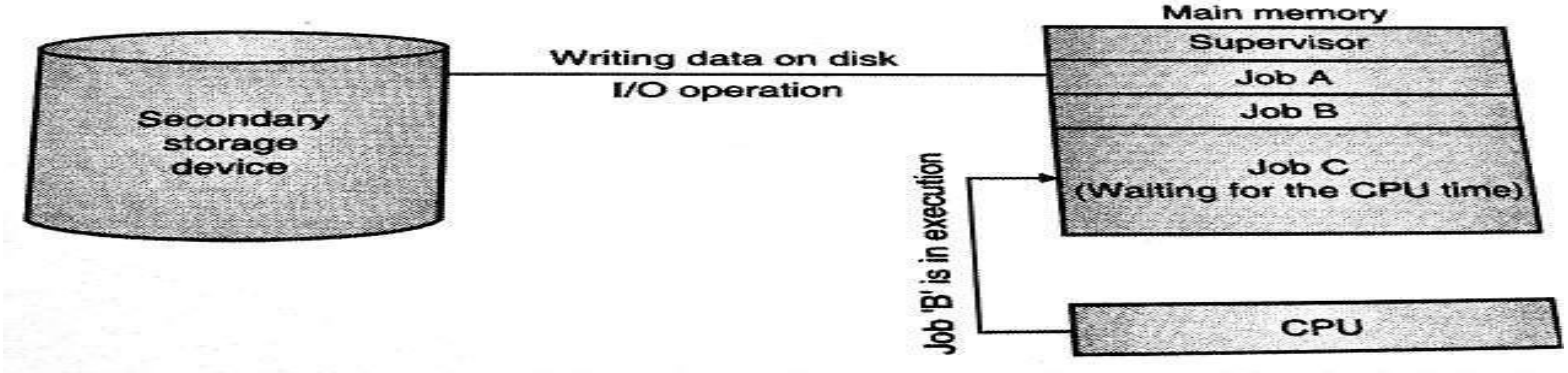
Fundamentals of Operating System

Ms. Minakshi Retharekar

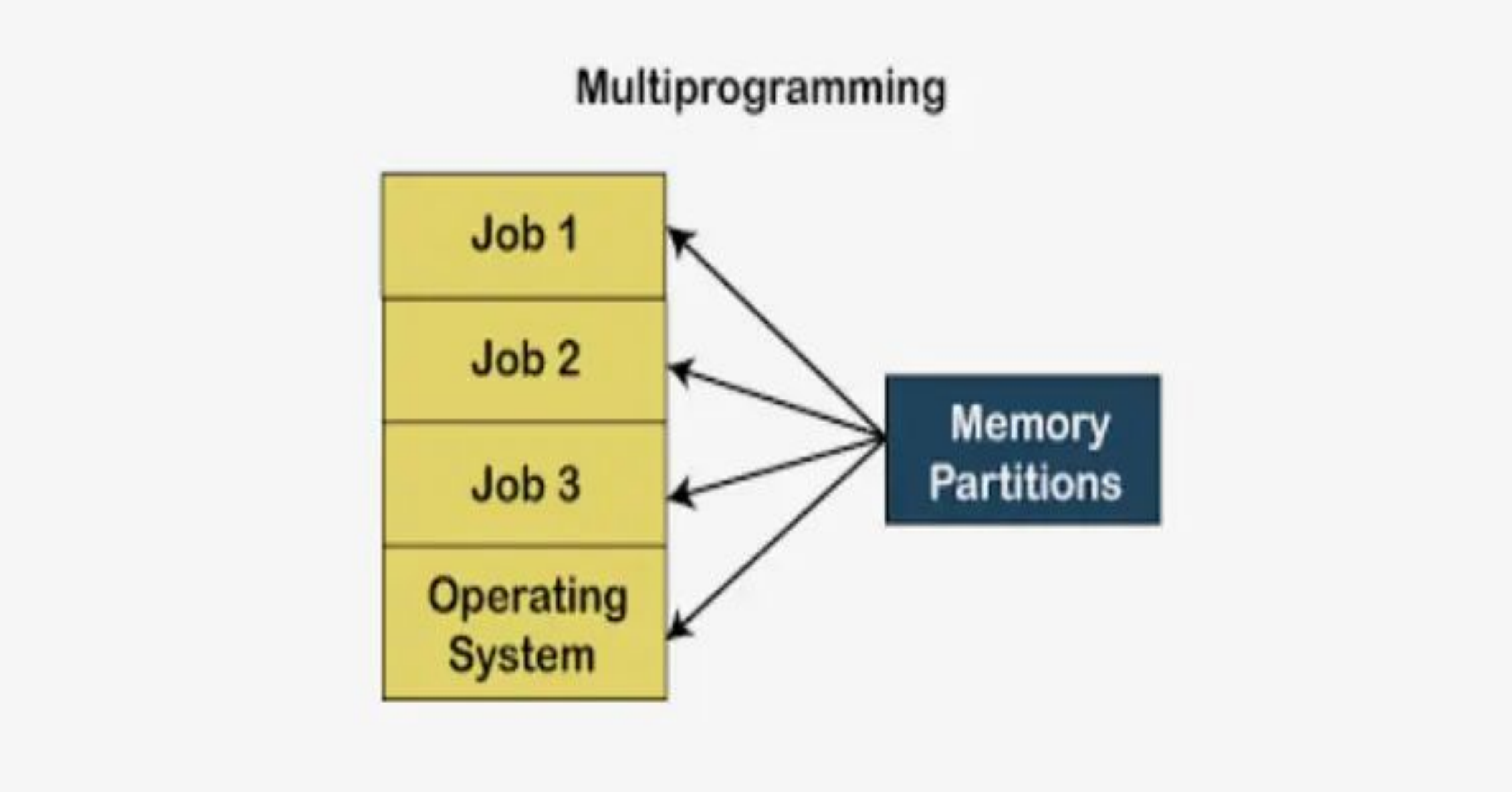
Multiprogramming OS

- This type of OS is used to execute more than one jobs **simultaneously** by a single processor.
- It increases CPU utilization by organizing jobs so that the CPU always has one job to execute.
- Multiprogramming operating systems use the mechanism of job scheduling and CPU scheduling.



Contd..

- Single user cannot keep CPU and I/O devices busy at all times
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via **job scheduling**
- When it has to wait (for I/O for example), OS switches to another job


Multiprogramming
Job 1
Job 2
Job 3
Operating System
Memory Partitions

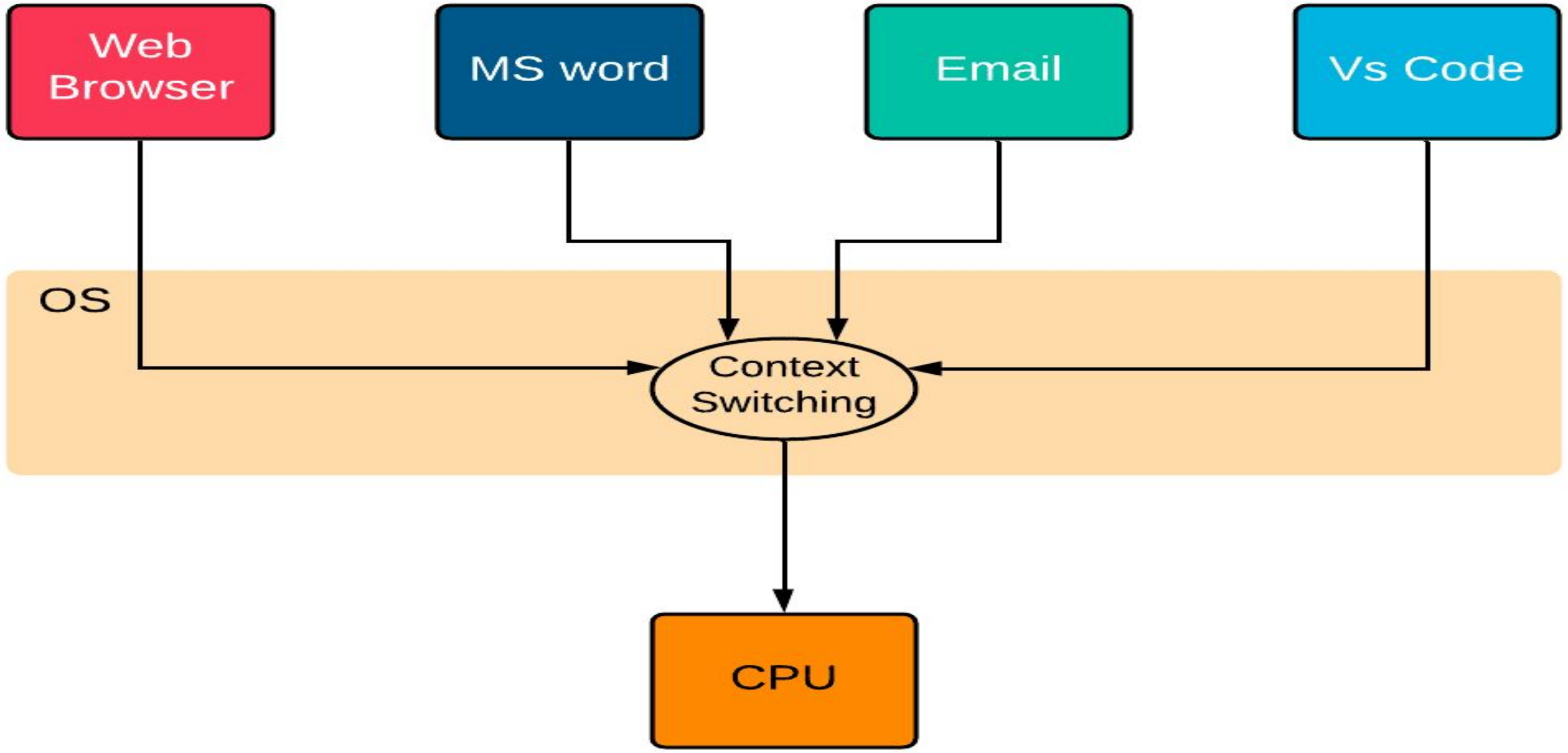
Time-Sharing Operating Systems

- Each task is given some time to execute so that all the tasks work smoothly.
- These systems are also known as **Multi-tasking Systems.**
- The task can be from a single user or different users also.
- The time that each task gets to execute is called quantum.
- After this time interval is over OS switches over to the next task.

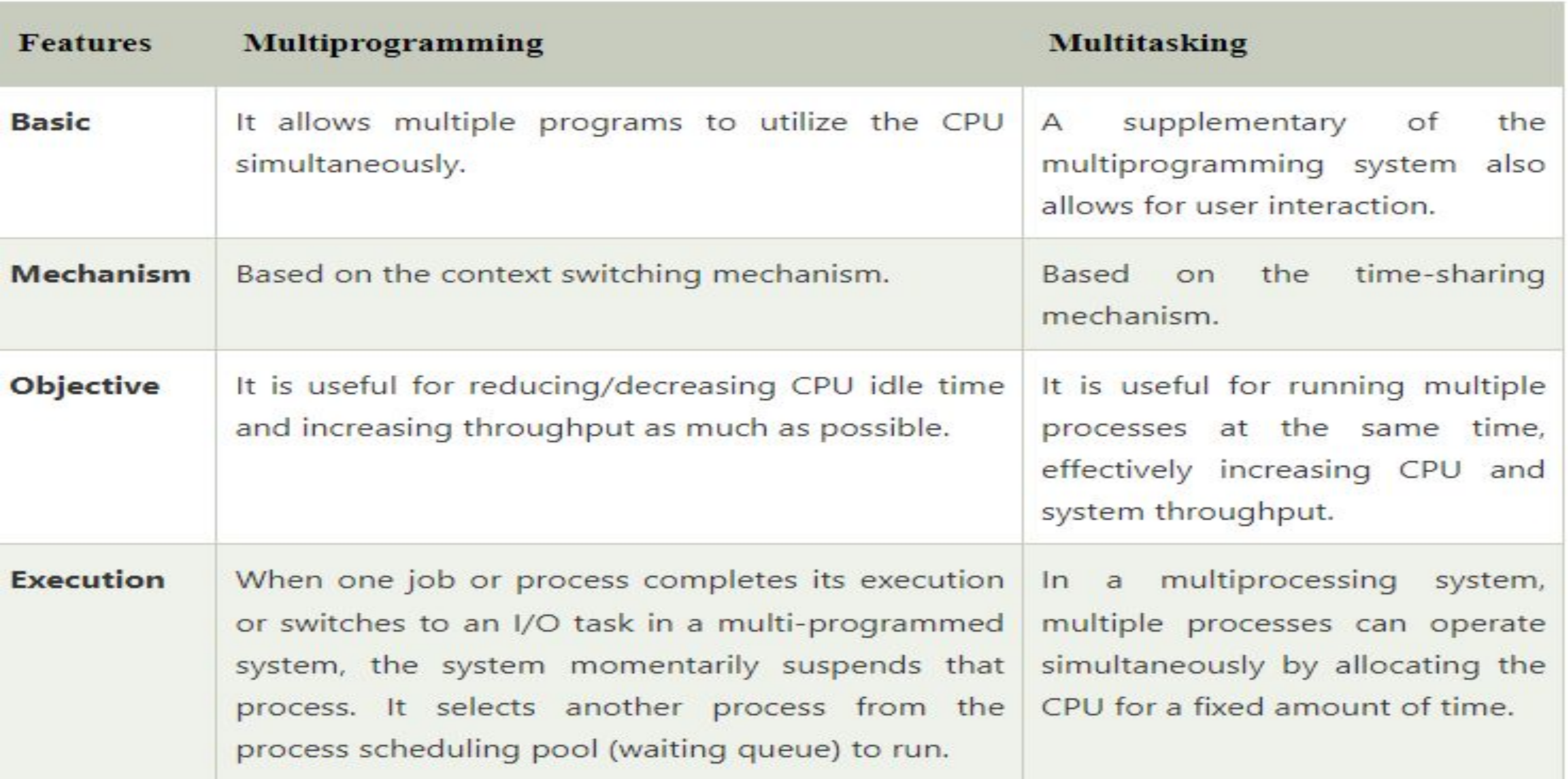
Time-Sharing Operating Systems cont

- **Advantages of Time-Sharing OS:**
  - Each task gets an equal opportunity
  - Fewer chances of duplication of software
  - CPU idle time can be reduced
- **Disadvantages of Time-Sharing OS:**
  - Reliability problem
  - One must have to take care of the security and integrity of user programs and data
  - Data communication problem
- **Examples of Time-Sharing OS**

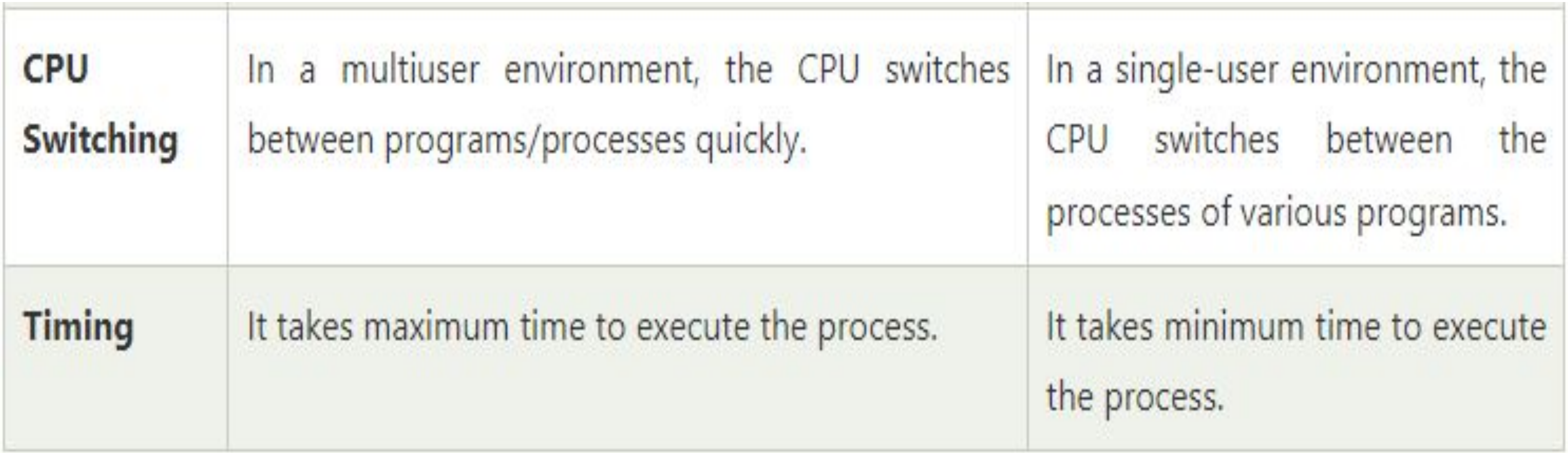
  Unix



Mulitasking

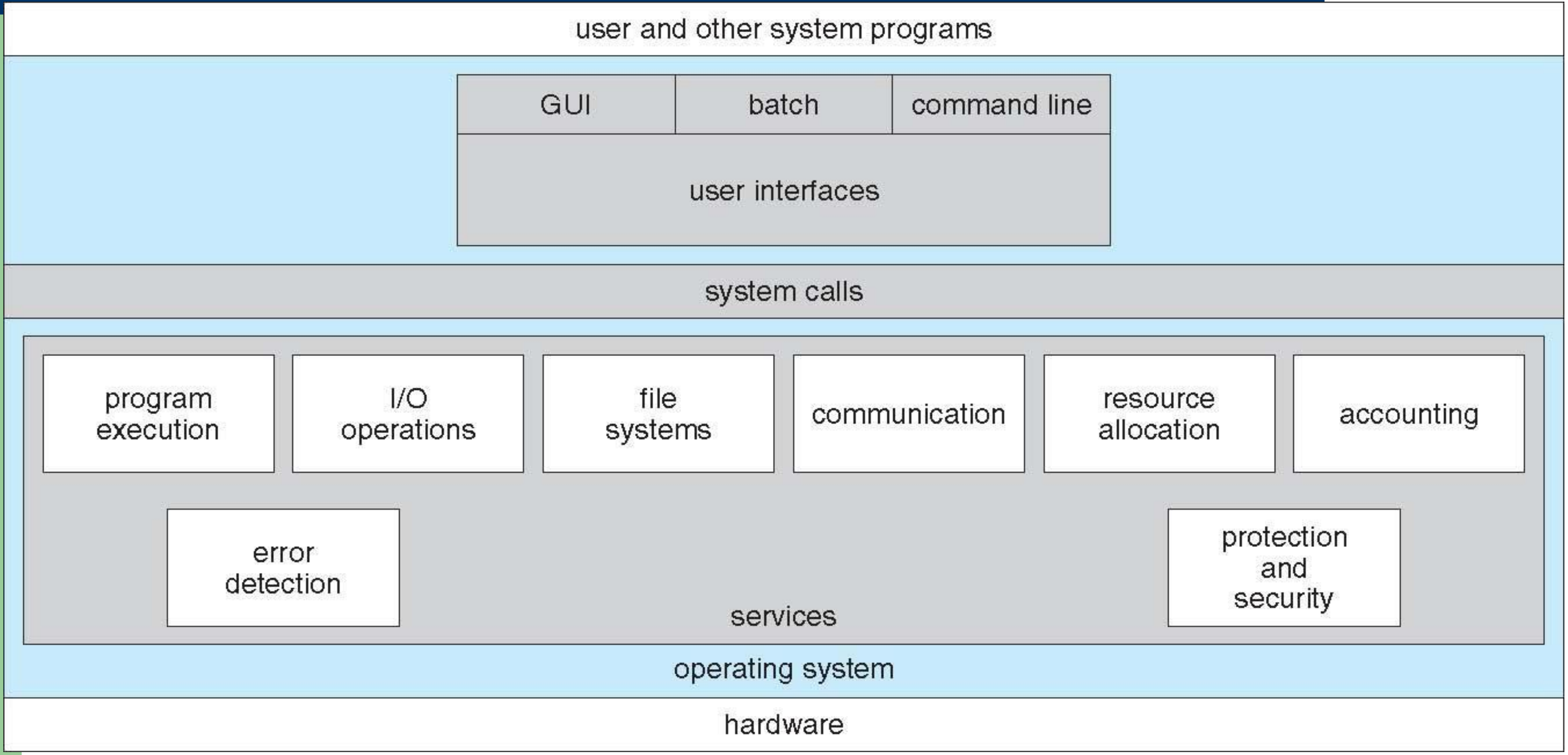


Features	Multiprogramming	Multitasking
Basic	It allows multiple programs to utilize the CPU simultaneously.	A supplementary of the multiprogramming system also allows for user interaction.
Mechanism	Based on the context switching mechanism.	Based on the time-sharing mechanism.
Objective	It is useful for reducing/decreasing CPU idle time and increasing throughput as much as possible.	It is useful for running multiple processes at the same time, effectively increasing CPU and system throughput.
Execution	When one job or process completes its execution or switches to an I/O task in a multi-programmed system, the system momentarily suspends that process. It selects another process from the process scheduling pool (waiting queue) to run.	In a multiprocessing system, multiple processes can operate simultaneously by allocating the CPU for a fixed amount of time.



CPU Switching	In a multiuser environment, the CPU switches between programs/processes quickly.	In a single-user environment, the CPU switches between the processes of various programs.
Timing	It takes maximum time to execute the process.	It takes minimum time to execute the process.

A View of Operating System Services



Services provided by OS

- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation
- Protection
- Accounting

1.Program Execution

- The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)

  - Loads a program into memory.
  - Executes the program.
  - Handles program's execution.
  - Provides a mechanism for process synchronization.
  - Provides a mechanism for process communication.
  - Provides a mechanism for deadlock handling.

2.I/O operations

- A running program may require I/O, which may involve a file or an I/O device.
  - I/O operation means read or write operation with any file or any specific I/O device.
  - Operating system provides the access to the required I/O device when required.

3.File System manipulation

The file system is of particular interest. Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

4.Communication

Processes may exchange information, on the same computer or between computers over a network.

- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or by Message Passing.

5.Error detection

- OS needs to be constantly aware of possible errors.
  - The OS constantly checks for possible errors.
  - The OS takes an appropriate action to ensure correct and consistent computing.
  - May occur in the CPU and memory hardware, in I/O devices, in user program.
  - For each type of error, OS should take the appropriate action to ensure correct and consistent computing.
  - Debugging facilities can greatly enhance the user’s and programmer’s abilities to efficiently use the system.

6.Resource allocation

- When multiple users or multiple jobs running concurrently, resources must be allocated to each of them.
  - The OS manages all kinds of resources using schedulers.
  - CPU scheduling algorithms are used for better utilization of CPU.
  - Many types of resources - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code.

7.Accounting - To keep track of which users use how much and what kinds of computer resources

8.Protection and security

- The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other

  Protection involves ensuring that all access to system resources is controlled

  Security of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts

  If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

User Operating System Interface - CLI

- Command Line Interface (CLI) or **command interpreter** allows direct command entry
    - Sometimes implemented in kernel, sometimes by systems program
    - Sometimes multiple flavors implemented – **shells**
    - Primarily fetches a command from user and executes it
        - Sometimes commands built-in, sometimes just names of programs
            - If the latter, adding new features doesn’t require shell modification

User Operating System Interface - GUI

- User-friendly **desktop** metaphor interface
  - Usually mouse, keyboard, and monitor
  - **Icons** represent files, programs, actions, etc
  - Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a **folder**)

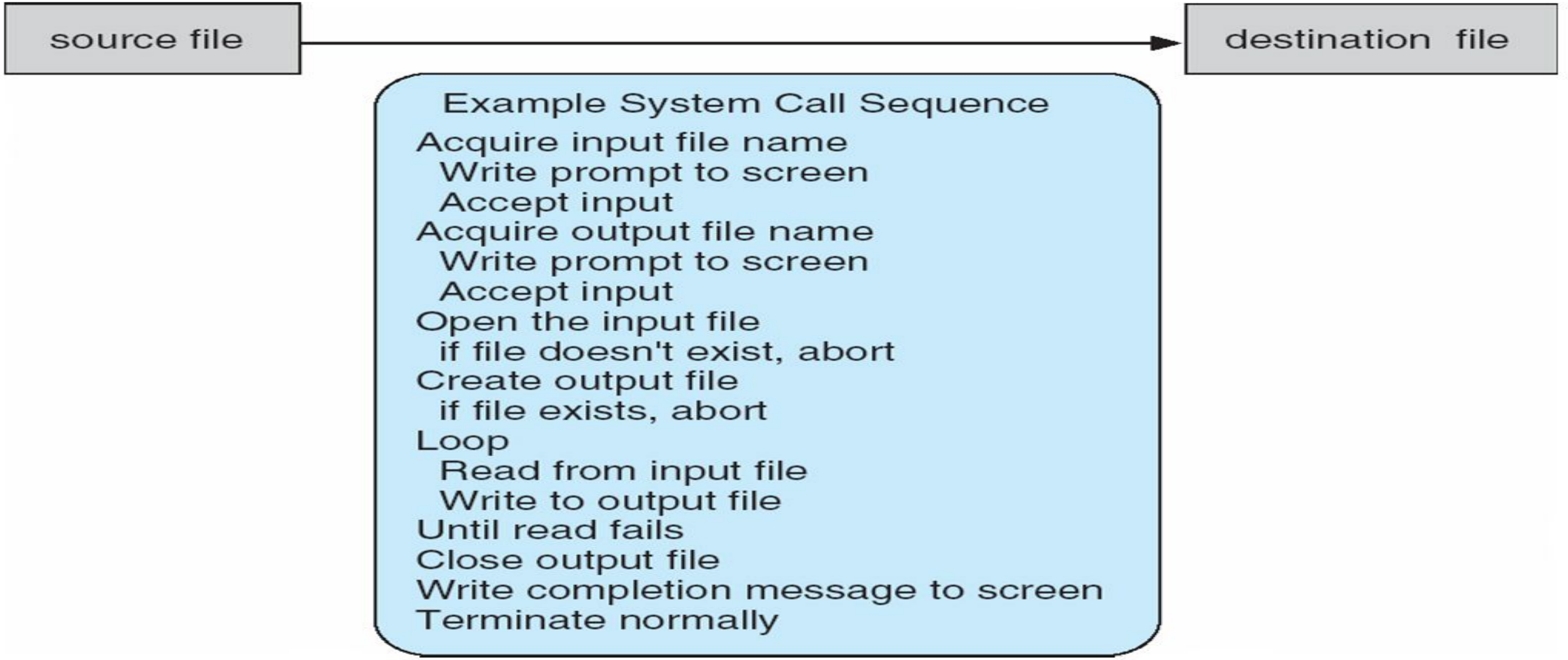
- Many systems now include both CLI and GUI interfaces
  - Microsoft Windows is GUI with CLI “command” shell
  - Apple Mac OS X as “Aqua” GUI interface with UNIX kernel underneath and shells available
  - Solaris is CLI with optional GUI interfaces (Java Desktop, KDE)

System Calls

- Programming interface to the services provided by the OS
- Typically written in a high-level language (C or C++)
- Mostly accessed by programs via a high-level **Application Program Interface (API)** rather than direct system call use
- Three most common APIs -
  - Win32 API for Windows,
  - POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X),
  - Java API for the Java virtual machine (JVM)

Example of System Calls

- System call sequence to copy the contents of one file to another file.



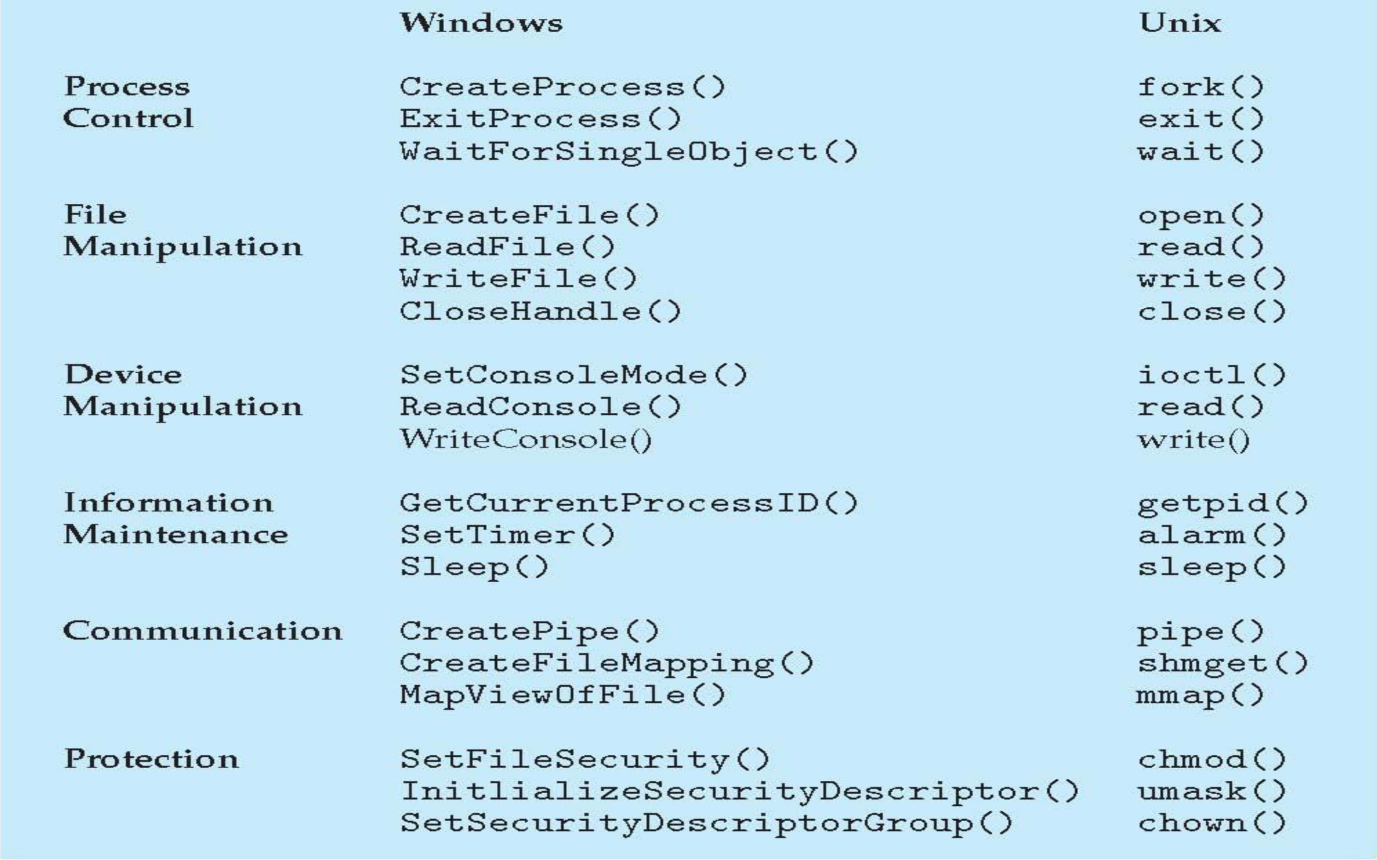
Example System Call Sequence

```
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

Types of System Calls

- **Process control**
  - end, abort
  - load, execute
  - create process, terminate process
  - get process attributes, set process attributes
  - wait for time
  - wait event, signal event
  - allocate and free memory
- **File management**
  - create file, delete file
  - open, close file
  - read, write, reposition
  - get and set file attributes

Examples of Windows and Unix System Calls



	Windows	Unix

Process Control	CreateProcess()
	ExitProcess()
	WaitForSingleObject()
		fork()
		exit()
		wait()

File Manipulation	CreateFile()
	ReadFile()
	WriteFile()
	CloseHandle()
		open()
		read()
		write()
		close()

Device Manipulation	SetConsoleMode()
	ReadConsole()
	WriteConsole()
		ioctl()
		read()
		write()

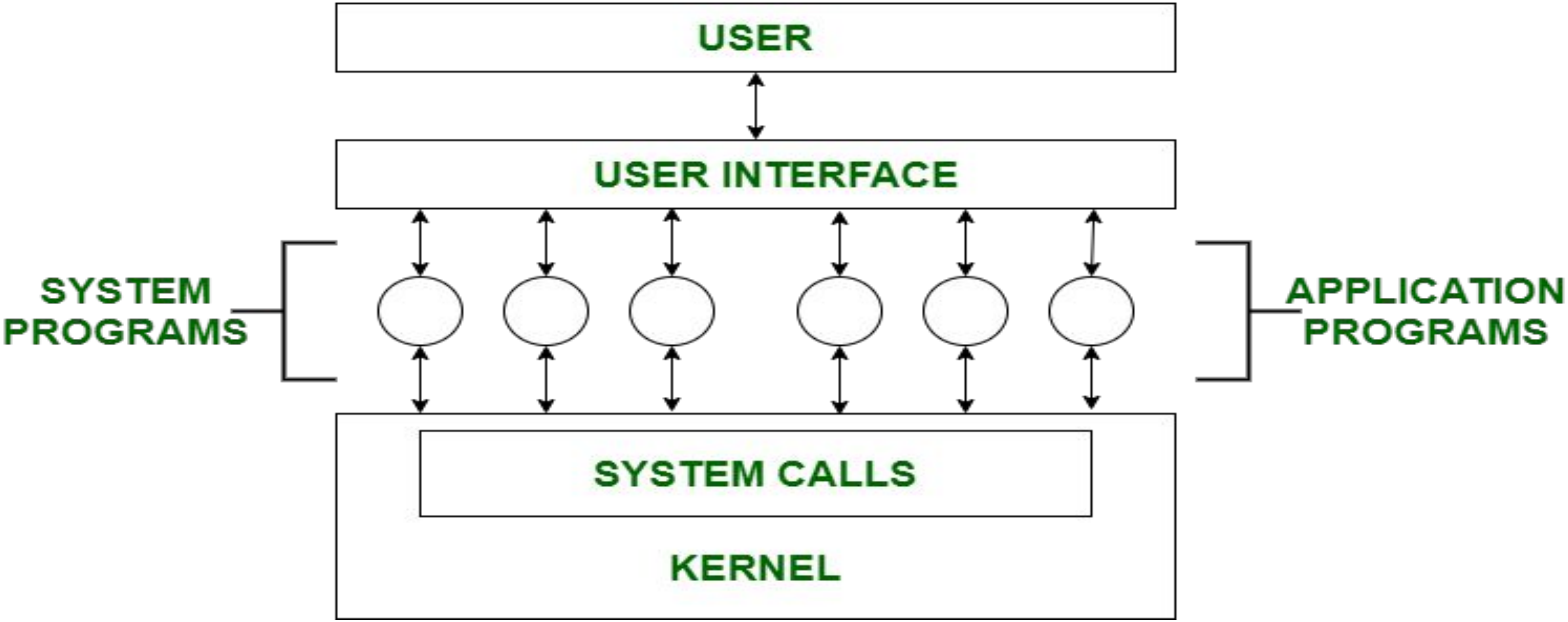
Information Maintenance	GetCurrentProcessID()
	SetTimer()
	Sleep()
		getpid()
		alarm()
		sleep()

Communication	CreatePipe()
	CreateFileMapping()
	MapViewOfFile()
		pipe()
		shmget()
		mmap()

Protection	SetFileSecurity()
	InitlializeSecurityDescriptor()
	SetSecurityDescriptorGroup()
		chmod()
		umask()
		chown()

System Programs

- System programs provide a convenient **environment for program development** **and execution.**
-



File management

- Create, delete, copy, rename, print, dump, list, and generally manipulate files and directories.
    - It helps to create new files in the computer system and placing them at specific locations.
    - It helps in easily and quickly locating these files in the computer system.
    - It makes the process of sharing files among different users very easy and user-friendly.
    - It helps to store files in separate folders known as directories.
    - These directories help users to search files quickly or to manage files according to their types of uses.
    - It helps users to modify the data of files or to modify the name of files in directories.

Status Information

- Some ask the system for info - date, time, amount of available memory, disk space, number of users
- Others provide detailed performance, logging, and debugging information
- Typically, these programs format and print the output to the terminal or other output devices
- Some systems implement a registry - used to store and retrieve configuration information

File modification

- Text editors to create and modify files
- Special commands to search contents of files or perform transformations of the text.

Programming-language support -

Compilers, assemblers, debuggers and interpreters sometimes provided

Program Loading and Execution –

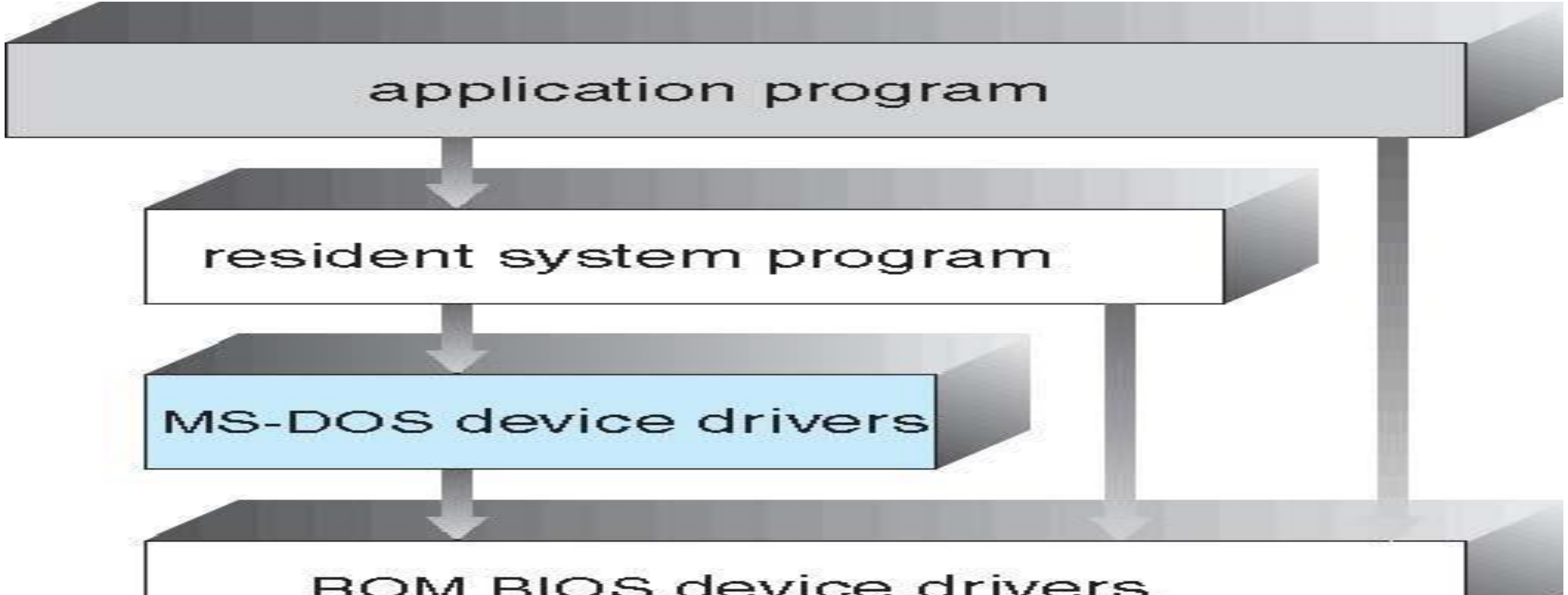
Absolute loaders, relocatable loaders, linkage editors, and overlay-loaders, debugging systems for higher-level and machine language

Communications -

- Provide the mechanism for creating virtual connections among processes, users, and computer systems.
- Allow users to send messages to one another’s screens, browse web pages, send electronic-mail messages, log in remotely, transfer files from one machine to another

1.Simple Structure

- Such operating systems do not have well defined structure and are small, simple and limited systems.
- The interfaces and levels of functionality are not well separated.
- MS-DOS is an example of such operating system.
- In MS-DOS application programs are able to access the basic I/O routines. These types of operating system cause the entire system to crash if one of the user programs fails.



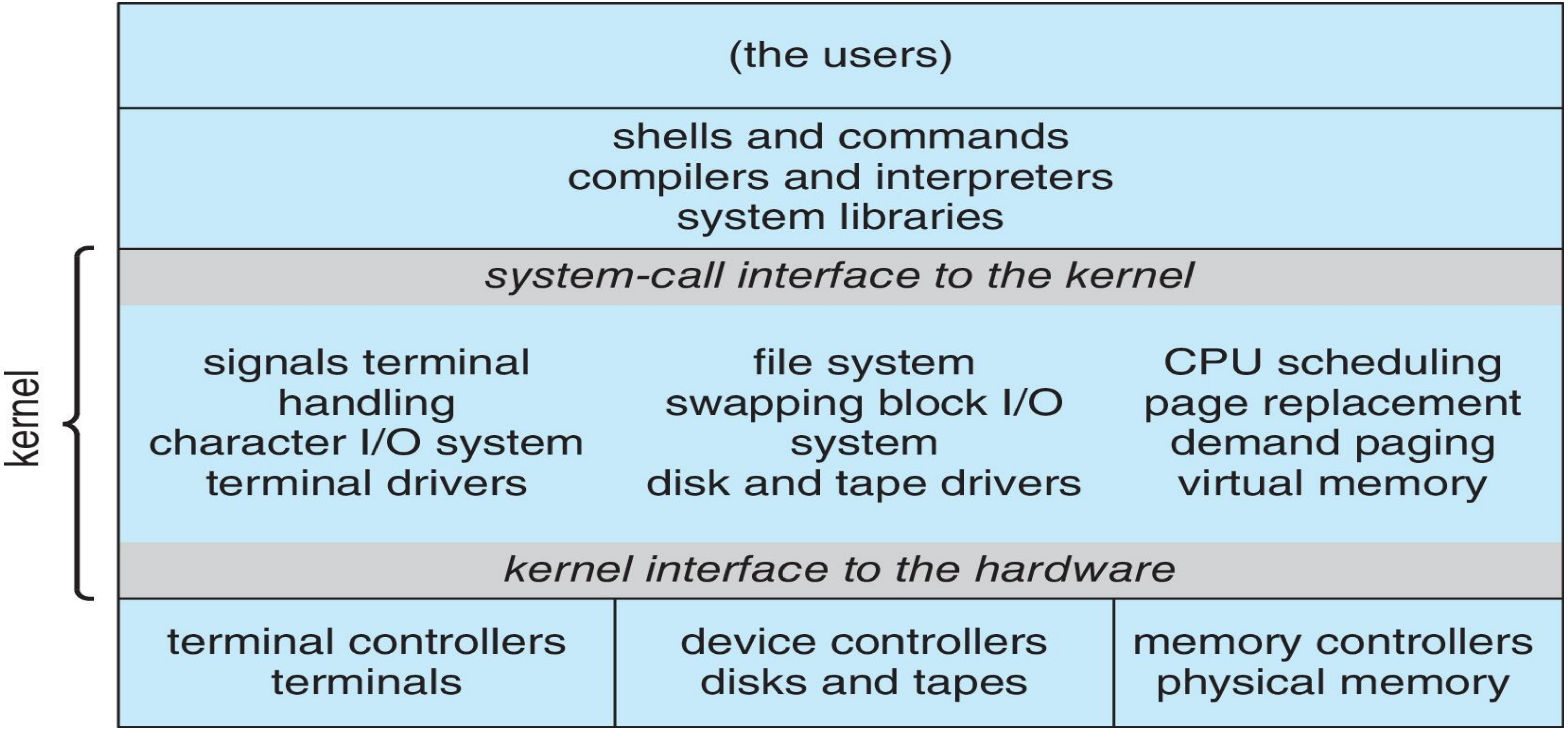
Advantages of Simple structure:

- It delivers better application performance because of the few interfaces between the application program and the hardware.
- Easy for kernel developers to develop such an operating system.

Disadvantages of Simple structure:

- The structure is very complicated as no clear boundaries exists between modules.
- It does not enforce data hiding in the operating system.

2. Monolithic Structure



(the users)

shells and commands
compilers and interpreters
system libraries

system-call interface to the kernel

signals terminal handling
character I/O system
terminal drivers

file system
swapping block I/O system
disk and tape drivers

CPU scheduling
page replacement
demand paging
virtual memory

kernel interface to the hardware

terminal controllers
terminals

device controllers
disks and tapes

memory controllers
physical memory

Monolithic Structure

- The Monolithic operating system in which the kernel acts as a manager by managing all things like file management, memory management, device management, and operational processes of the Operating system.
- The Kernel is heart of computer Operating system. Kernel delivers basic services to all other elements of the system.
- It serves as the primary interface between the operating system and hardware.

Advantages of Monolithic structure:

- Layering is unnecessary and the kernel alone is responsible for managing all operations, it is easy to design and execute.

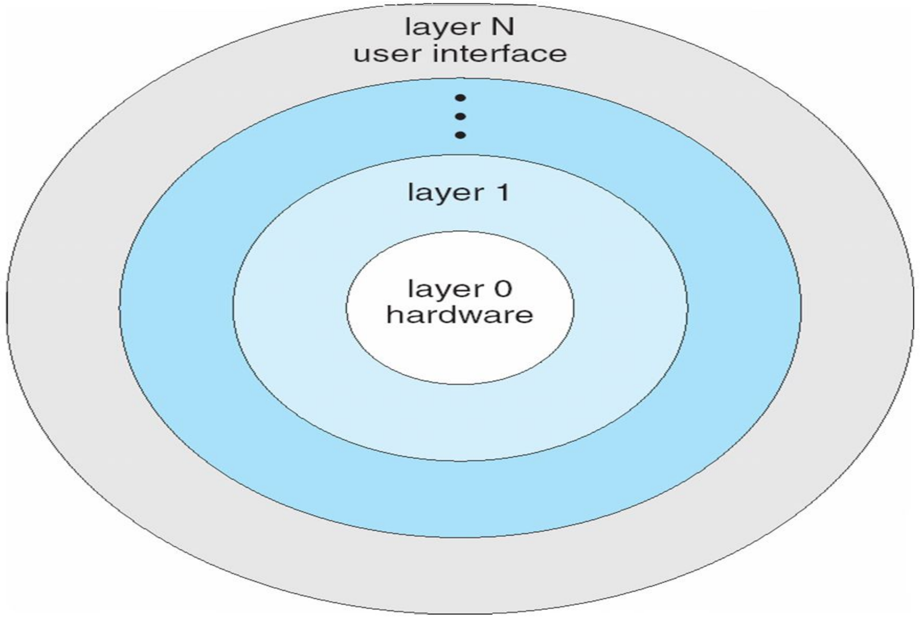
Disadvantages of Monolithic structure:

- The monolithic kernel services are interconnected in one address space and have an impact on one another, so if any of them malfunctions, the entire system does as well.
- It is not adaptable. Therefore, launching a new service is difficult.

3.Layered structure

- The operating system is divided into a 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.

- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers



Advantages of Layered structure:

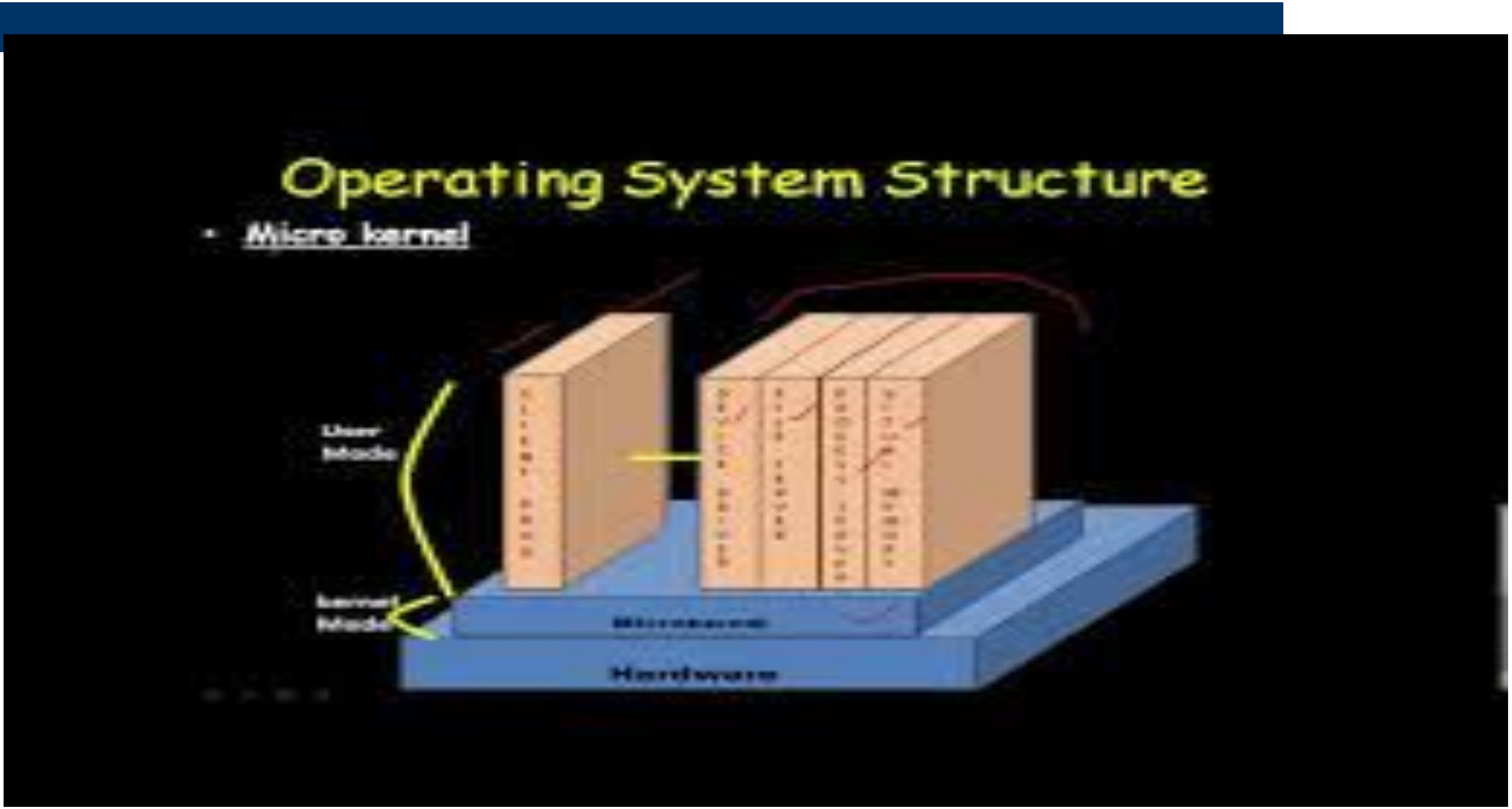
- Layered approach provides Modularity. Layering makes it easier to enhance the operating system as implementation of a layer can be changed easily without affecting the other layers.
- It is very easy to perform debugging and system verification.

Disadvantages of Layered structure:

- In this structure the application performance is degraded as compared to simple structure.
- It requires careful planning for designing the layers as higher layers use the functionalities of only the lower layers.

4.Microkernel

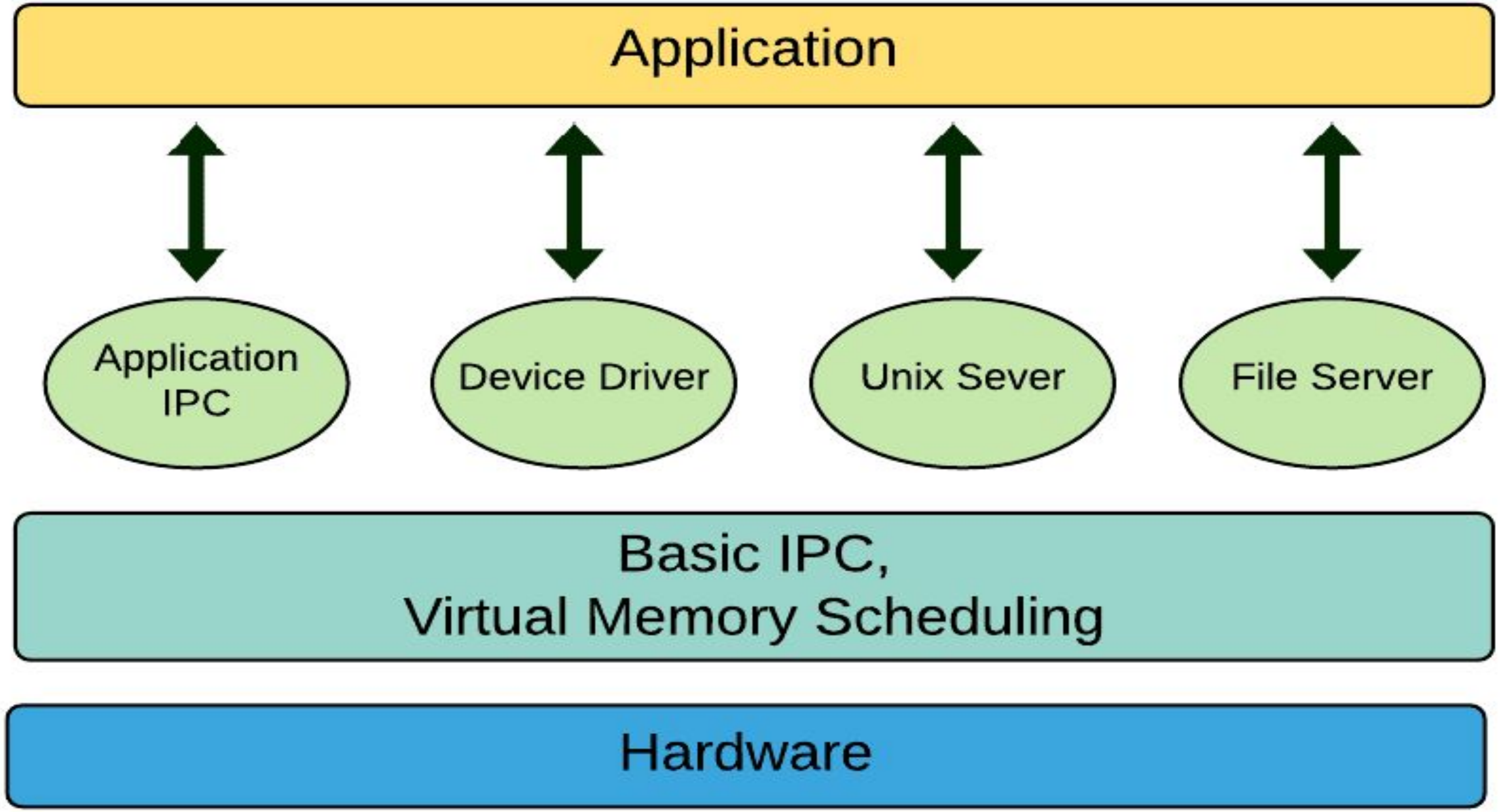
A microkernel is one of the classifications of the kernel. Being a kernel it manages all system resources. But in a microkernel, the **user services** and **kernel services** are implemented in different address spaces. The user services are kept in **user address space**, and kernel services are kept under **kernel address space**, thus also reduces the size of kernel and size of an operating system as well.



Operating System Structure

- Micro kernel

MicroKernel Operating System



Microkernel System Structure

This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel.

Advantages of Micro-kernel structure:

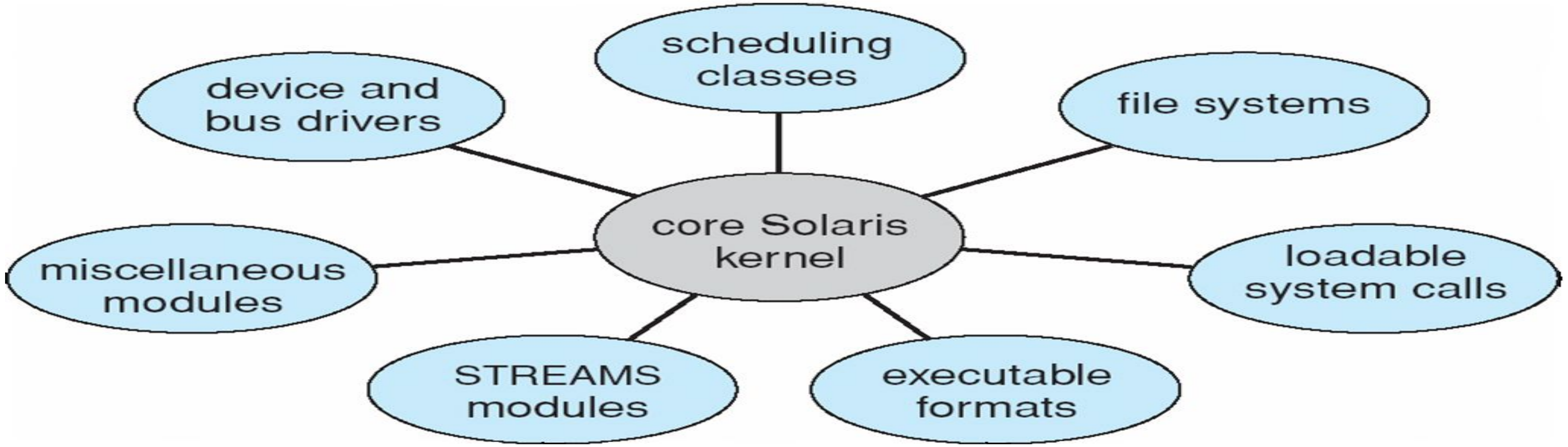
- It makes the operating system portable to various platforms.
- As microkernels are small so these can be tested effectively.

Disadvantages of Micro-kernel structure:

- Increased level of inter module communication degrades system performance.

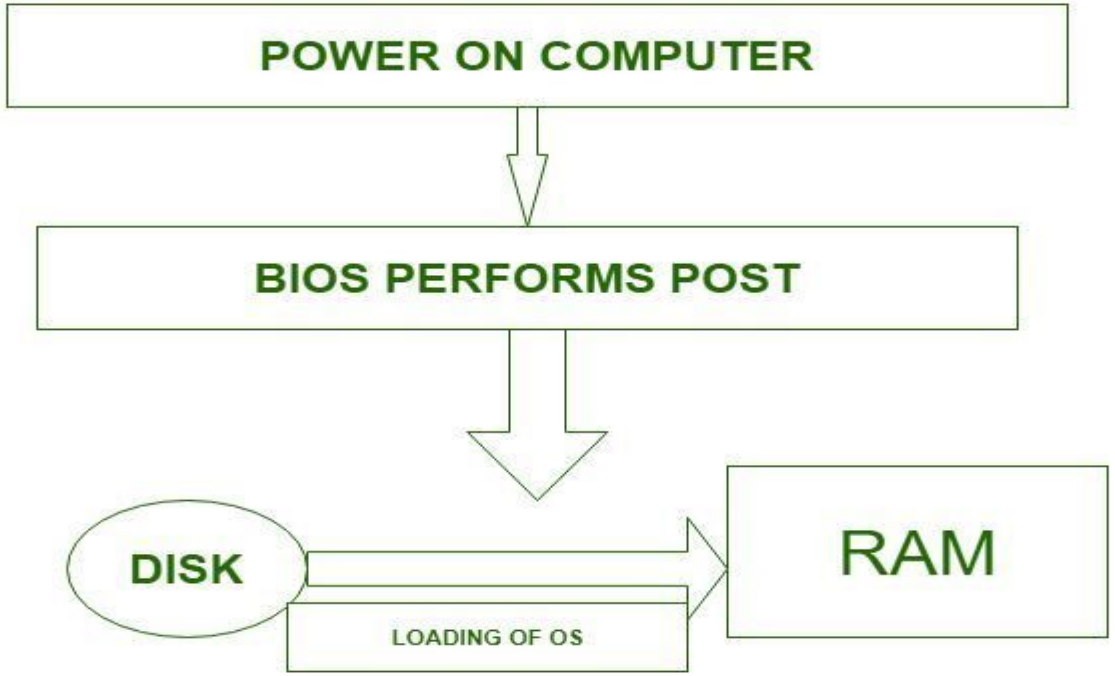
5.Modular structure or approach

- It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time.
- It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module. For example Solaris OS is organized as shown in the figure.



System Boot

- Booting is a startup sequence that starts the operating system of a computer when it is turned on.
- A boot sequence is the initial set of operations that the computer performs when it is switched on

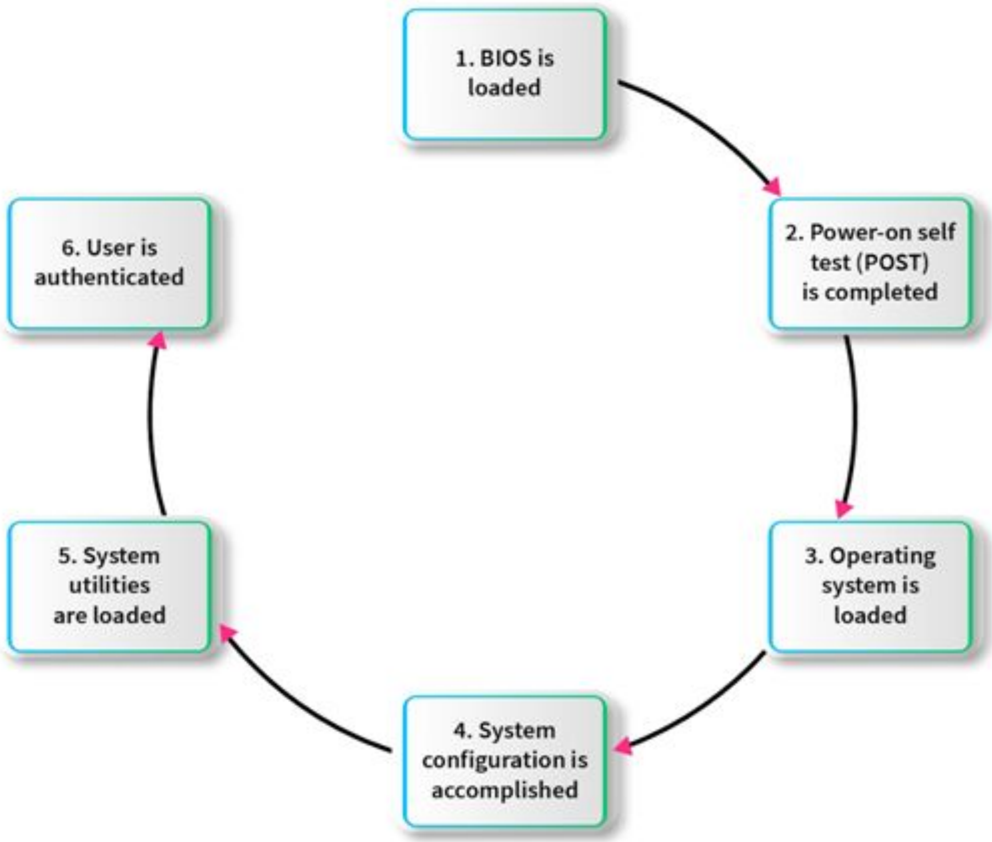


System Boot

- Operating system must be made available to hardware so hardware can start it
  - Small piece of code – **bootstrap loader**, locates the kernel, loads it into memory, and starts it
  - Sometimes two-step process where **boot block** at fixed location loads bootstrap loader
  - When power initialized on system, execution starts at a fixed memory location

    -Firmware used to hold initial boot code.

Steps in Booting



1. BIOS is loaded
2. Power-on self test (POST) is completed
3. Operating system is loaded
4. System configuration is accomplished
5. System utilities are loaded
6. User is authenticated

1. **Loading of BIOS:** The small set on instructions present in the ROM is loaded into the computer memory and CPU executes those instruction.

2. **Power-On Self Test (POST):** In order to check the operability of all the hardware connected to our computer system, BIOS carries out POST which will check the hardware components and if any problem is found user is alerted with POST beeps and POST screen messaged.

3. **Loading of Operating System:**

- After the successful completion of POST, the bootable sequence present in CMOS (Common Metal Oxide Semiconductor) is read by BIOS.
- Based on the bootable sequence it will search for Master Boot Record (MBR) in bootable device like floppy disk, CD-ROM and hard disk.
- If MBR is not found in any of them, the system will halt by displaying "No Boot Device Found".
- if MBR is found, the BIOS will load the special application program called Boot Loader, that will eventually load the Operating system.

1. **System Configuration is Accomplished:** After the OS is loaded, device drivers are loaded into the memory so that our devices can function correctly.

2. **System Utilities are Loaded:** System utilities like antivirus, volume control etc. are loaded into the memory in this step

3. **User Authentication:** If any user authentication is being set, the system will ask the user to enter the credentials and on receiving the correct credentials the computer system will run GUI shell (in most cases) or CLI shell.

Thank you!