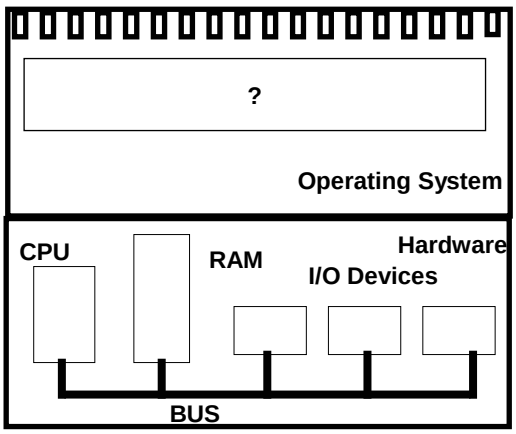
Operating System Structure

- Services
- System Components
- System Calls

- System Programs
- System Organization

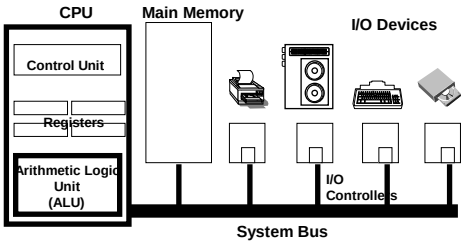


Operating System Services

- **Program Execution**
  - load into memory and run it.
  - Should start and stop gracefully.
- **I/O Operation**
- **File System Manipulation**
  - read, write, create, delete, search files and directories.
- **Communication**
  - exchange info between processes running in the same computer or in networks. It can be via shared memory of via message passing.
- **Error detection**
- **Resource allocation**
- **Accounting**
- **Protection**

Common System Components

- **Process Manager**
- **Main-Memory Manager**
- **Storage Manager**
- **I/O Manager**
- **File Manager**
- **Protection System**
- **Networking Manager**
- **Command Interpreter**



Process Manager

- **A Process is a program in execution. A process needs certain resources, including CPU time, memory, files and I/O devices to accomplish its task.**

- **The Manager is responsible for:**
  - ♦ **process creating and deletion.**
  - ♦ **Process suspension and resumption.**
  - ♦ **Provision of mechanisms for:**
    - » **process synchronization**
    - » **process communication**

Main Memory Management

- **Main memory is the repository of quickly accessible data shared by the CPU and I/O devices.**
- **It is volatile, not too large, not too slow.**

- **The Manager is responsible for:**
  - ♦ **Keep track of used and unused memory segments.**
  - ♦ **Keep track which process is using which segment.**
  - ♦ **Decide which processes to load when memory is available.**
  - ♦ **Allocate and deallocate memory space.**

Secondary-Storage Management

- **Secondary-Storage is used to back up the volatile main memory**
- **Manager is responsible for:**
  - ♦ **Free-space management**
  - ♦ **Storage allocation**
  - ♦ **Disk scheduling**

I/O System Manager

- **Manager is Responsible for:**
  - ♦ **buffer-cacheing**
  - ♦ **device driver interface**
  - ♦ **drivers for specific hardwaress**

File System Manager

- **A file is a collection of related information defined by user. A file can represent a program or the data used by a program.**

- **Manager is Responsible for:**
  - **File creation and deletion**
  - **Directory creation and deletion**
  - **Support of elementary operations for file manipulation**
  - **Mapping of files onto physical storage (secondery storage)**
  - **File backup on stable storage media.**

Protection System

- **Protection refers to a mechanism for controlling access by programs, processes, or users to both system and user resources.**

- **Protection System is responsible for:**
  - **distinguish between authorised and unauthorised useage.**
  - **Specify the control to be imposed**
  - **provide a means for enforcement.**

Networking Manager

- **A distributed system is a collection of processors that do not share memory or a clock. Each processor has its own local memory.**
- **The processors in the system are connected by communication network.**

- **Manager is Responsible for:**
  - **Access of one processors into another processors' resources**
  - **Communication between the OS components of the two systems.**
  - **Reliability and speed of communication.**

Command Interpreter

- **Many commands are given to the OS by control statements. Also known as:**
  - **control-card interpreter**
  - **command line interpreter**
  - **shell (unix)**

- **Interpreter is Responsible for:**
  - **getting and executing the command**

System Calls

Interface between a Process and Operating System

- **Process Control**
  - end, abort
  - load, execute
  - create, terminate process
  - get, set process attributes
  - wait for time, event, signal
  - allocate, free memory
- **File manipulation**
  - create, delete
  - open, close
  - read, write, reposition
  - get, set file attributes
- **Device manipulation**
  - request, relese device
  - read, write, reposition
  - get, set device attributes
  - attach, detach devices
- **Information Maintenance**
  - get, set time, date
- **Communication**
  - create, delete connection
  - send, receive messages
  - transfer status info
  - attach, detach remote devices

Data x-far mechanisms: Register, table+pointer OR stack

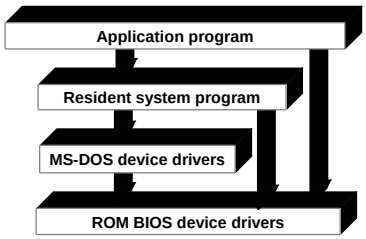
System Programs

Interface between a User and Operating System

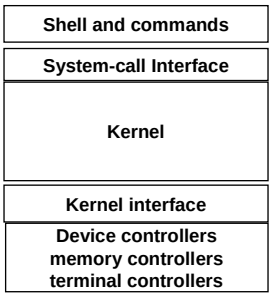
- **Program Loading and Execution**
  - run a program
  - halt or kill a process
- **File manipulation**
  - list a directory
  - create delete a file
- **File Modification**
  - edit a text file.
  - Edit file dates
- **Device manipulation**
  - request, relese device
  - read, write, reposition
  - get, set device attributes
  - attach, detach devices
- **Information Maintenance**
  - get, set time, date
  - list all running process
- **Communication**
  - telnet, ftp
  - ping

OS Software Structures

- **MS-DOS:**
  - ♦ **monolithic**



- **UNIX: two layered:**
  - ♦ **system programs**
  - ♦ **kernel**

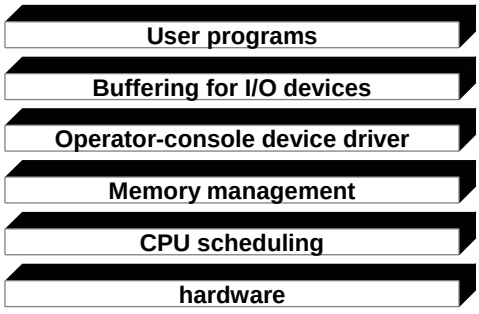


OS Software Structures

- **THE/ Venus: multi layered OS**

•Disk space device driver and memory manager-- who should be below?

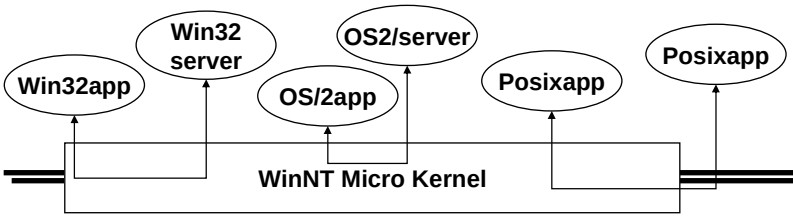
•Backing store driver or CPU scheduler-- who should be below?



Careful considerations needed in Layered design:
•Device driver for disk space management must be below memory management.
•Backing-store driver should be above CPU scheduler. But, what if the CPU scheduler may have more information about all the processes which may need backing up.

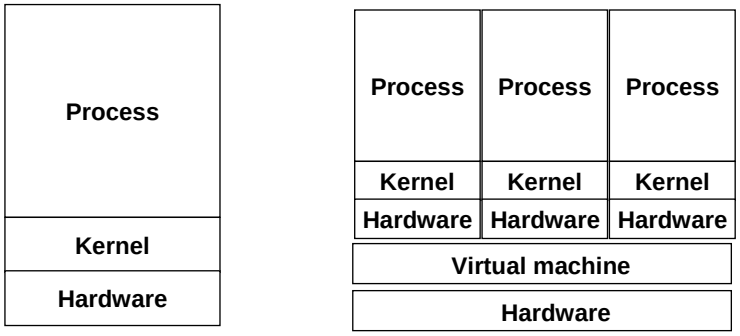
Microkernels

- **Developed in CMU in Mach (1985's)**
- **Remove all non essential components from kernel.**
- **Kernel Provides minimum:**
  - **process management.**
  - **Memory management.**
  - **Communication via process to process messaging.**
- **The main objective is to provide communication between client program and various services running in user space.**



Os-slide#15

Virtual Machine: JAVA



Os-slide#16

Advantages/Disadvantages of VM

- **Complete protection due to complete isolation**
- **Ideal for OS research/development**
- **CPU scheduling can create illusion of dedicated machine.**
- **Difficult to implement**
- **Can be slow**

OS Design Goals

- **OS should be convenient to use, easy to learn, reliable, safe and fast**
- **OS should be easy to design, implement, and maintain**