

## ★ Structure of OS

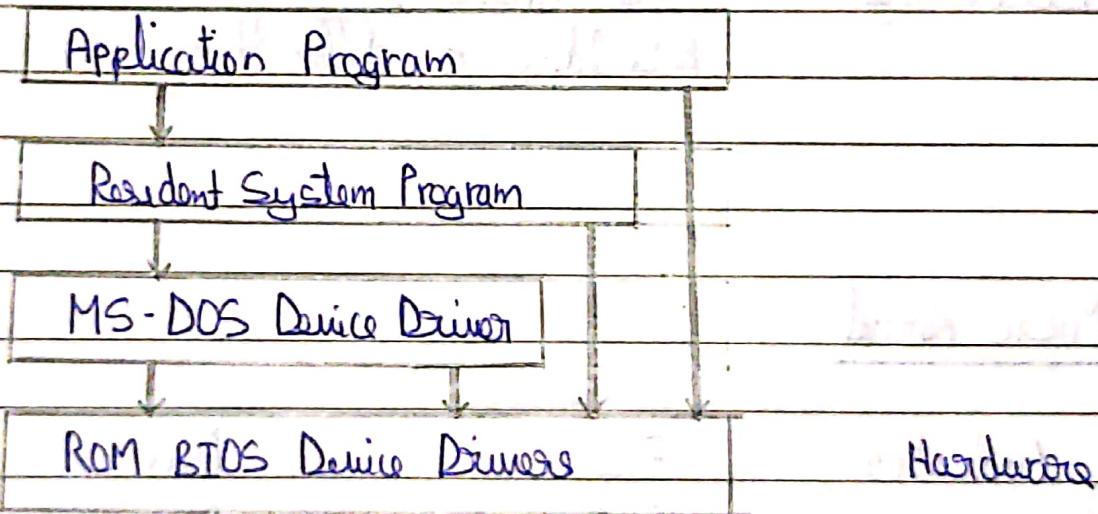
Simple Structure

Layred Structure

Micro Kernel

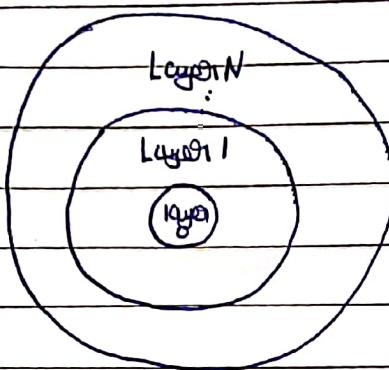
Modular Kernel

### Simple Structure



For eg.: MS-DOS

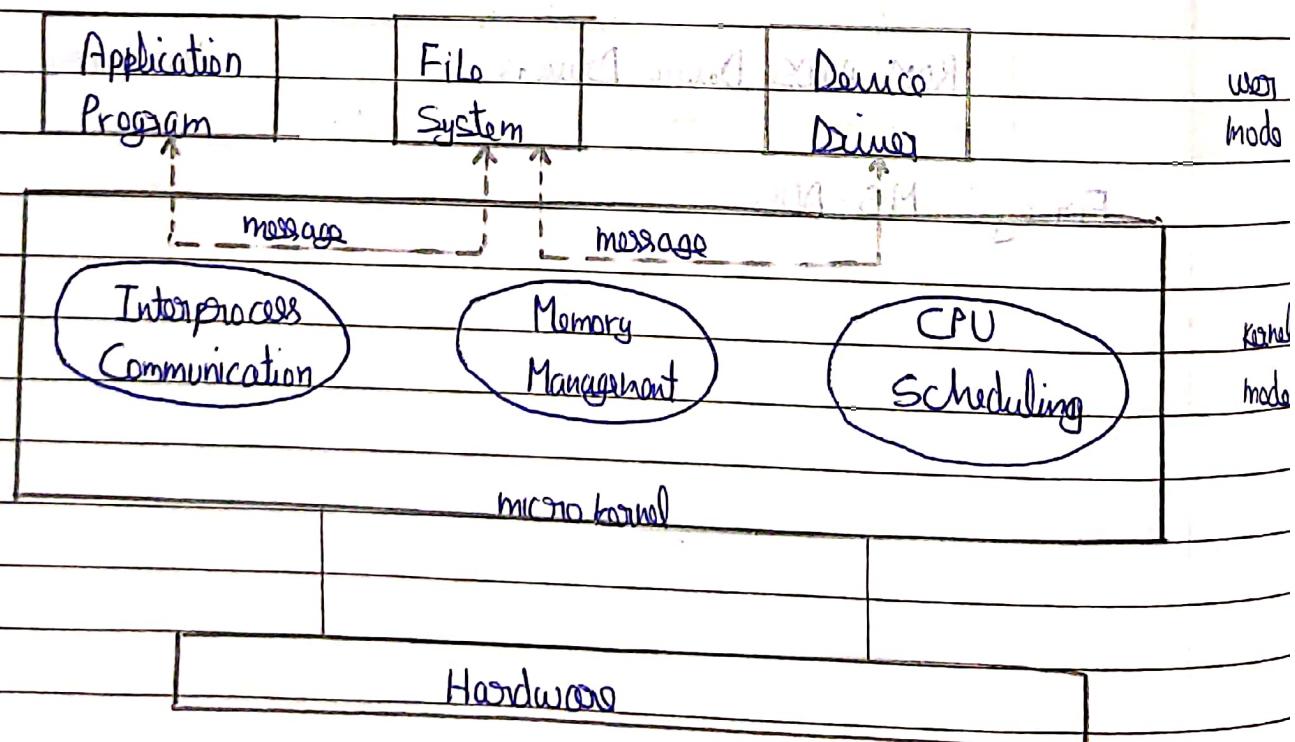
## Layered Structure



Disadvantage : Time consuming

Rebuilding is difficult

## Micro Kernel



## Modular Structure

Device & Bus  
Drivers

File System

miscellaneous  
modules

Kernel

Stream  
modules

executable  
Format

Advantage

Modularity

Disadvantage

Rebuilding is easier



## Operating System Services & System Calls

### User & Other Programs

GUI	batch	CommandLine
-----	-------	-------------

User Interface

Two types of modes are

User-mode - This mode is safer, any problem in this mode do not affect the system.

Kernel-mode - we use kernel mode for using the hardware. If any problem occurs in kernel mode, it affects the entire system.

## ► System Call

- It act as an interface between user & OS
- System calls are similar to function calls, they accept and works on arguments. They are executed in the kernel

### • Sequence of System Call

- Copying from source file to destination file involves a sequence of system call.

#### Steps

1. Source file should be there.

2. Create destination file.

3. read source file

4. copy to destination file

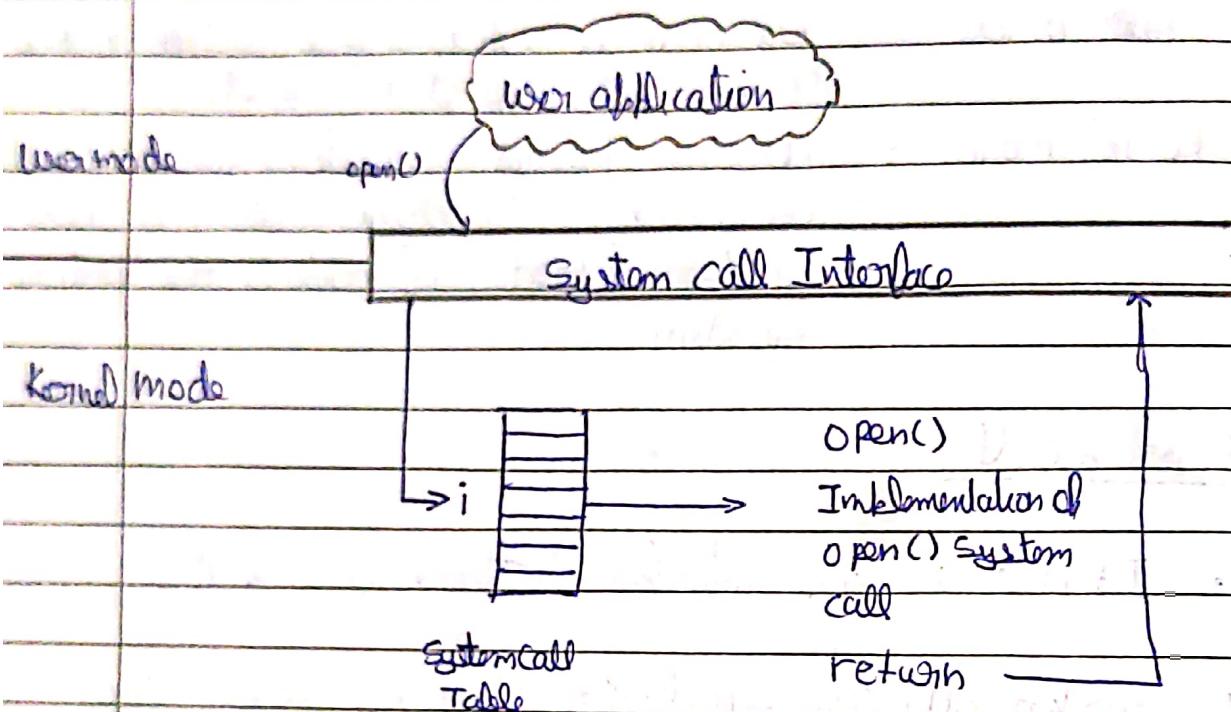
5. close source file

6. close destination file

7. close operation

: if any of the condition fails about the System call.

- System call Interface



► Types of System Calls

1. Process Control

- create process & terminate process
- abort
- load & execute
- get process attributes, set process attributes
- wait for time
- wait event, signal event
- allocate and free memory

2. File Management

- create file & delete file
- open & close file
- read, write & traversal
- get & set file attributes

### 3. Device Management

- request device and release device
- read, write and subscription
- get & set device attributes
- logically attach & detach devices

### 4. Information Maintenance

- get time or date & set time or date
- get system data, set system data
- get & set process
- get & set file or device attributes

### 5. Communication

- create & delete communication connections.
- Processes should communicate with each other for accessing shared resource

### 6. Protection

- control access to resources.
- get & set permissions
- allow & deny user access

Windows

Unix

Process  
Control

CreateProcess()

ExitProcess()

WaitForSingleObject()

fork()  
exit()

```
#include <stdio.h>
int main () {
```

```
    printf("Hello");
}
```

```
    return 0;
```

{}

Standard C Library

write()

User mode

Kernel mode

Write System  
call

## Important System Calls Used in OS

### 1) wait()

It helps a process wait for another process to complete its execution.

### 2) fork()

Process uses this system call to create process that are a copy of themselves.

### 3) exec()

This system call runs when an executable file in the context of an already running process executable file.

I/O Time : Process use peripheral time

CPU Time : Process use the CPU

## 8 Types of OS

- 1) Batch OS
- 2) Multiprogramming / Multi tasking OS
- 3) Multi Processing
- 4) Real Time

### ► Job

A process is called Job. The OS takes a job and provides it CPU time and I/O time.

### Batch OS

- In Batch OS the inputs are provided as batch. Batch of jobs are used here.
- Different Job batches are written using different languages like Cobol, Fortran etc.
- The OS takes Jobs one by one and execute it to provide the output.
- Drawback :
  - Less speed.
  - The CPU remains idle sometimes, even if other processes are waiting for it.

## ► Multiprogramming

- For every job some time slot is provided, the OS suspend the job if it does not complete execution and allocates the CPU time to other process.

## ► Multiprocessing

- More than 1 CPU is available for this
- Each CPU implements Batch OS or Multiprogramming

## ► Realtime OS

## System boot process

- The procedure of starting a computer by loading the kernel is known as booting the system.

OS is present in the ROM, OS need to be loaded into the main memory when computer is switched ON.

**BIOS - Basic Input Output System**

This is the first program that is loaded first into the main memory when the computer turns ON.

When current pass through BIOS chip, it loads the bios software into the main memory.

Power ON self test is done to check whether the peripheral devices are working.

**Bootstrap**

On most computer system a small piece of code known as the bootstrap program that creates a memory location for OS in the main memory.

## ▶ fork() - System call

Whenever there is a fork() system call, it creates a child process (copy of a process) with different Process Id.

$$\text{No. of processes} = 2^n$$

$n = \text{no. of fork() called}$

## ▶ getpid() System call

This gives us the Id of a process.

## Exc() - System Call

It's a function which is used to change the  
current running program environment.

It's a system call. It's a command.

It's a function.

Find the output.

main() {

int a = 10;

if (fork() == 0) {

a++;

printf("%d\n", a);

}

3

10 & 11

11