1968

- Software crisis
- First UNIX systems

1969

  - SO Written in high level language C

1988

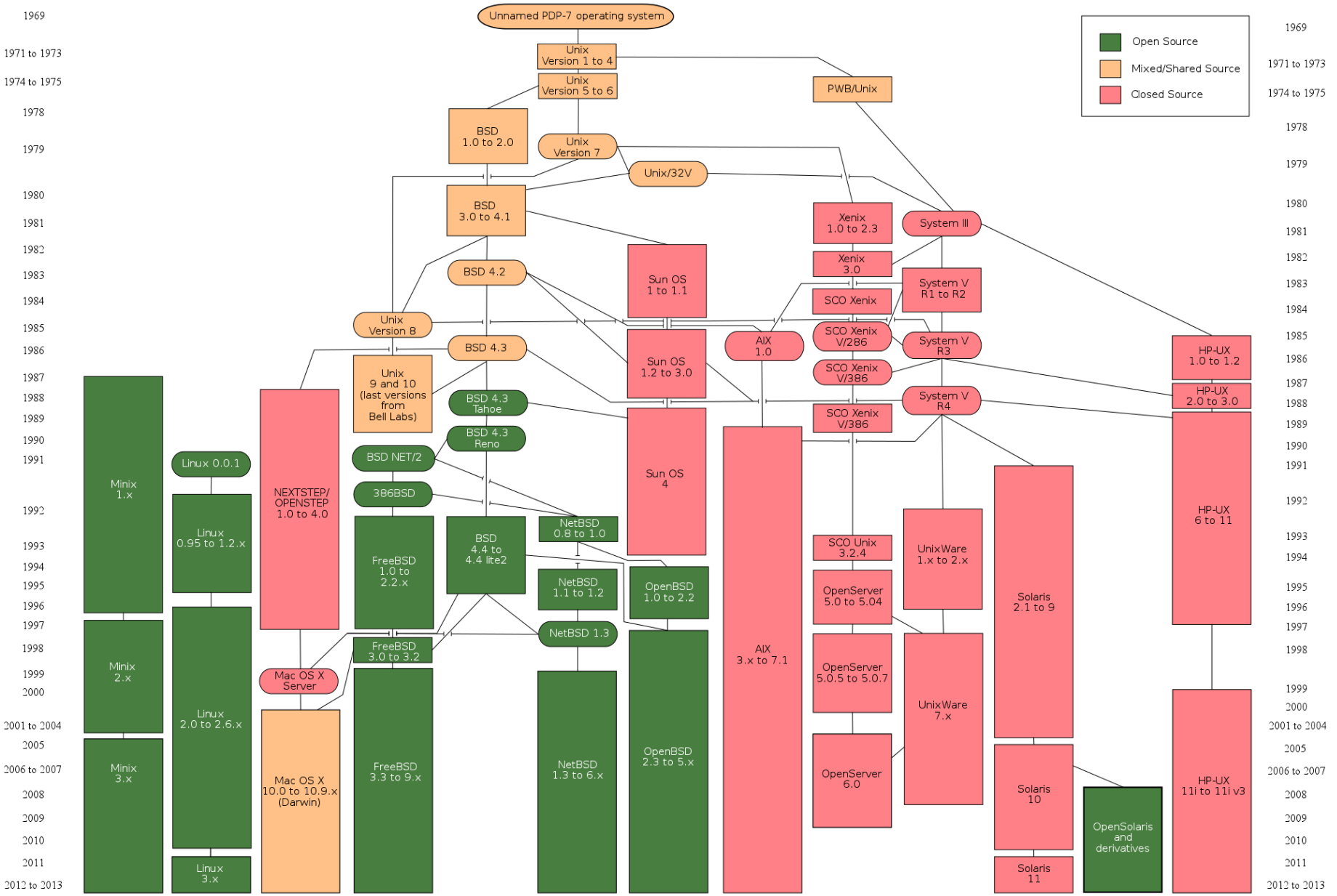
  - First version of POSIX standard IEEE 1003: Standardization of the system calls interface and other UNIX components. Interoperability at the source code level

Dennis Ritchie and Ken Thompson working on PDP-11 computers during the early development of Unix

1975

- Incorporation of virtual memory addressing in the PDP-11 processor
  - Digital Computer (DEC) VAX 11/780 with VAX11/VMS OS (VMS: Virtual Memory System)

UNIX chronicle



Historical evolution

- The birth of personal computer

1964 – ¿*Programma 101*

→ *programmable calculator*?

1972 – ¿*Xerox Alto* → *Alto Executive*

Workstation?

1975 – ¿*Altair 8800 Kit*?

CP/M, Altair BASIC, ..

1977 – *Apple II* → intérprete BASIC

¿DISER Lilith (*Workstation*)

→ Oberon?

1980 – 86-DOS /x86DOS /QDOS

1981 – IBM PC → PC-DOS / MS-DOS

1982 – Amstrad CPC → CP/M

ZX Spectrum → Sinclair BASIC

Commodore 64 → GEOS

1984 – Apple Macintosh → MacOS

Commodore Amiga 1000 → AmigaC

Amstrad PCW → CP/M Plus

1985 – Atari ST → Atari T

Fuente: imágenes de <http://www.wikipedia.com >

Exercise

- Try to find the version of the Linux kernel installed and active in your system, and the Linux distribution, using the shell commands `uname` and `lsb_release`.

  a) Find the kernel version running the command:

  ```
  $ uname -rs
  ```

  b) Find the SO name with the command:

  ```
  $ uname -o
  ```

  c) Find the processor architecture with the command:

  ```
  $ uname -m
  ```

  d) Find the SO distribution version with the command:

  ```
  $ lsb_release -i
  ```

WARNING The character `$` at the beginning of every command it the UNIX prompt **you don't have to write it**

WARNING You can use the commands `man uname` and `man lsb_release` to get help

Summary

- In this Unit, we have seen:
  - A global view of what is an Operating System
  - The role of OSes in the past, current and embedded systems
  - Examples of OSes and their impact on the market
  - Internal structure of an OS
  - Evolution and trends