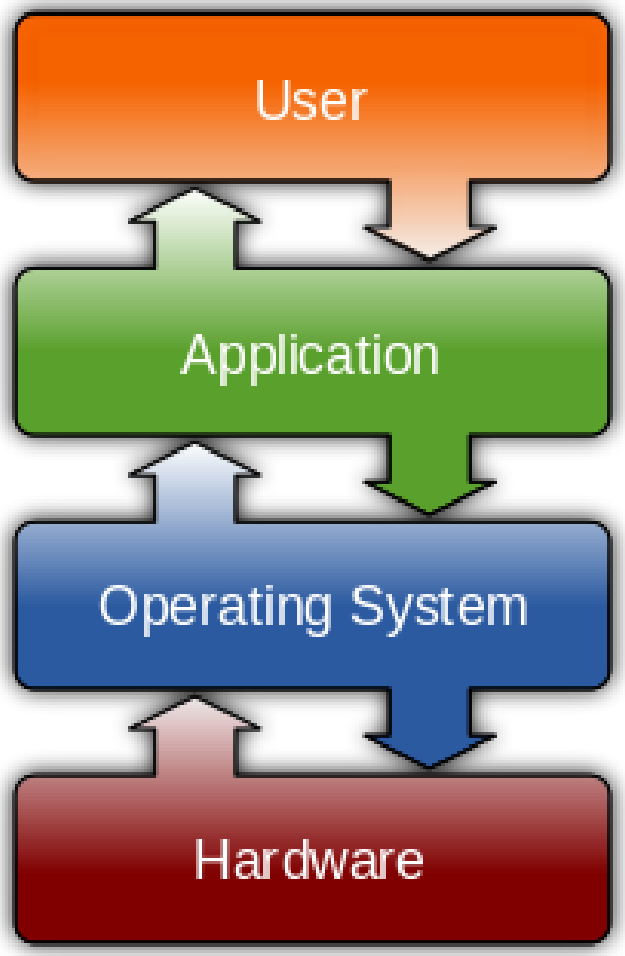
Operating Systems

Lecture 1

Introduction to Operating System and Computer System

Computer System Components

- ***Hardware:*** provides basic computing resources (CPU, memory, I/O devices)
- ***Operating system:*** controls and coordinates the use of the hardware among the various application programs for the various users
- ***Applications programs:*** define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs)
- ***Users:*** people, machines, other computers



Operating System

- Is a program that controls the execution of application programs
- Is an interface between the user and hardware
- Masks the details of the hardware to application programs, so the OS must deal with hardware details

Examples of OS

- Some popular Operating Systems include:
  - Linux-based OS
  - Microsoft Windows
  - macOS (used for Apple's computers and client models)
  - iOS (used for Apple's smartphone/tablet models)
  - Android

All of these operating systems have different generations, versions, and upgrades.

What is an Operating System?

- The general definition of any system is:

Some components integrated together for perform a desired task.

- So, the definition of computer operating system is

Some components integrated together for operating the computer system (HW &SW).

What is an Operating System?

- The general definition of any system is:

Some components integrated together for perform a desired task.

- So, the definition of computer operating system is

Some components integrated together for operating the computer system (HW &SW).

What is an Operating System?

- The general definition of any system is:

Some components integrated together for perform a desired task.

- So, the definition of computer operating system is

Some components integrated together for operating the computer system (HW &SW).

What is an Operating System?

The definition of the operating system is:

An operating system, commonly referred to as the OS, is a program that controls the execution of other programs running on the system.

Itacts as a facilitator and intermediate layer between the different software components and the computer hardware

- When any operating system is built, it focuses on three main objectives:
  - *Efficiency* of the OS in terms of responsiveness, fluidity, and so on
  - *Ease of usability* to the user in terms of making it convenient
  - *Ability to* abstract and *extend* to new *devices and software*

What Operating Systems Do?

User View

- Users want convenience, **ease of use** and **good performance.**
  - Don't care about **resource utilization.**
- But shared computer such as mainframe or minicomputer must keep all users happy.

What Operating Systems Do?

System View

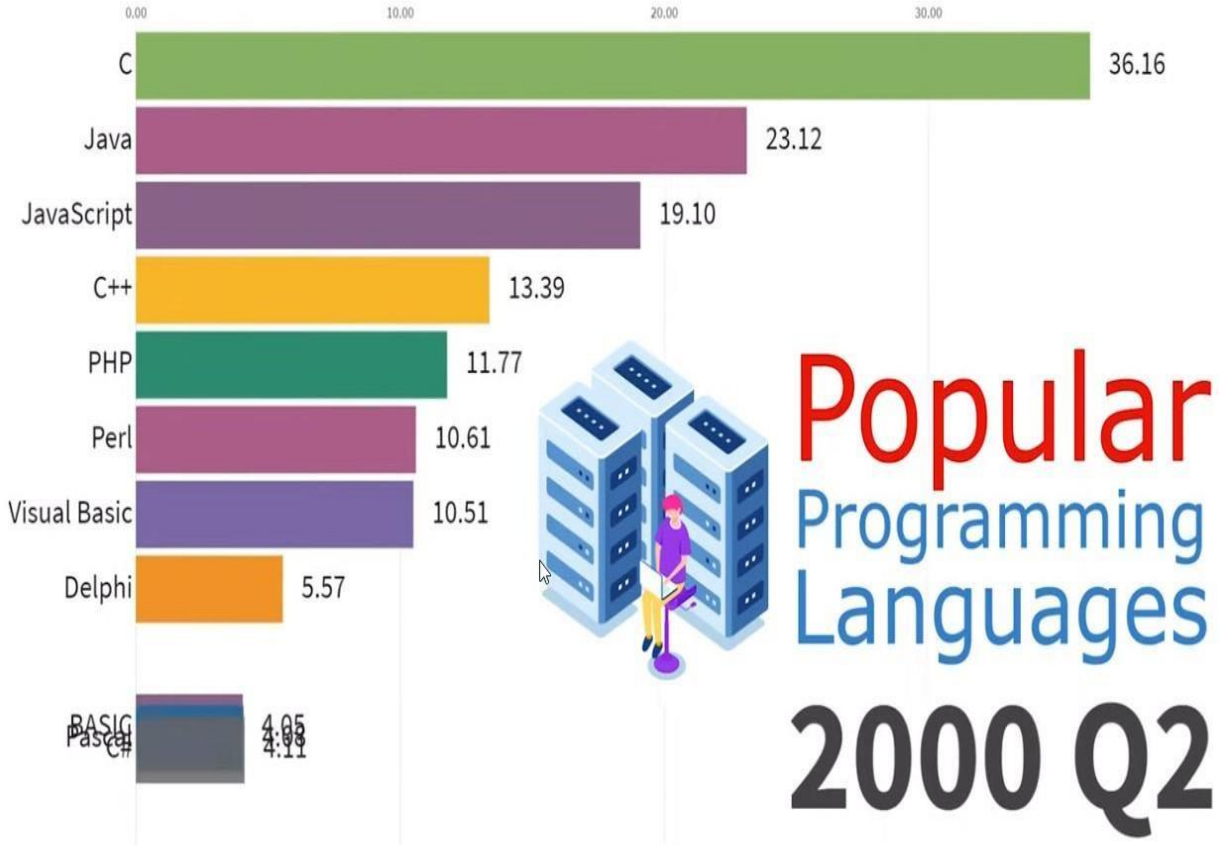
- From the computer's point of view, the operating system is the program most intimately involved with the hardware. In this context, we can view an operating system as a **resource allocator**.
- A computer system has many resources that may be required to solve a problem: CPU time, memory space, file-storage space, I/O devices, and so on. The operating system acts as the **manager** of these **resources**.

What Operating Systems Do?

- 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.

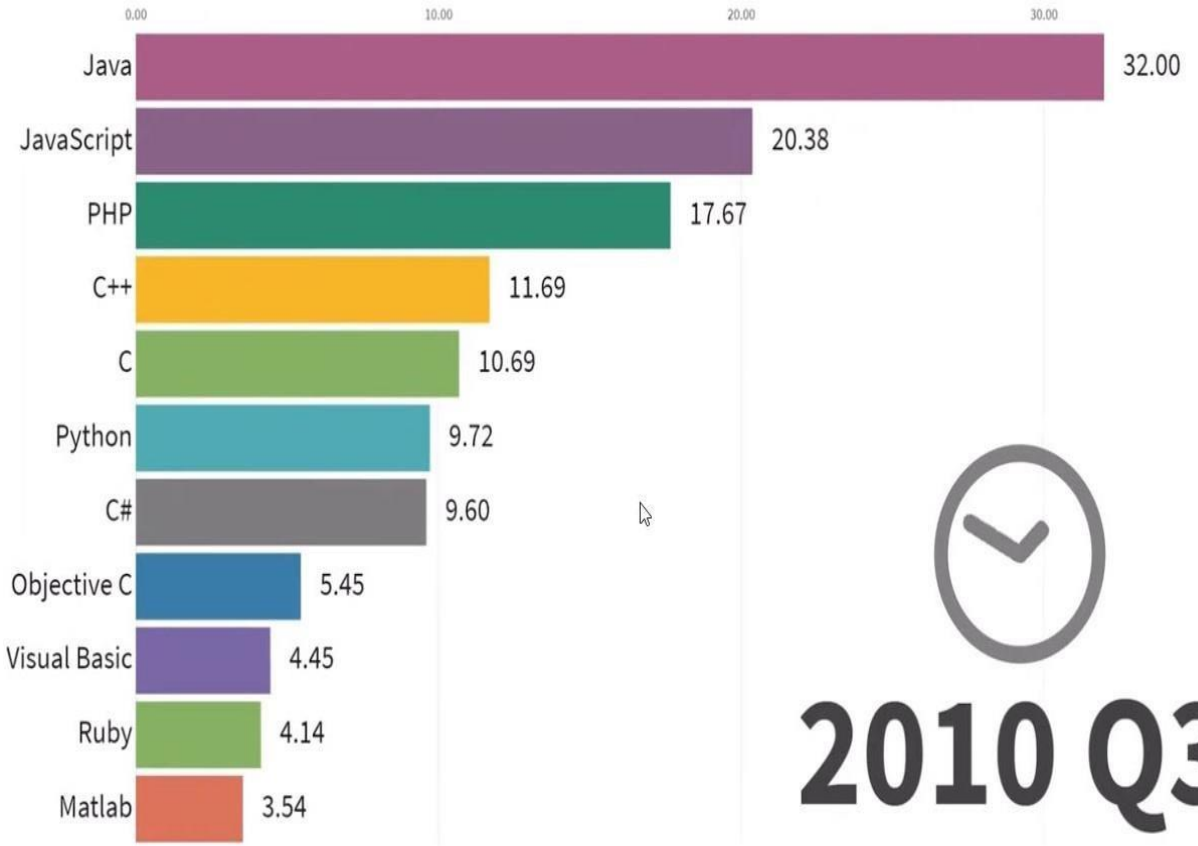
In which programming language the OS written ?

Most Popular Programming Languages



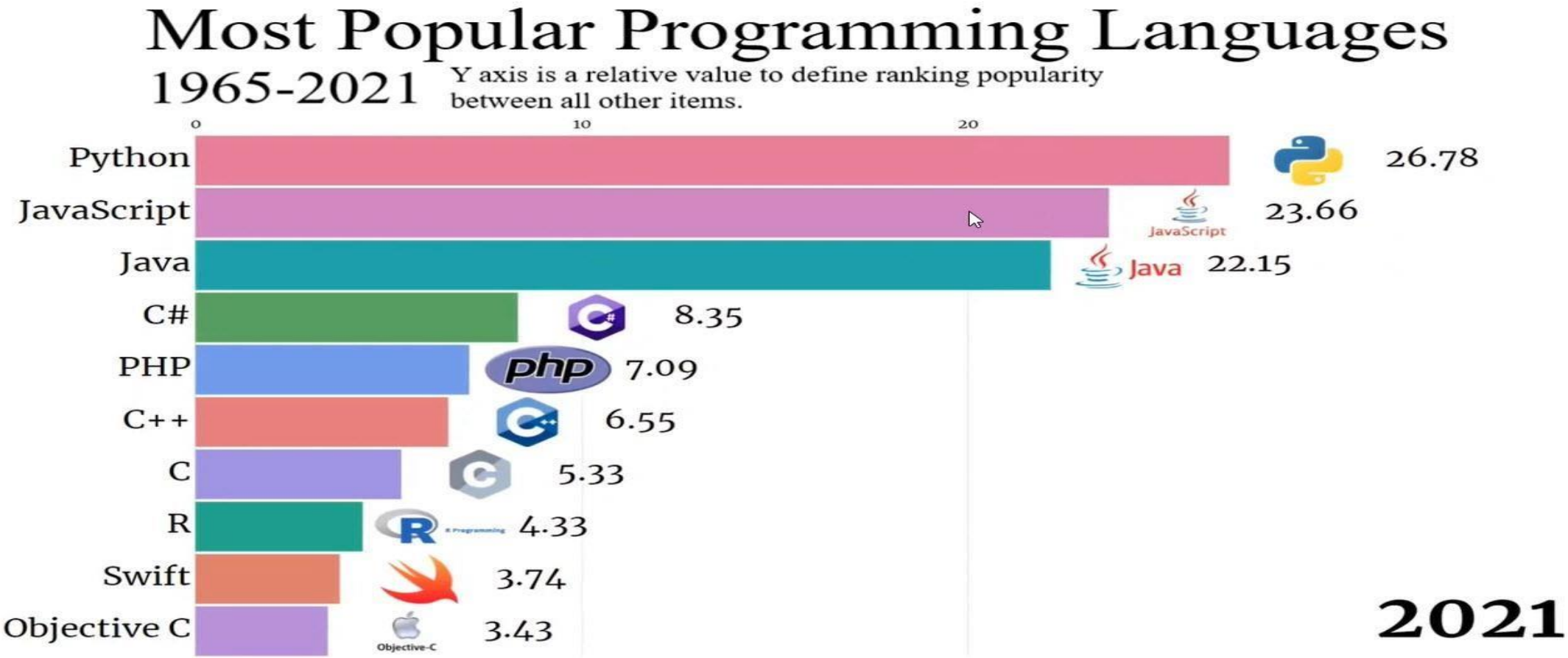
Popular Programming Languages

2000 Q2



2010 Q3

Most Popular Programming Languages



In which programming language the OS written ?

- The **UNIX** operating system's development started in 1969, and its code was **rewritten in C in** 1972. The C language was actually created to move the UNIX kernel code from assembly to a higher level language.

- In 1985 **Windows** 1.0 was released. Although Windows source code is not publicly available, it's been stated is **mostly written in C** with some parts in assembly.

- **Linux** kernel development started in 1991, and it is also **written in C**.

Ref: https://www.toptal.com/c/after-all-these-years-the-world-is-still-powered-by-c-programming

In which programming language the OS written ?

- Most of the operating systems are written in the C/C++ languages.

- These not only include Windows or Linux (the Linux kernel is almost entirely written in C), but also Google Chrome OS, RIM Blackberry OS 4.x, Apple Mac OS X, iPAD OS, Apple iPhone iPod Touch, and Cisco IOS (which is mainly comprised of compiled C and C++ code).

Computer Operating System

- **When a computer turns on**, the processor will execute the instructions that are presented to it; generally, the **first code** that runs is for the boot flow, called ***bootstrap***, which is a framework stored in **ROM** contains some **instructions** called basic input output instructions (**BIOS**).
- All these need to be abstracted and handled efficiently and seamlessly. The user expects the system to "just work." The ***operating system*** facilitates all of this and more.

Role of the Bootstrap

- Bootstrapping is the process of loading a set of instructions when a computer is first turned on or booted.

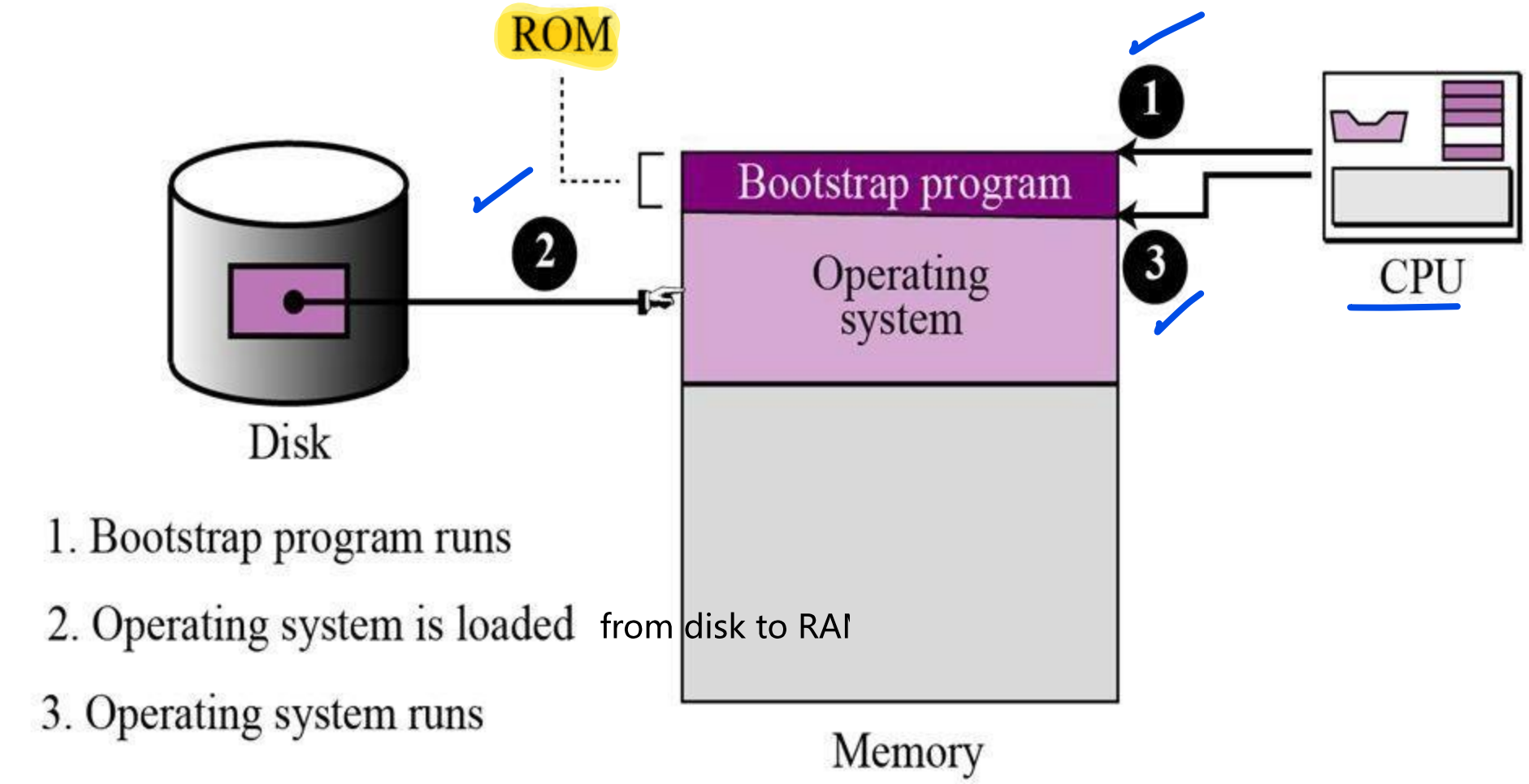
- The bootloader or bootstrap program is then loaded to initialize the OS.

bootstrap

- Basically for a computer to start running to get an instance when it is powered up or rebooted it need to have an initial program to run. And this initial program which is known as **bootstrap** needs to be simple. It must initialize all aspects of the system, from CPU registers to device controllers and the contents of the main memory, and then starts the operating system.
- To do this job the bootstrap program basically finds the operating system kernel on disk and then loads the kernel into memory and after this, it jumps to the initial address to begin the operating-system execution.

Ref: https://www.geeksforgeeks.org/boot-block-in-operating-system/

Bootstrap Process



Services Provided by the OS

- **Facilities for Program creation** البرامج لإنشاء مرافق
  - editors, compilers, linkers, and debuggers
- **Program execution**
  - loading in memory, I/O and file initialization
- **Access to I/O and files**
  - deals with the specifics of I/O and file formats
- **System access**
  - Protection in access to resources and data
  - Resolves conflicts for resource contention

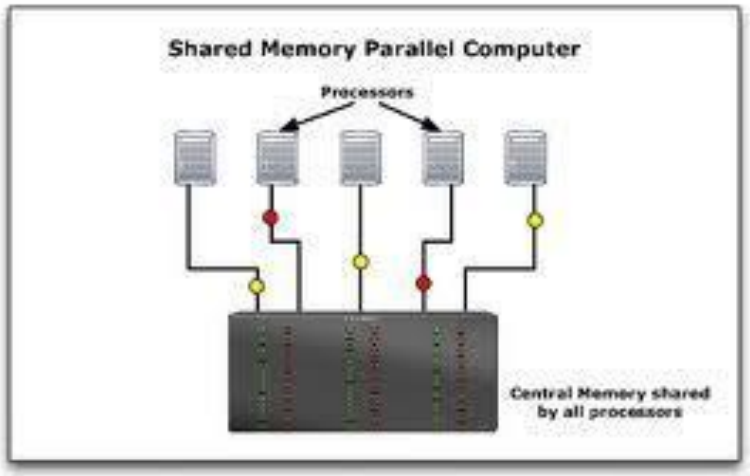
OS Categories – Usage Types

- Based on this, there are five main categories:
  1. ***Batch***: The batch operating system grouped jobs that perform similar functions. These job groups are processed as a batch and executed simultaneously. This operating system allows a system to execute the following batch processing tasks **(first come, first served)such as : (Old Mainframe and DOS)**
  2. ***Time Sharing***: For systems where many users (or many applications) access common hardware, there could be a need to timeshare the limited resources. The OSs in such cases are categorized as time-sharing OSs. (Windows, Mac, and Linux)
  3. ***Parallel (over tightly coupled systems)***: For hardware that is distributed physically and a single OS needs to coordinate their access, we call these systems distributed OSs. (AIX for IBM RS/6000 and Solaris for workstations)
  4. ***Network – Distributed (loosly coupled)***: Another usage model, similar to the distributed scenario, is when the systems are connected over a network protocol, like IP (Internet Protocol), and therefore referred to as network OSs. (Windows server 2008, Novell Netware)
  5. ***Real Time***: In some cases, we need fine-grained time precision in execution and responsiveness. We call these systems **real-time OSs.**

Ref: https://www.javatpoint.com/types-of-operating-systems

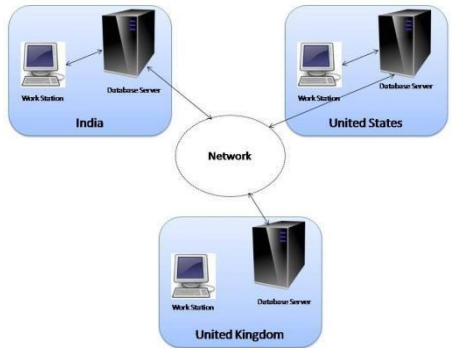
Parallel Systems

- Systems with more than one CPU in close communication
  - Also known as *multiprocessor systems*
- *Tightly coupled system*
  - processors share memory and a clock; communication usually takes place through the shared memory
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - graceful degradation
    - fail-soft systems



Distributed Systems

- Distribute the computation among several physical processors
- *Loosely coupled system*
  - Each processor has its own local memory
  - processors communicate with one another through various communications lines, such as high-speed buses or telephone lines
- Advantages of distributed systems
  - Resources Sharing
  - Computation speed up
    - load sharing
  - Reliability



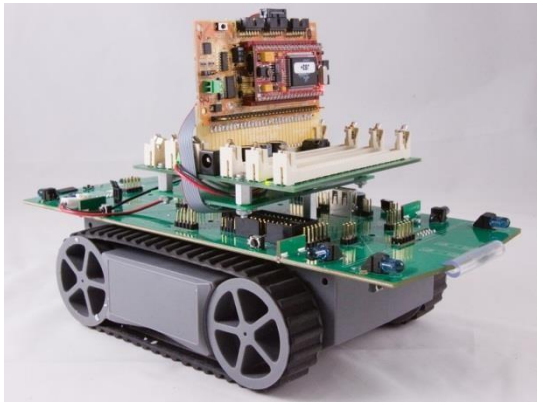
Loosely and Tightly Coupled Multiprocessor System (Comparison)

	LOOSELY COUPLED (Distributed systems)	TIGHTLY COUPLED (Parallel Systems)
memory	Each processor has its own memory module.	Processors have shared memory modules.
Efficient	when tasks running on different processors, **has minimal interaction**.	Efficient for **high-speed processing**.
Memory conflict	No memory conflict.	more memory conflicts.

Ref: https://techdifferences.com/difference-between-loosely-coupled-and-tightly-coupled-multiprocessor-system.html

Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems



- Well-defined fixed-time constraints

- Real-Time systems may be either *hard* or *soft* real-time

Real-Time Systems Cont'd

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems
- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring tight response times

Thank You