Laurea Specialistica in Ingegneria dell'Automazione

Sistemi in Tempo Reale

Giuseppe Lipari

Introduzione alla concorrenza

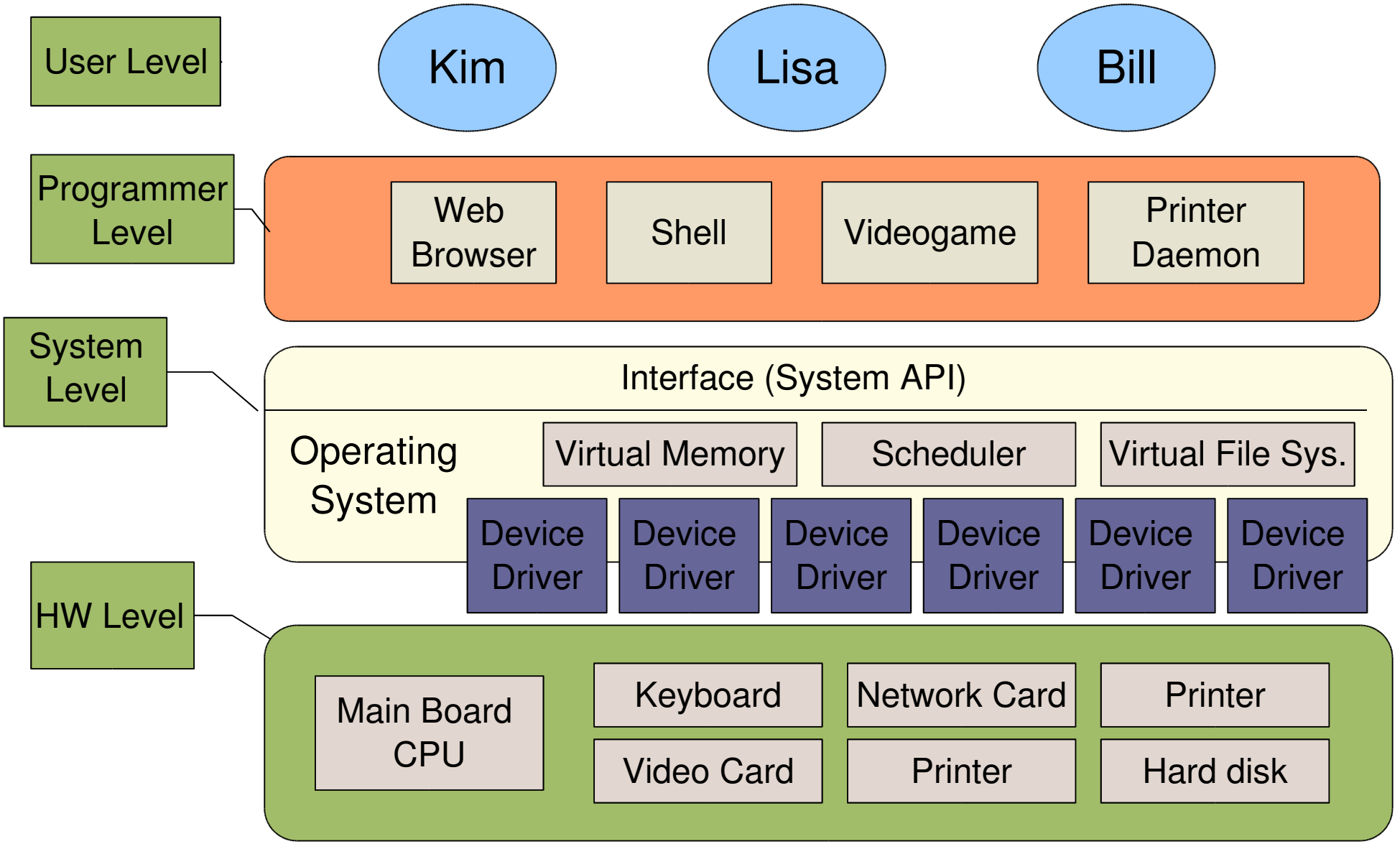
Fundamentals

- Algorithm:
  - It is the logical procedure to solve a certain problem
  - It is informally specified a a sequence of elementary *steps* that an “execution machine” must follow to solve the problem
  - It is not necessarily expressed in a formal programming language!
- Program:
  - It is the implementation of an algorithm in a programming language
  - Can be executed several times with dufferent inputs
- Process:
  - An instance of a program that given a sequence of inputs produces a set of outputs

Operating System

- An operating system is a program that
  - Provides an "*abstraction*" of the physical machine
  - Provides a simple interface to the machine
  - Each part of the interface is a "*service*"
- An OS is also a resource manager
  - With the term "resource" we denote all physical entities of a computing machine
  - The OS provides access to the physical resources
  - The OS provides *abstract resources* (for example, a file, a virtual page in memory, etc.)

Levels of abstraction



Abstraction mechanisms

- Why abstraction?
  - Programming the HW directly has several drawbacks
    - It is difficult and error-prone
    - It is not portable
  - Suppose you want to write a program that reads a text file from disk and outputs it on the screen
    - Without a proper interface it is virtually impossible!

Abstraction Mechanisms

- Application programming interface (API)
  - Provides a convenient and uniform way to access to one service so that
    - HW details are hidden to the high level programmer
    - One application does not depend on the HW
    - The programmer can concentrate on higher level tasks
  - Example
    - For reading a file, linux and many other unix OS provide the **open(), read()** system calls that, given a “file name” allow to load the data from an external support

Classification of Operating Systems

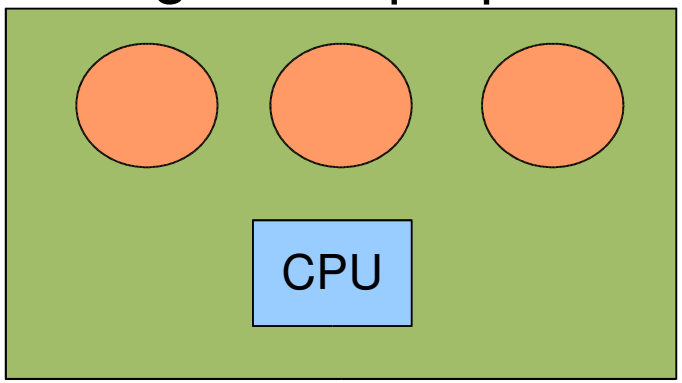
- The OS provides an abstraction of a physical machine
  - To allow portability
  - To make programmer's life easier
- The level of abstraction depends on the application context
  - It means that the kind of services an OS provides depend on which kind of services the application requires
    - General purpouses OS should provide a wide range of services to satisfy as many users as possible
    - Specialised OS provide only a group of specialised services
- OS can be classified depending on the application context

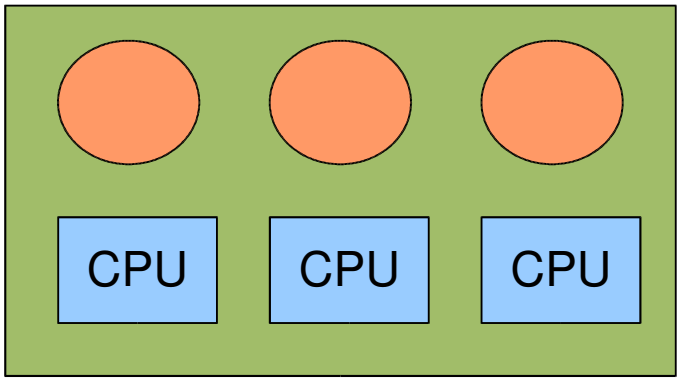
Classification of OS - II

- By objective
  - General purpouse (windows, linux, etc),
  - Servers,
  - RTOS
- By size
  - Full featured
  - Embedded OS
  - Nano-kernels
- By internal structure
  - Monolithic
  - Micro-kernel

Services

- Virtual processor
  - An OS provides “concurrency” between processes
    - Many processes are executed at the same time in the same system
    - Each process executes for a fraction of the processor bandwidth (as it were on a dedicated slower processor)
  - Provided by the scheduling sub-system
  - Provided by almost all OS, from nano-kernels to general-purpouse systems





Services

- Virtual memory
  - Physical memory is limited;
  - In old systems, the number of concurrent processes was limited by the amount of physical memory
  - IDEA: extend the physical memory by using a “fast” mass storage system (disk)
    - Some of the processes stay in memory, some are temporarily saved on the disk
    - When a process must be executed, if it is on the disk it is first loaded in memory and then executed
    - This technique is called “swapping”

Virtual Memory

- Advantages
  - Virtual infinite memory
  - The program is not limited by the size of the physical memory
- Disadvantages
  - If we have too many programs, we spend most of the time swapping back and forth
  - Needs support from the HW (MMU)
    - Small OS (like nano-kernels) do not support VM
  - This technique is NOT used in RTOS
    - It can introduce a large and unpredictable delay
    - It is not possible to guarantee a short response time because it depends on the number of processes, on the memory organization, etc.

Virtual File System

- Basic concepts
  - File: sequence of data bytes
    - It can be on a mass storage (hard disk, cd-rom, etc.)
    - It can be on special virtual devices (i.e. RAM disks)
    - It can be on a remote system!
  - Directory: list of files
    - Usually organised in a tree
    - Represents how files are organised on the mass storage system
- Virtualisation
  - In most OS, external serial devices (like the console or the video terminal) can be seen as files (i.e. stdin, stout , stderr)

Virtual file system

- A good virtual file system provides additional features:
  - Buffering & caching
    - For optimising I/O from block devices
  - Transactions
    - For example the Reiser FS
  - Fault tolerance capabilities
    - For example, the RAID system
- Virtual file system is not provided by all OS categories
  - Micro and nano kernels do not even provide a file system!