**Computer System Structure**

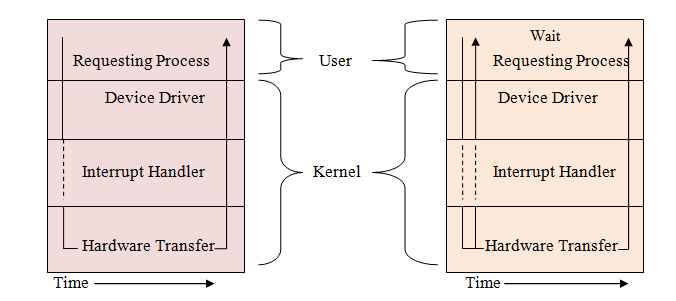
**Computer System Architecture**

* Single Processor System: One main CPU capable of executing a general-purpose instruction set, including instruction from user processes
* Multiprocessor Systems :also Known as parallel systems or coupled systems, which have two or more processors in close communication ,sharing the computer bus and sometimes the clock, memory and peripheral devices.
  + Two types of multiprocessor systems:
    - Asymmetric multiprocessing.(master-salve computing environment)
    - Symmetric multiprocessing. (peer-to-peer environment computing)

**I/O Structure**

Two methods for I/O:

* Synchronous
  + - In this case after I/O starts, control returns to user program only upon I/O completion.
    - Wait instruction idles the CPU until the next interrupt
    - Wait loop (contention for memory access)
* Asynchronous:
  + - After I/O starts, control returns to user program without waiting for I/O completion.

**Two I/O methods**

A) Synchronous B) Asynchronous

**Figure (1): Synchronous and Asynchronous I/O methods**

* + - System call – request to the operating system to allow user to wait for I/O completion.
    - Device – status table contains entry for each I/O device indicating its type, address and state
    - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt

**Operating System Structure**

1. **Simple Structure :**
   * MS-DOS :written to provide the most functionally in the least space
   * Not divided into modules
   * Its interface and levels of functionally are not well separated

**Figure (2): MS-DOS Layer Structure**

1. **Layered Structure:**

The operating system is divided into number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware, the highest (layer n) is the user interface.

-

-

-

**Figure (3): Layered Structure Operating System**

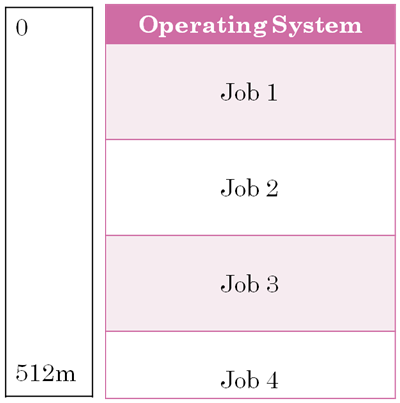
|  |  |
| --- | --- |
| Number layer | Content |
| Layer 5 | User program |
| Layer 4 | Buffering for input and output devices |
| Layer 3 | Operator-console device driver |
| Layer 2 | Memory management |
| Layer 1 | CPU scheduling |
| Layer 0 | Hardware |

**Table (1): the Layered Structure of operating system**

**Operating System Architecture**

* Multiprogramming:
  + Single user cannot keep CPU and I/O devices busy at all times.
  + Multiprogramming organize jobs (code and data) so CPU always has one to execute
  + One job selected and run via job scheduling
  + When it has to wait (for I/O for example), OS switch to another job
* **Ti**me sharing (multi tasking) in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing.
  + Response time should be <1 second
  + Each user has at least one program executing in memory process
  + If several jobs ready to run at the same time CPU scheduling
  + If processes don’t fit in memory, swapping moves them in and out to run
  + Virtual memory allows execution of processes not completely in memory

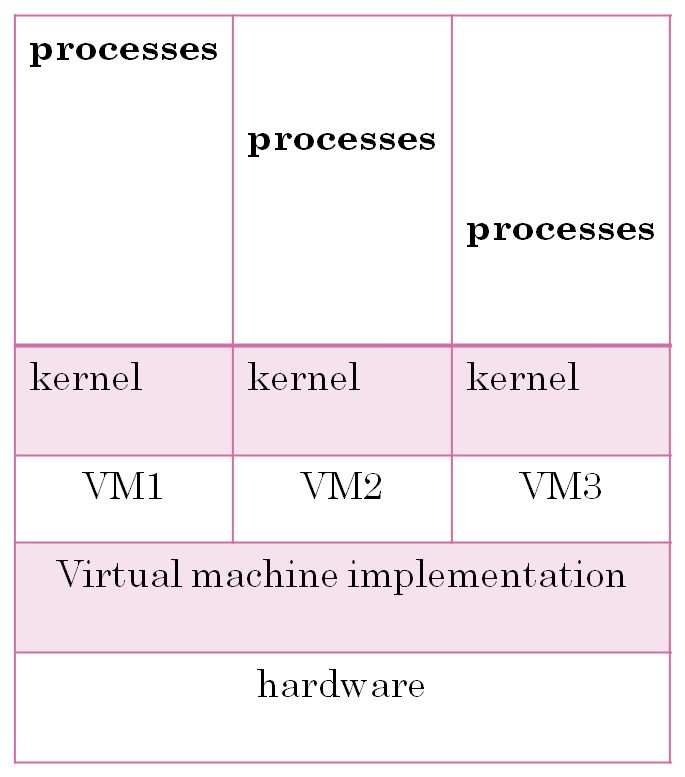
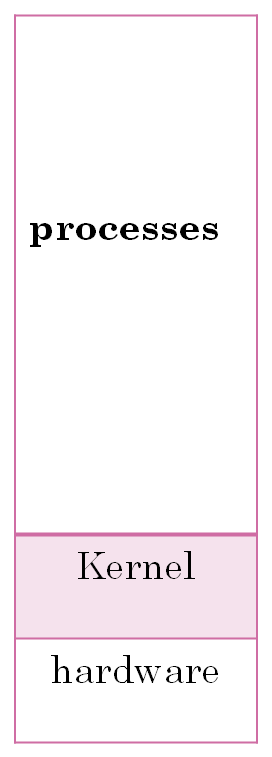
**Memory Layout for Multi-programmed System**

****

**Figure (4): Memory Layout for Multi-programmed System**

**Virtual Machines**

* **Virtual machine provides an interface identical to the underlying bare hardware.**
* **The operating system creates the illusion of multiple processes, each executing on its own processor with its own memory (virtually).**

****

**Non virtual machines virtual machines**

**Figure (5): Virtual Machines**