

# Operating System Concepts & Structure

## Operating System

- Manages computer hardware resources
- Provide common services for application software
- Acts as an intermediary between application programs and the computer hardware

## Operating System Concepts

- Processes
- Memory
- Files

## Processes

- Process – program in execution
- Associated with each process :
  - address space, a list of memory locations from some minimum (usually 0) to some maximum, which the process can read and write. Contains the executable program, the program's data, and its stack
  - some set of registers, including the program counter, stack pointer, and other hardware registers, and all the other information needed to run the program

## Processes

- all the information about each process, other than the contents of its own address space, is stored in an operating system table called the **process table**, which is an array (or linked list) of structures, one for each process currently in existence.
- Operating system is responsible for :
  - Creating and terminating/deleting process
  - Suspend and continue process
  - Interprocess communication

## Memory Management

- Main memory - To hold executing programs
- In a very simple operating system, only one program at a time is in memory. To run a second program, the first one has to be removed and the second one placed in memory.
- Now, OS allow multiple programs to be in memory at the same time. To keep them from interfering with one another (and with the operating system), some kind of protection mechanism is needed, which controlled by the operating system.
- OS responsible for managing and protecting the computer's main memory

## I/O Management

- Many kinds of input and output devices exist, including keyboards, monitors, printers, and so on.
- It is up to the operating system to manage these devices.

## Files

- a major function of the operating system is to hide the peculiarities of the disks and other I/O devices and present the programmer with a nice, clean abstract model of device-independent files.
- OS - create files, remove files, read files, and write files.
- Manage file pathname, directory, etc

## Security

- Computer contains the important informations
- operating system manages the system security so the files, for example, are only accessible to authorized users

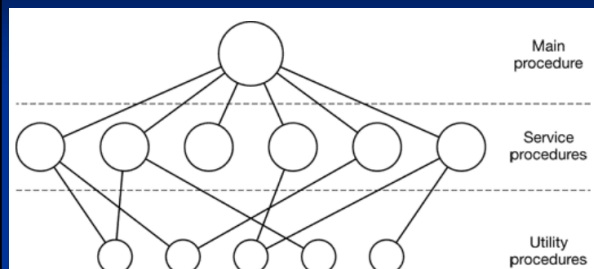
## Operating System Structure

- Monolithic systems
- Layered systems
- Virtual machines
- Exokernels
- Client-server model

## Monolithic Systems

- There is no structure
- The operating system is written as a collection of procedures, each of which can call any of the other ones whenever it needs to
  - A main program that invokes the requested service procedure.
  - A set of service procedures that carry out the system calls.
  - A set of utility procedures that help the service procedures.

## Monolithic Systems



## Layered Systems

- organize the operating system as a hierarchy of layers, each one constructed upon the one below it
- Example : THE (Technische Hogeschool at Eindhoven) operating system, had 6 layers

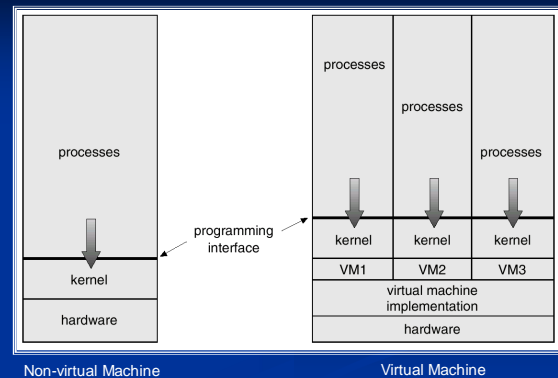
## Example : THE OS

- THE (Technische Hogeschool at Eindhoven) Operating System (Dijkstra, 1968)
  - Level 5 : operator / user
  - Level 4 : user program
  - Level 3 : I/O management
  - Level 2 : operator-process communication
  - Level 1 : memory management
  - Level 0 : processor allocation & multiprogramming

## Virtual Machines

- Exact copies of the bare hardware, including kernel/user mode, I/O, interrupts, and everything else the real machine has
- Because each virtual machine is identical to the true hardware, each one can run any operating system that will run directly on the bare hardware.
- Different virtual machines can, and frequently do, run different operating systems

## Virtual Machine

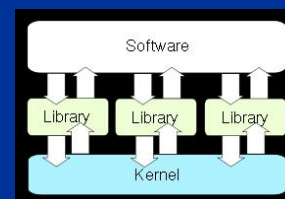


## Examples of VM Software

- VirtualBox
- VMWare
- VirtualPC

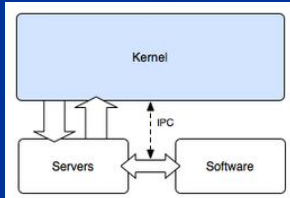
## Exokernels

- Minimal hardware abstraction
- Can access hardware directly
- Abstraction - Using library (libOS)
- Its job is to allocate resources and then check attempts to use them to make sure no machine is trying to use somebody else's resources



## Client-Server Model

- implement most of the operating system in user processes. To request a service, such as reading a block of a file, a user process (now known as the client process) sends the request to a server process, which then does the work and sends back the answer.



## Client-Server Model

- It's also called mikrokernel

