

OPERATING SYSTEMS

LECTURE – 1

INTRODUCTION

K.ARIVUSELVAN

Assistant Professor (Senior) – (SITE)

VIT University

Why Operating system?

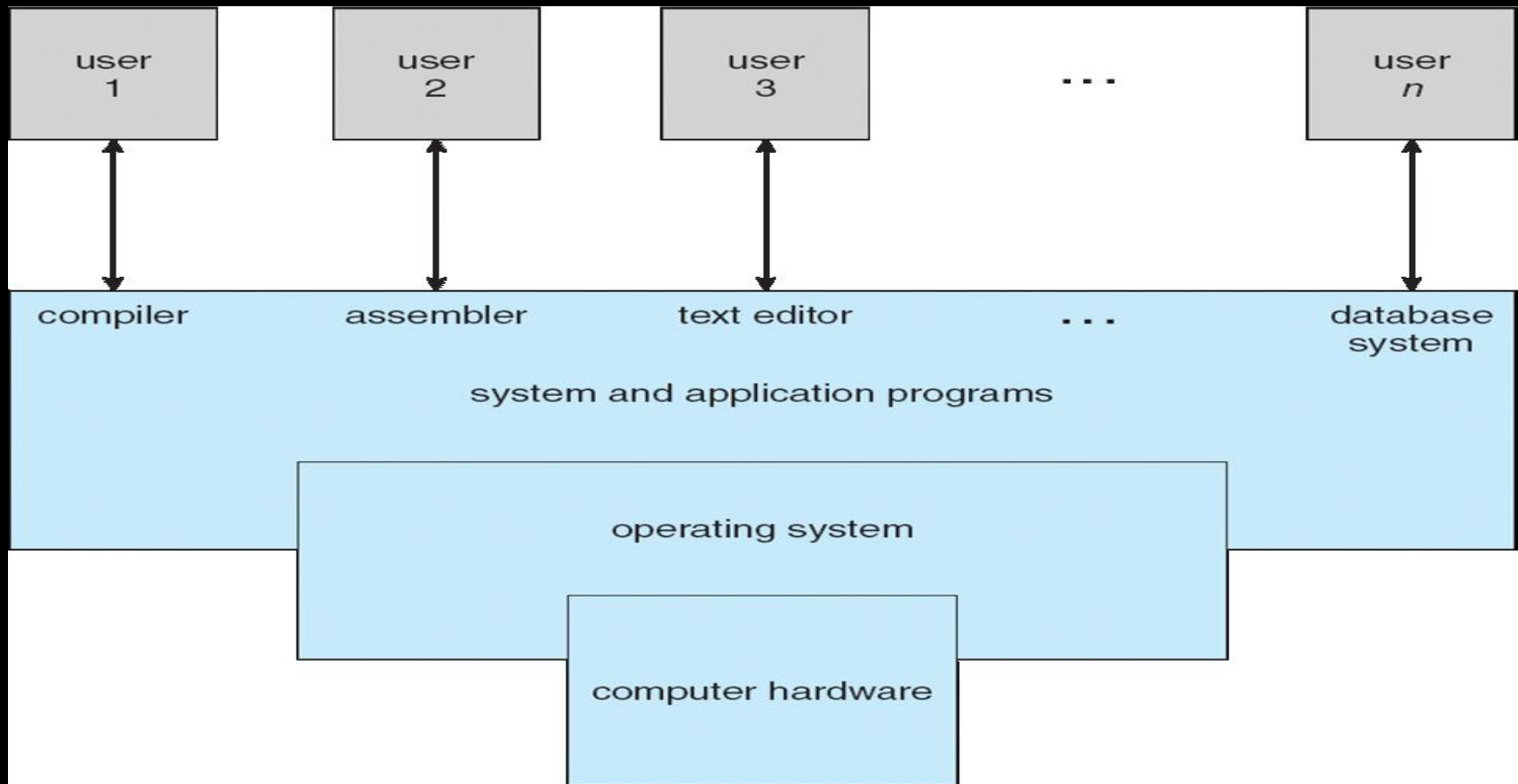
A modern computer **consists** of:

- One or more processors
- Main memory
- Disks
- Printers
- Various input/output devices

Managing all these varied **components** requires a **layer of software** – the **operating system**

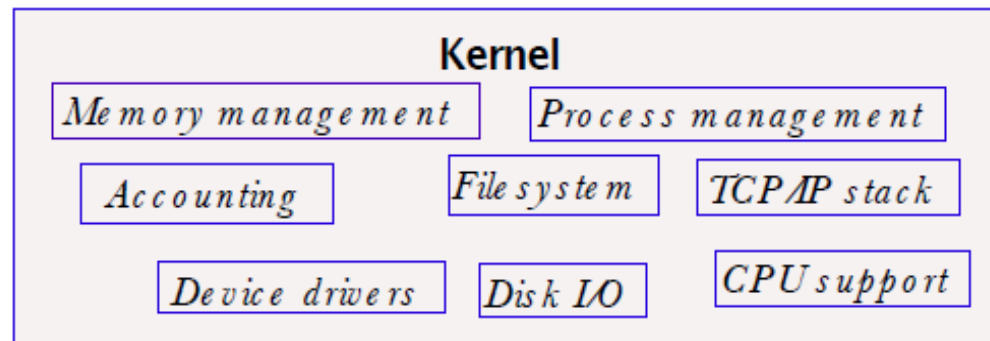
What is an Operating System?

An Operating System is a **program** that acts as an **intermediary/interface** between a **user** of a computer and the computer **hardware**



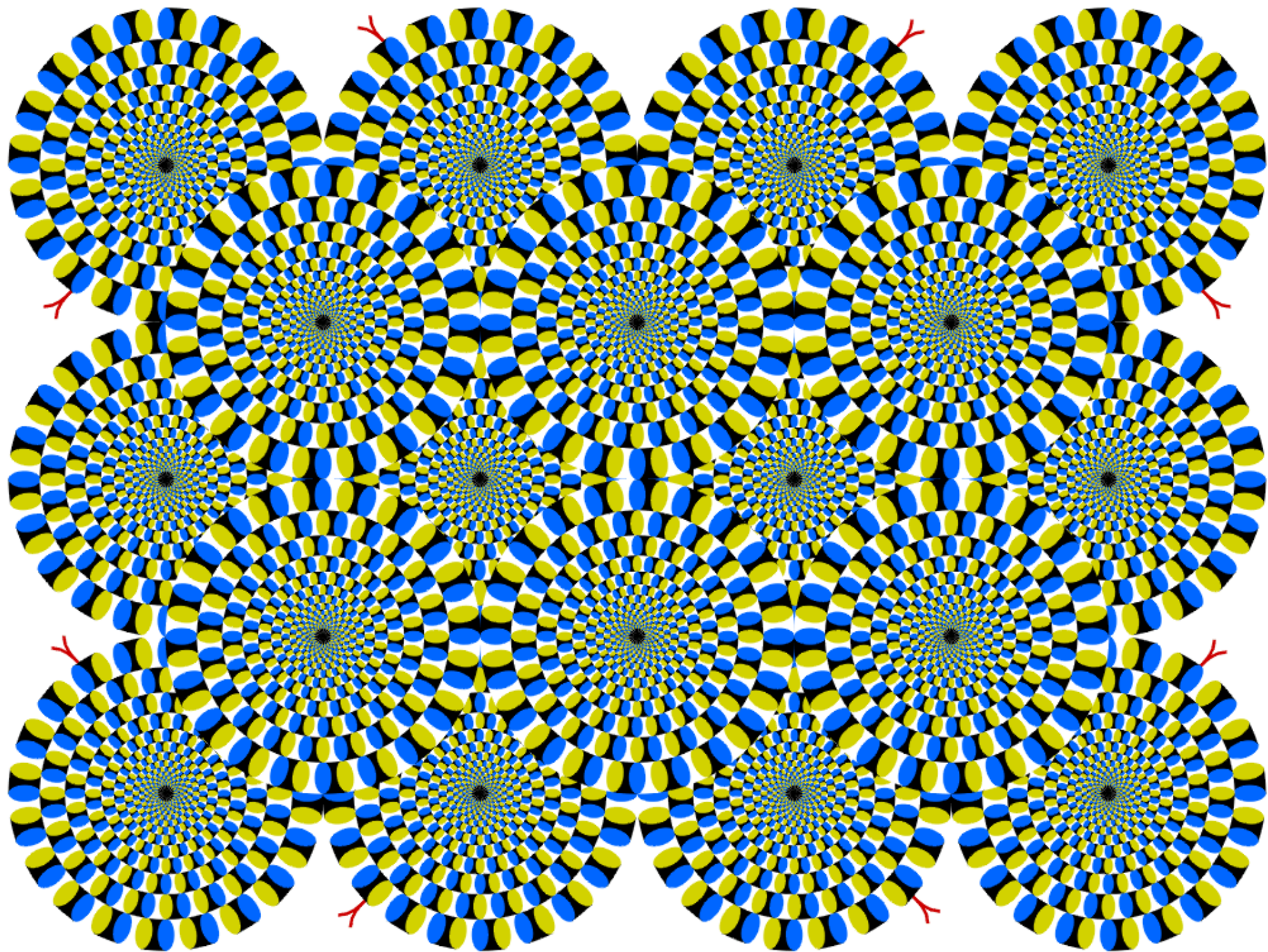
What is an operating system?

❑ Software that provides an **elaborate illusion** to **applications**



Hardware / software
interface







EMERGENCY DAY 2006

Important features of OS:

(1) ABSTRACTION:

- ❑ **Hides details of different hardware configurations**

(2) ARBITRATION:

- ❑ **Manages access to shared hardware resources**

OS Functions

(1) Concurrency:

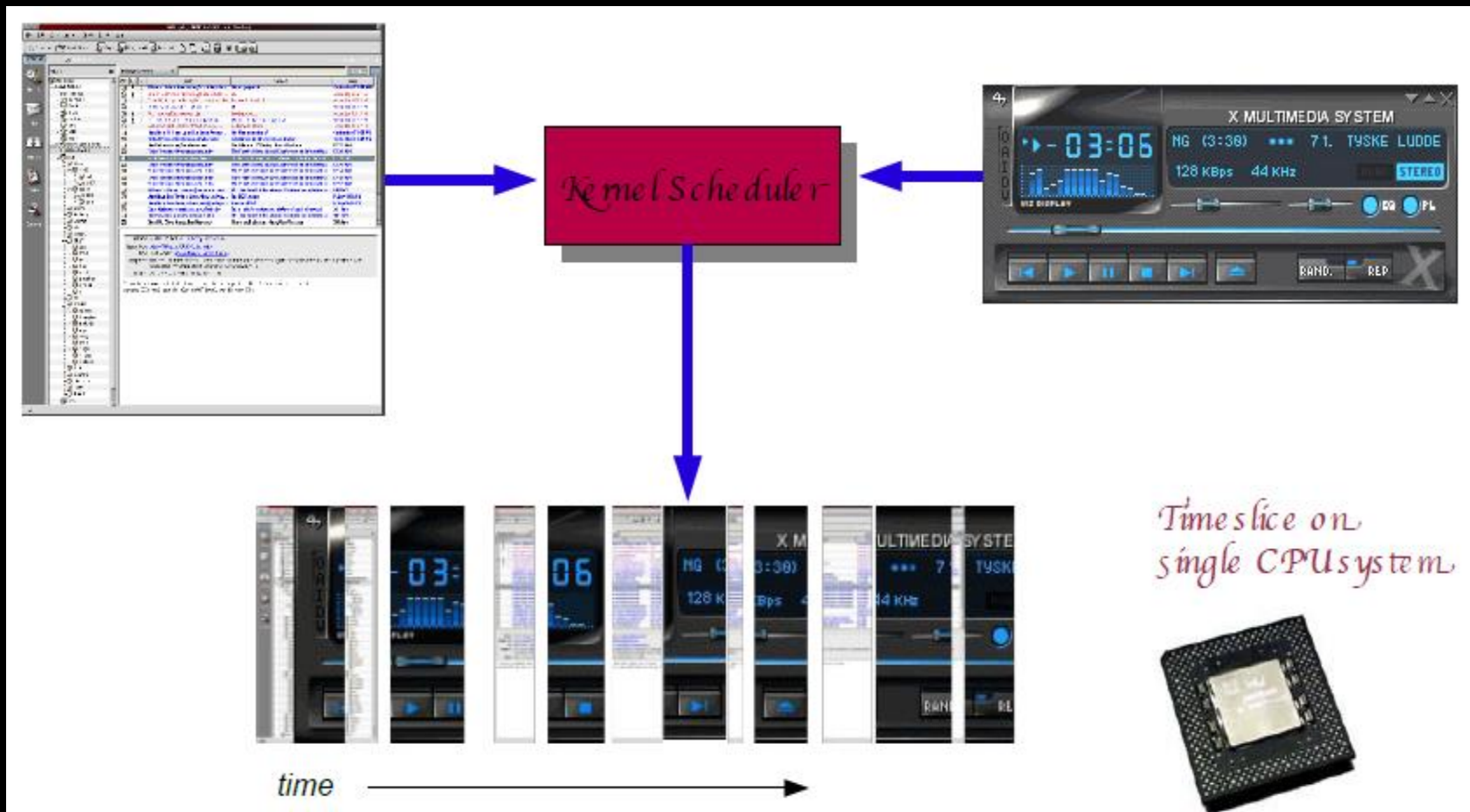
❑ Give every application the **illusion of having its own CPU!**



Concurrency <<< . >>> HOW?

❑ The OS *timeslices* each application on a *single CPU*

- Switches between applications extremely rapidly, i.e., **100 times/sec**



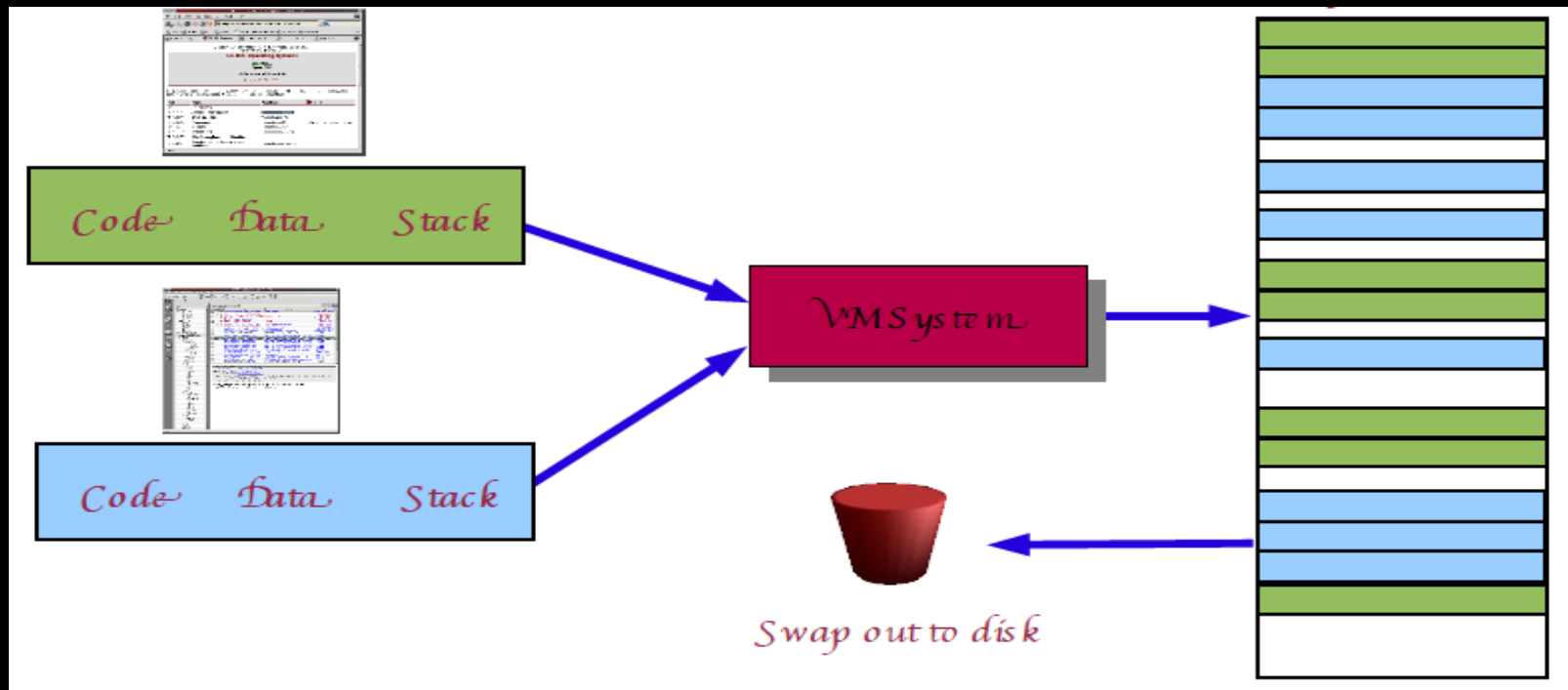
OS Functions

(2) Virtual Memory:

□ Give every application the illusion of having infinite memory

- And, that it can access any memory address it likes!

- In reality, RAM is split across multiple applications



(3) Multiprocessor support:

- Modern systems have **multiple CPUs**
- Can run multiple applications ***in parallel***

(4) Filesystems:

- Real disks have a **sector-based access** model
- User applications see flat files arranged in a **hierarchical namespace**

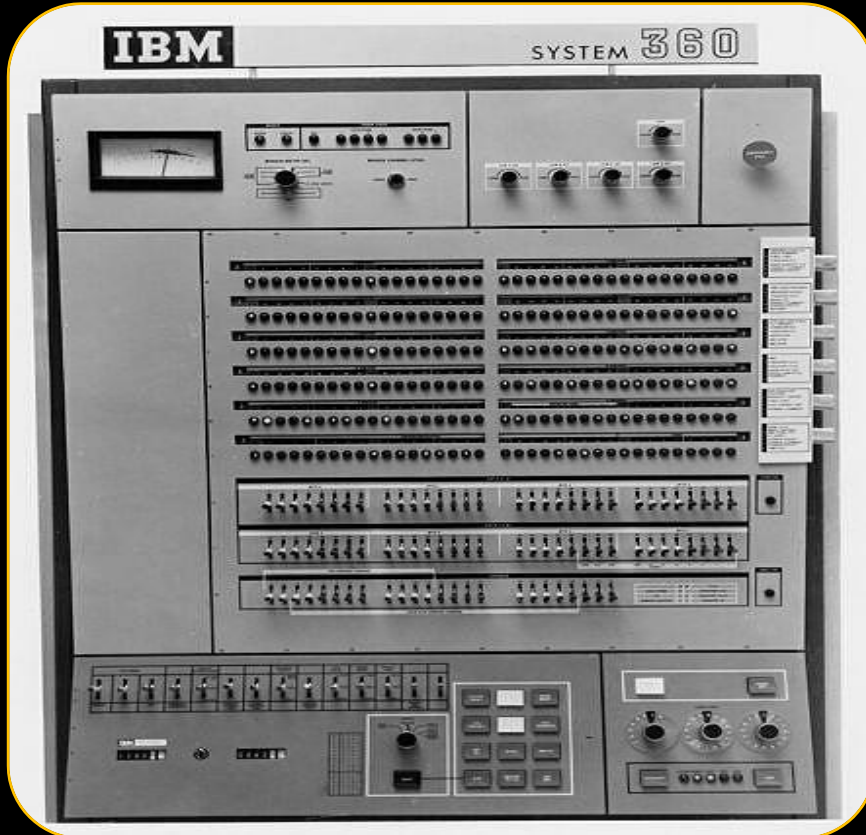
(5) Security and protection:

- Prevent multiple apps from **interfering with each other** and with normal system operation

OS TYPES

(1) In the Beginning...

- There was no OS – just libraries
- Computer only ran one program at a time, so no need for an OS



OS TYPES

(2) Simple batch systems

❑ (mid 1950s – mid 1960s)



❑ Permanently resident OS in primary memory

❑ Loaded a single job from **card reader**, ran it, loaded next job...

Cons:

- **Lack of interaction** between user & Job
- CPU is Often Idle

IBM Model 701 (Early 1950's)



IBM 7094 (Early 1960's)

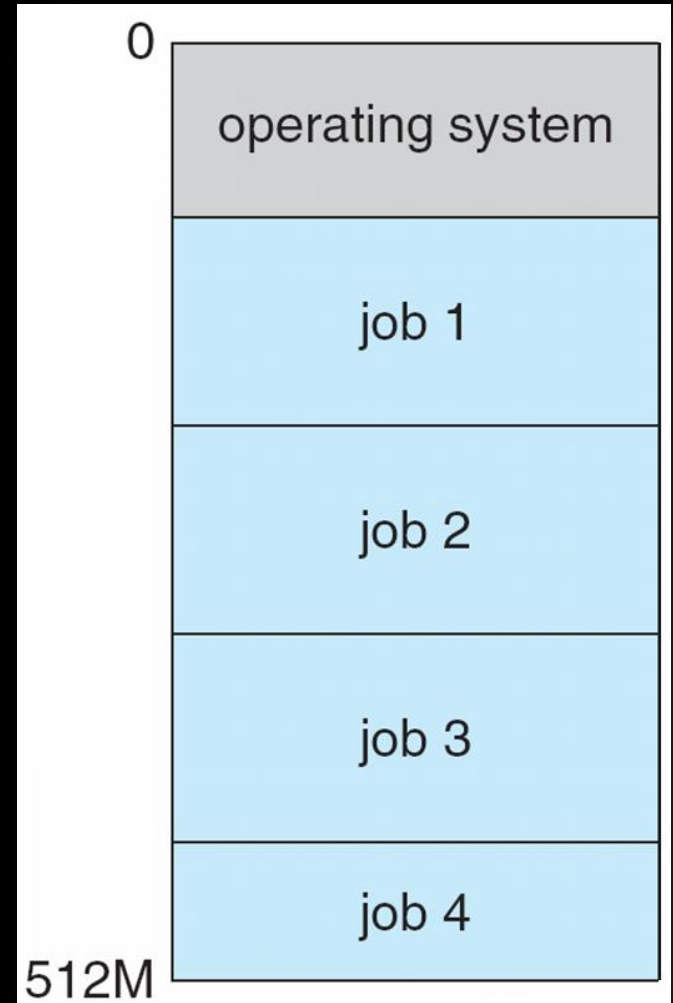


(2) Multi Programming System:

IDEA: Multiple Jobs reside in Main memory
(CPU Switched between jobs)

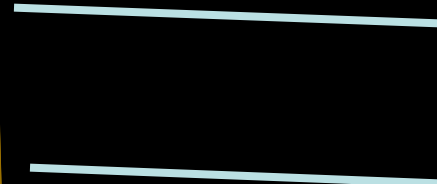
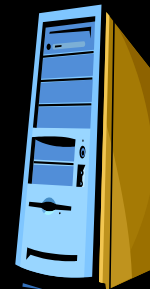
Pros:

- CPU, Memory, I/O Utilized effectively



(3) Time Sharing (Multi Tasking)

- Multiple users **simultaneously access** the system through terminals
- **Processor's time** is shared among **multiple users**
- The CPU **switches jobs so frequently** that users can interact with each job while it is running, creating interactive computing
- Each **user have a impression** that he has her **own computer**, but actually **one computer** is being **shared among** many users



Distributed Systems (Loosely Coupled)

- Distribute **Computation** among **several processors**
- Processors **do not share** memory or a clock;
- Processors **communicate** with one another through **communication lines**

Pros:

- Resource sharing (**Sharing files** at remote sites)
- Load sharing