OPERATING SYSTEMS

CSC203S2, CSC203G2

About the course

- 2 credits = 30 hours of lectures and tutorials
- Slides will be uploaded to LMS

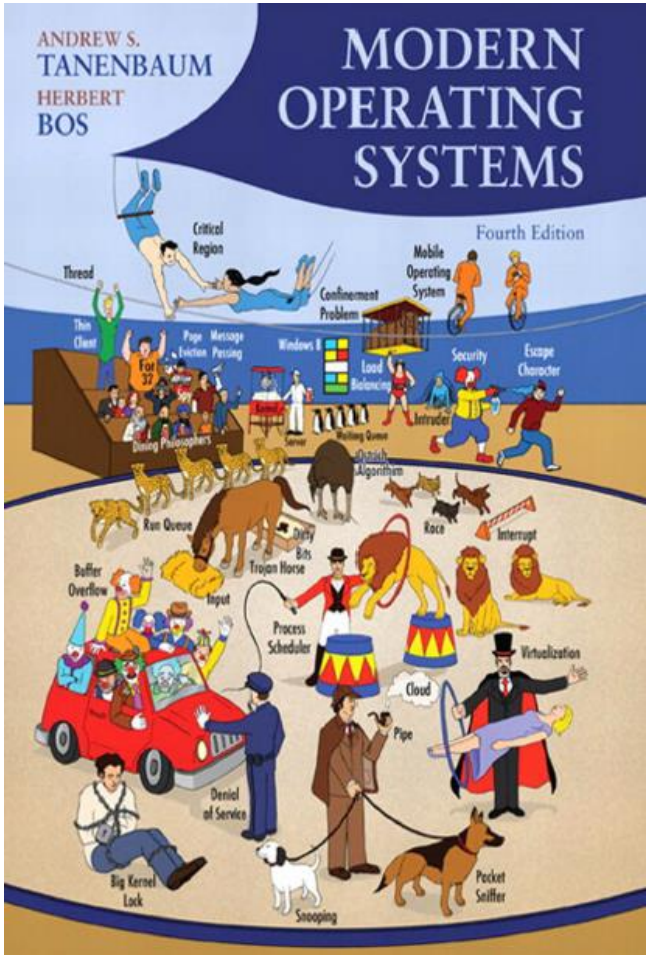
  - In-course Assessments ————————30%
  - End-of-course Examination ————70%

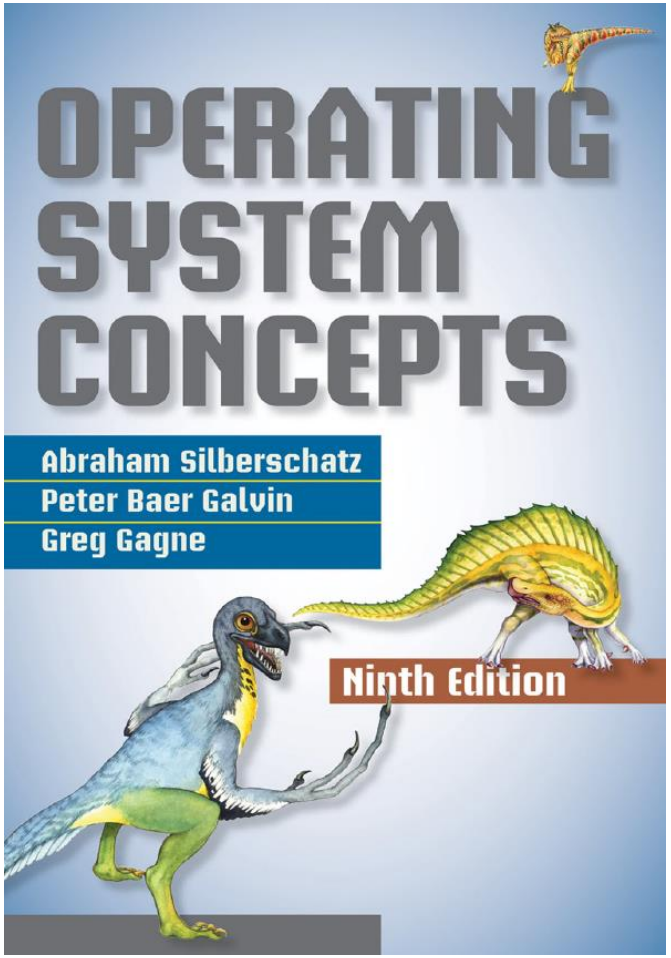
- Lecturers: Prof. M. Siyamalan & Dr. (Ms). R. Nirthika

Contents

- Introduction to operating system: Architecture of modern operating systems (OS), evolution of OS, OS operations and functionalities, and open source OS
- Processes and Threads: Concept of process, process states, process control block, schedulers, context switch, interprocess communication, process scheduling, overview of threads, multicore programming and multithreading models
- Concurrency: Process synchronisation (race condition, critical-section problem, mutex locks, semaphores, classic problems of synchronization and monitors), deadlock (characterization, prevention, avoidance, detection and recovery)
- Memory management: Swapping, memory allocation, fragmentation, paging, segmentation, virtual memory and address translation
- Storage management: Mass Storage, host attached storage, network attached storage, storage area network and RAID
- File and I/O Device management: File organization and access, file system security, device drivers, direct memory access and interrupt handling

Reading


ANDREW S.
TANENBAUM
HERBERT
BOS
MODERN
OPERATING
SYSTEMS
Fourth Edition
Critical Region
Thread
Mobile Operating System
Confinement Problem
Thin Client
Pipe
Message Passing
Windows
Load Balancing
Security
Escape Character
Intruder
Dining Philosophers
Server
Waiting Queue
Banish Algorithm
Run Queue
Dirty Bits
Trojan Horse
Race
Interrupt
Buffer Overflow
Input
Process Scheduler
Virtualization
Cloud
Pipe
Denial of Service
Big Kernel Lock
Snooping
Pocket Sniffer


OPERATING
SYSTEM
CONCEPTS
Abraham Silberschatz
Peter Baer Galvin
Greg Gagne
Ninth Edition

Chapter 1
Introduction

Outline

1.1 What is an Operating System (OS)

1.2 Services provided by OS

1.3 The history of OS

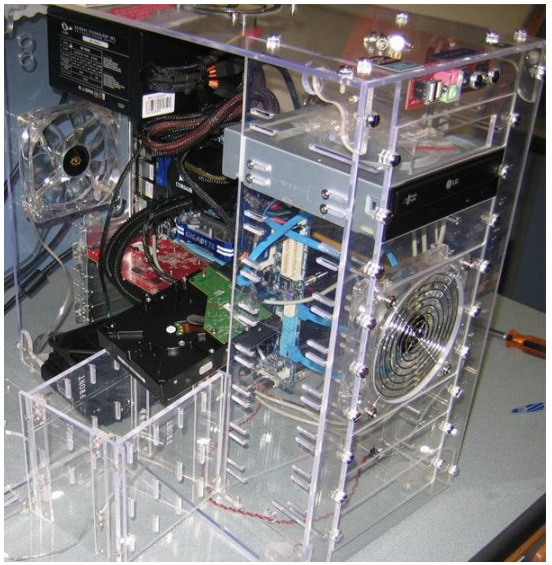
1.4 The OS zoo

1.5 OS Architectures

1.6 Important concepts of OS

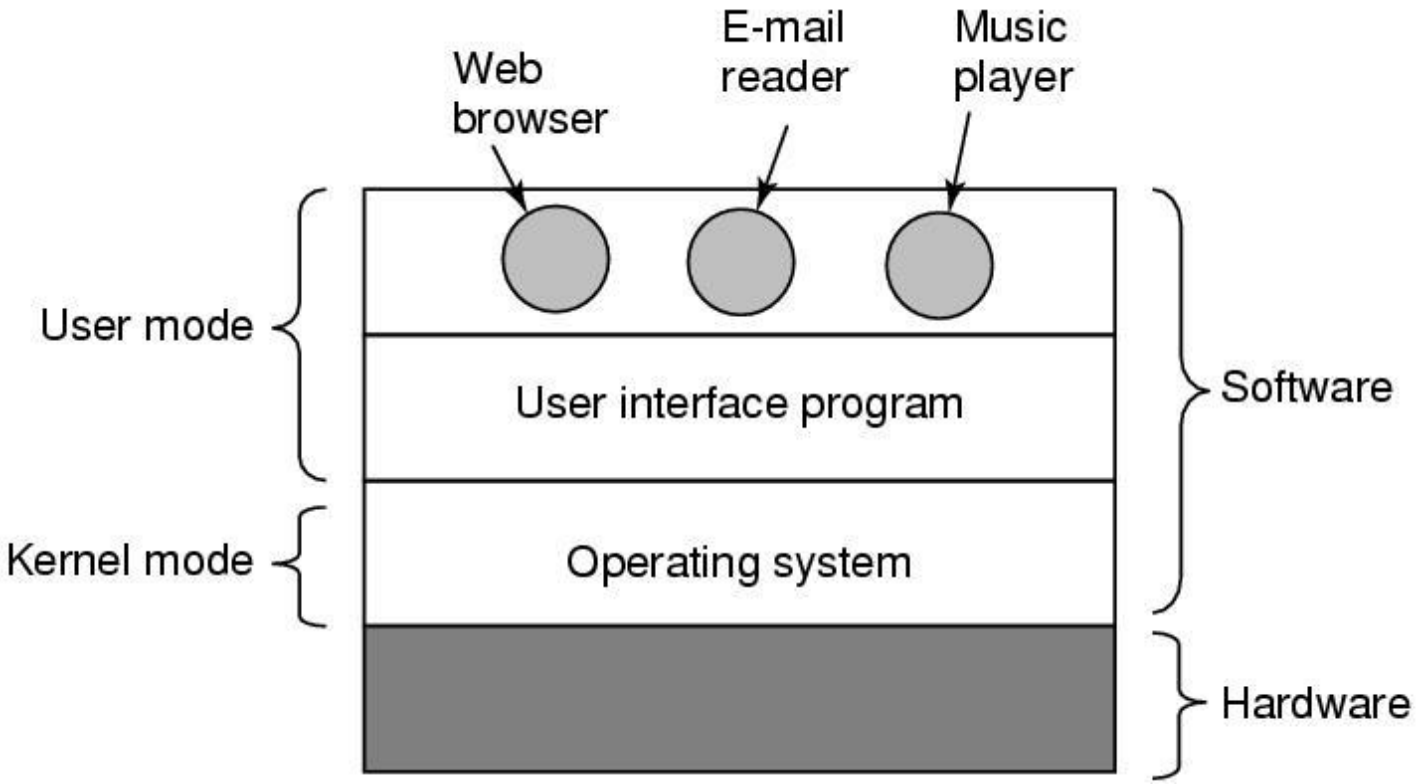
1.1 What is an Operating System?

- A modern computer is very complex. Consists of:
  - One or more processors
  - Main memory
  - Disks
  - Video/audio card
  - Various input/output devices
  - Etc.

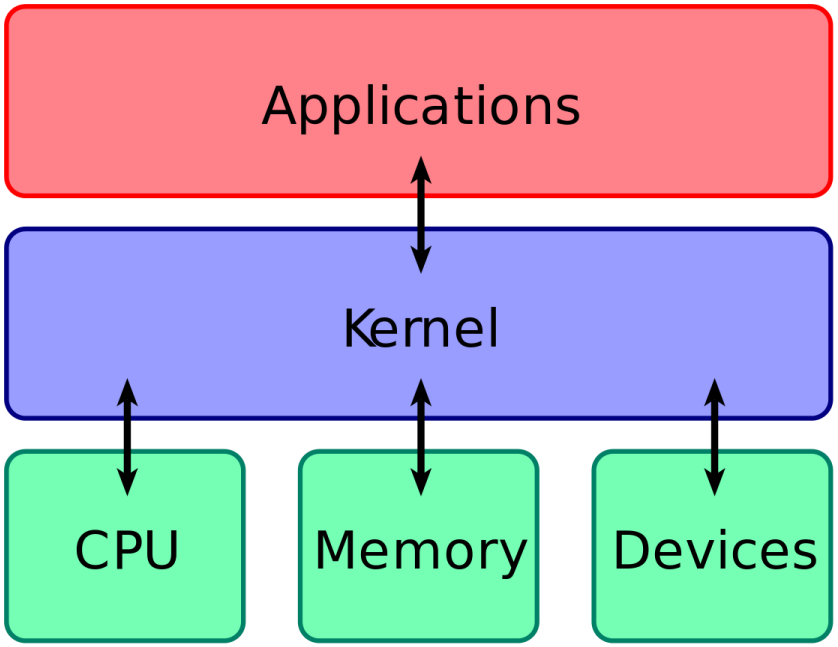


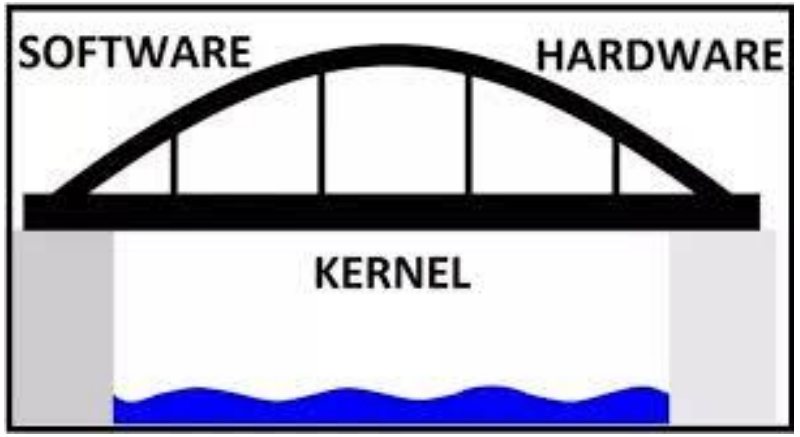
- It is impossible for every application programmer to understand every detail
- Managing all these components requires a layer of software – the **operating system**

What is an Operating System?



What is an Operating System?





What is an Operating System?

- An OS is a software which acts as an interface between the end user and computer hardware.

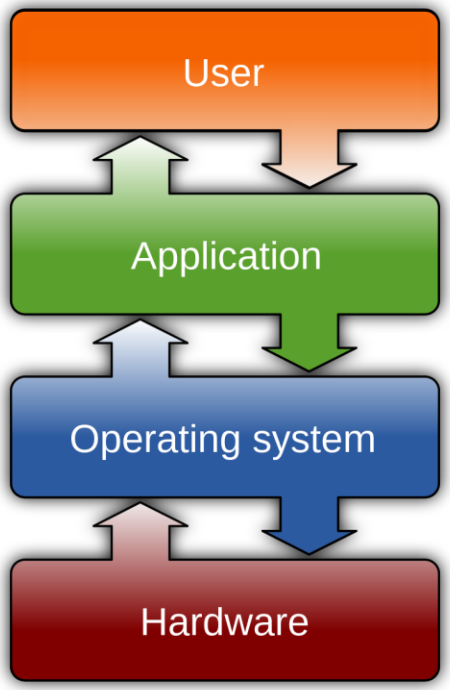
- Every computer must have at least one OS to run other programs.
  - An application like Chrome, MS Word, Games, etc. needs some environment in which it will run and perform its task.

- The OS helps you to communicate with the computer without knowing how to speak the computer's language. It is not possible for the user to use any computer or mobile device without having an OS.

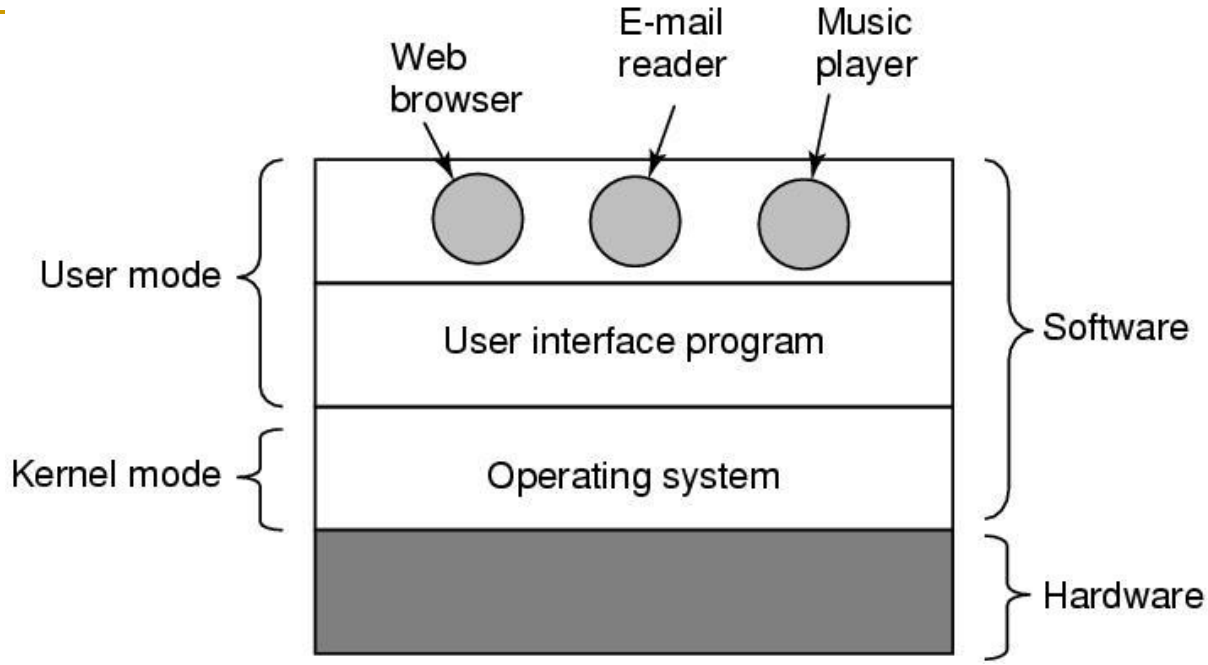
What is an Operating System?

Many definitions

- An OS is a software which acts as an interface between the end user and computer hardware.
- An OS is a software program that enables the computer hardware to communicate and operate with the computer software.
- An OS is *system* software that manages computer hardware, software resources, and provides common services for computer programs.

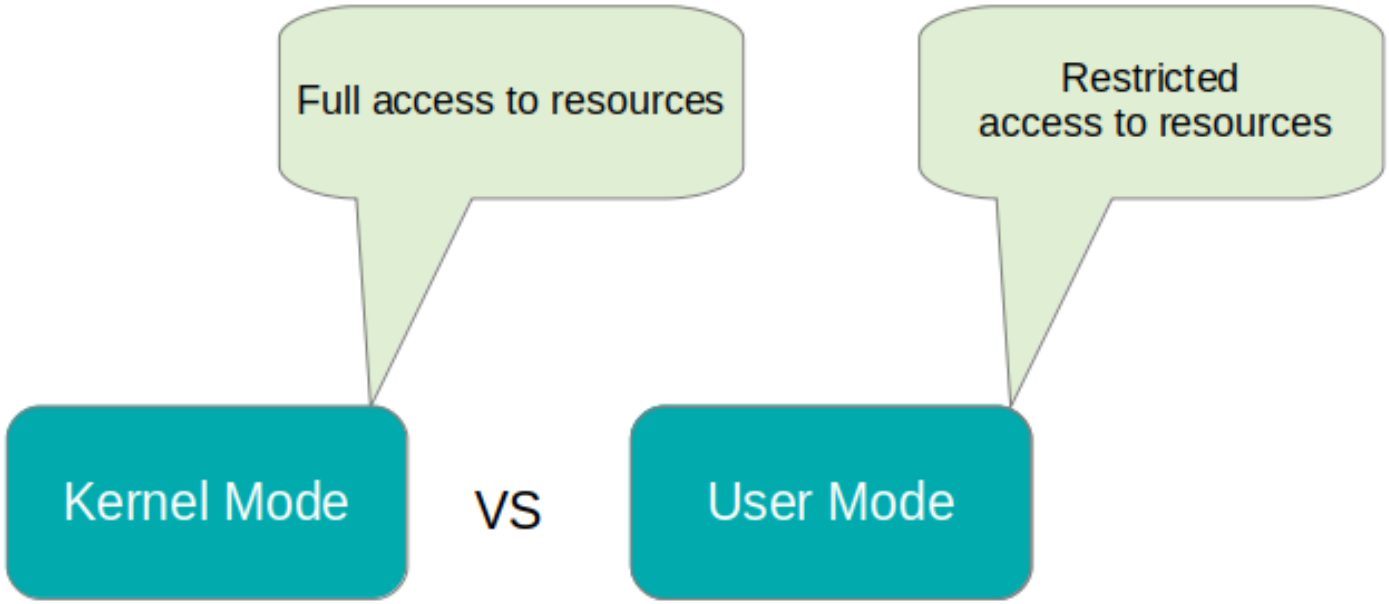


Kernel vs user mode

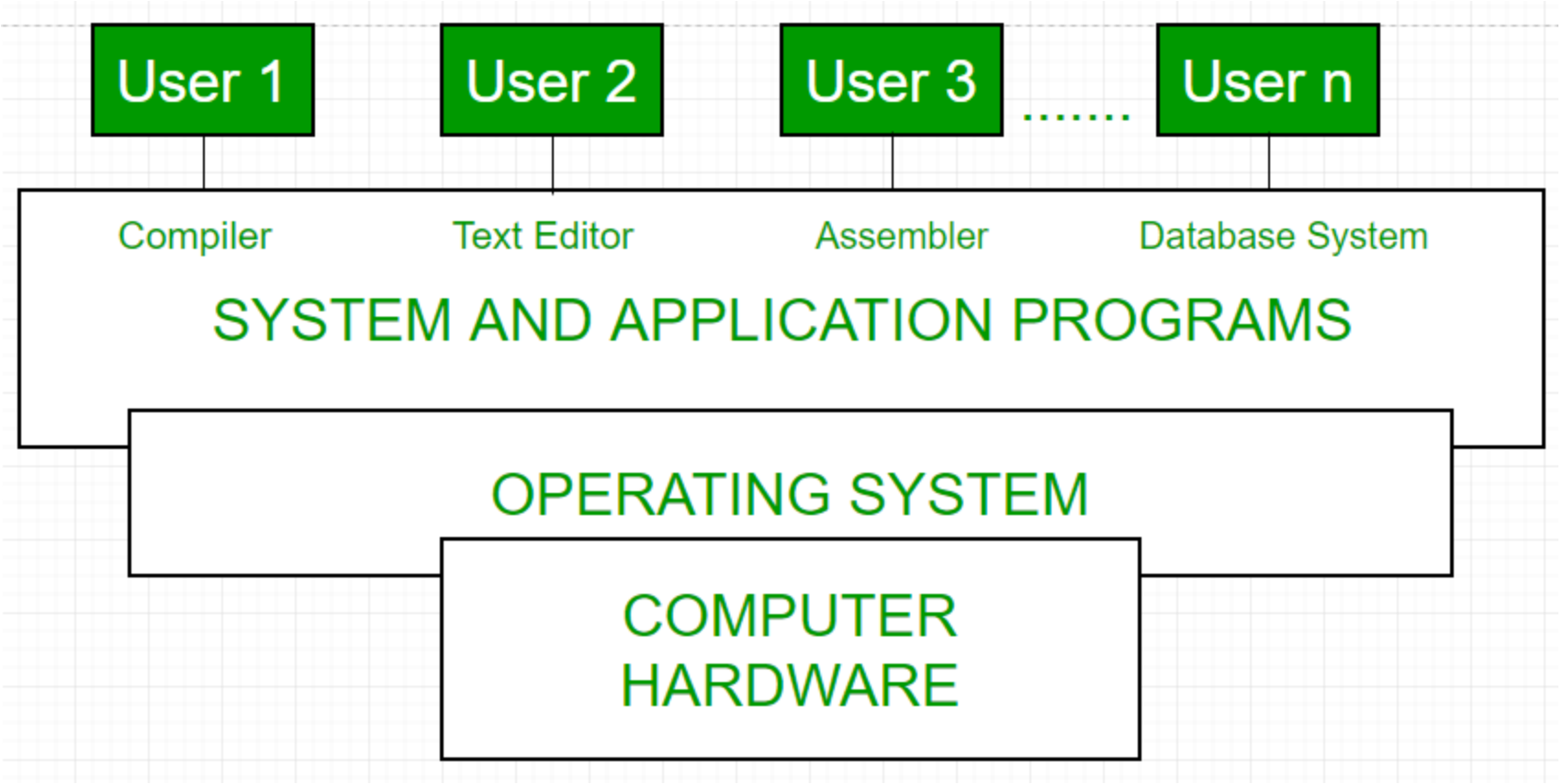


- Kernel mode (also called the supervisor mode):
  - OS runs in kernel mode
  - Has complete access to all hardware and can execute any instruction the machine is capable of executing.
- User mode:
  - User programs and application softwares (e.g. MS Word) run in user mode
  - Has limited capability – only a subset of machine instructions are available.

Kernel vs user mode



OS vs. User Programs



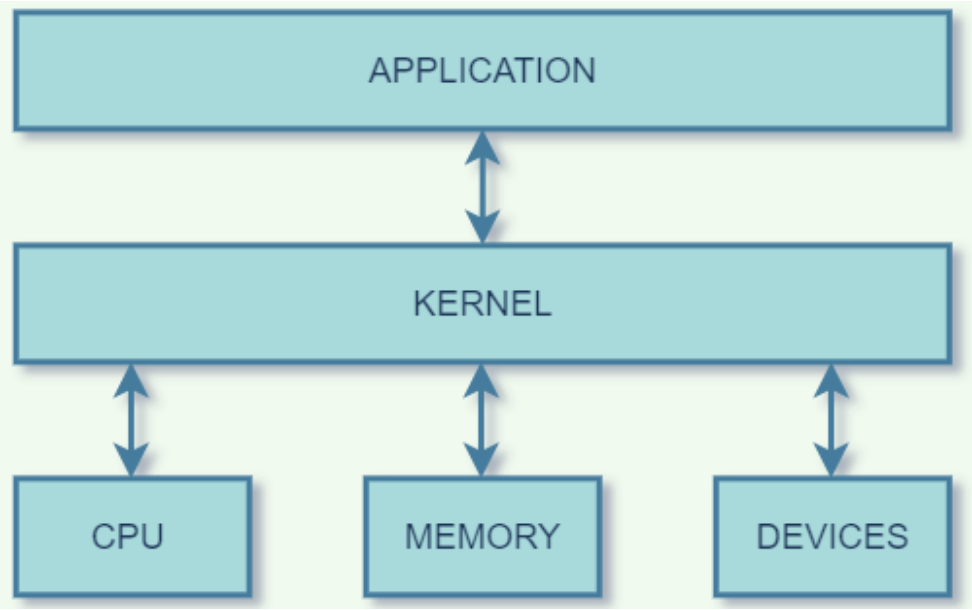
OS vs. User Programs

- OS
  - A software runs in kernel mode
  - Designed to provide general services required by user programs and to manage hardware resources (e.g., read, write)
  - Acts as an intermediary between programs and hardware
  - OS is huge, complex and long-lived
    - Huge – Windows is over 70 millions of code
    - Complex – has to manage hw, provide common services
    - Long-lived – OS is hard to write, needs lots of efforts, so nobody wants to throw away once it is written
- User programs
  - A software runs in user mode
  - Designed to perform a group of related functions for the benefit of the user. E.g. MS Word is for word processing
  - Runs on top of OS, uses services from OS
  - Not complex and huge like OS, and some programs may be modified by the user, but the OS cannot be modified.

Two main functions of OS

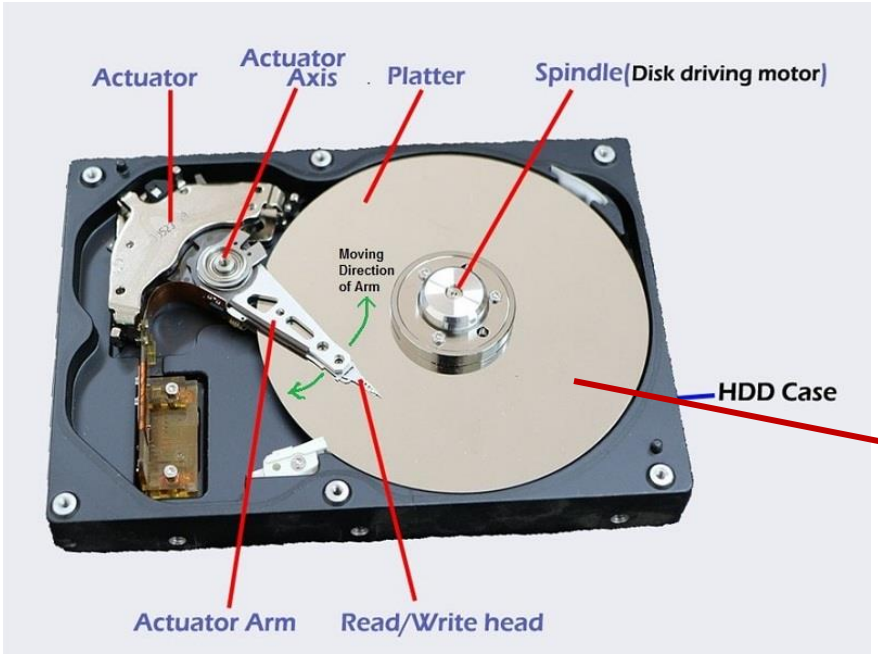
1. Provide application programmers a clean abstract set of resources instead of hardware ones

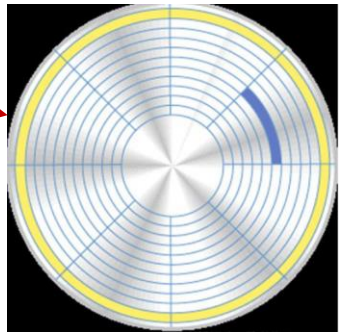
2. Manage the hardware resources



Abstractions

- Abstraction helps to deal with complexity
  - Hide lower-level detail

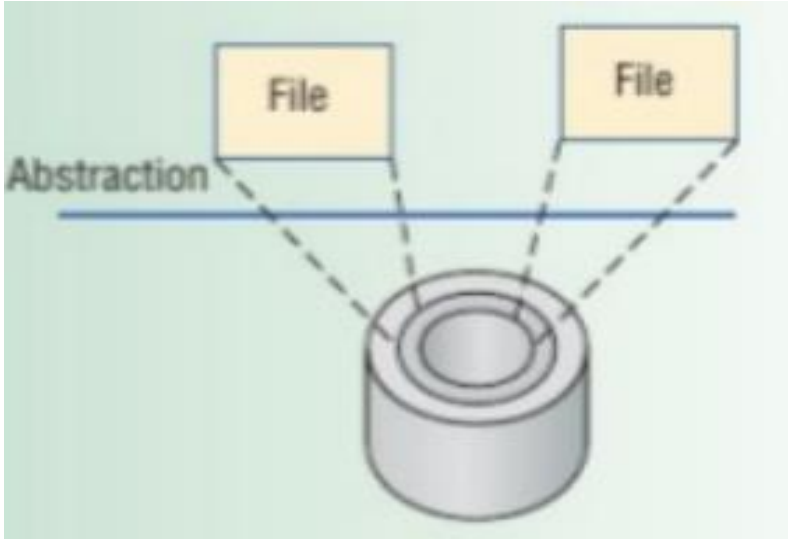




Platters spin at 3600-7200 rpm. The arm moves back and forth from the inside to outside of the platter at 50 times per second. This allows the hard drive to write data in **tracks** (yellow), and **sectors** (blue).

Abstractions

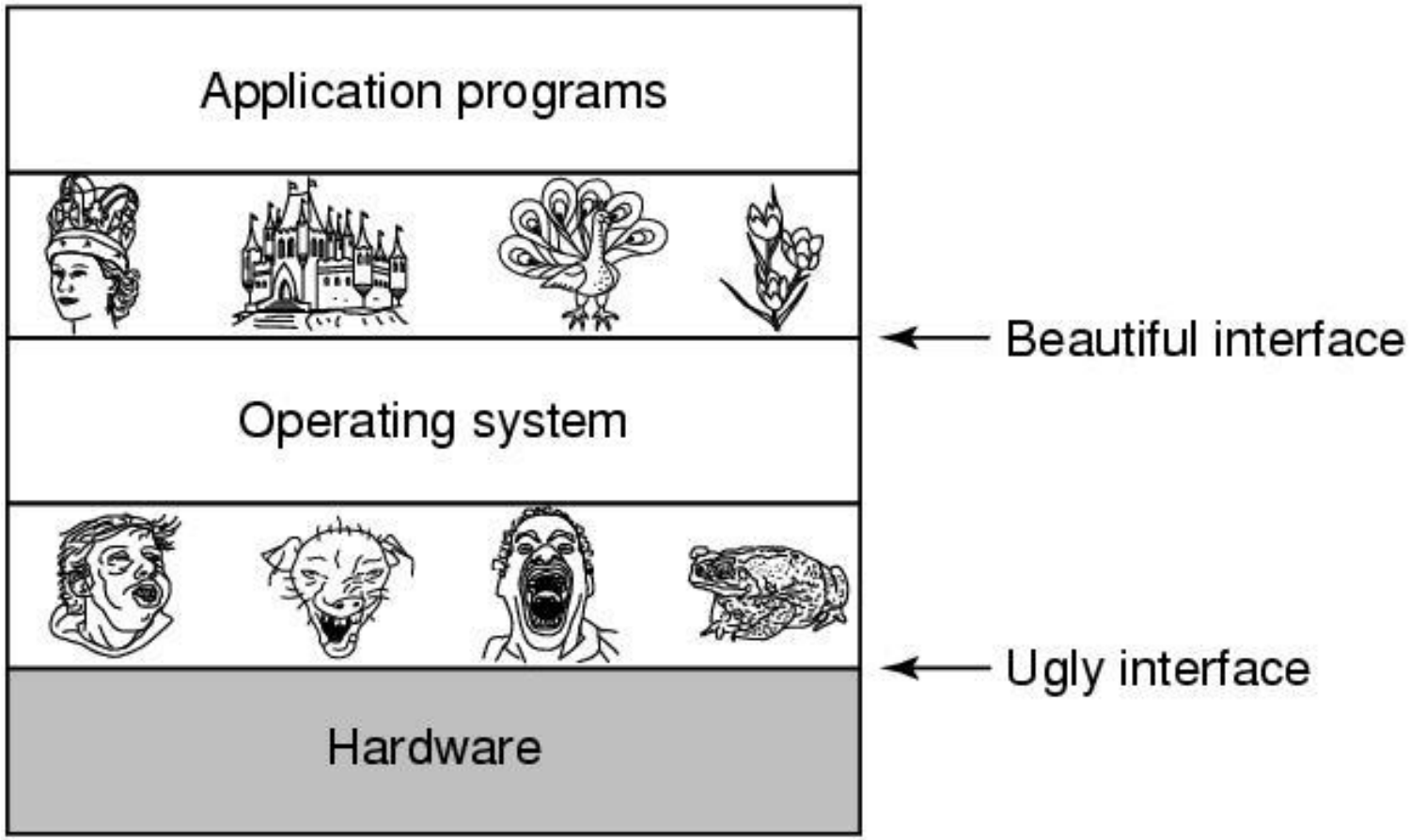
- Abstraction helps to deal with complexity
  - Hide lower-level detail



  - Programmer/user does not worry about hard-disk addressing details (sectors, tracks, etc.)
  - It appears to application software as a variable sized files
  - User can create, write and read files without knowing the underneath details.

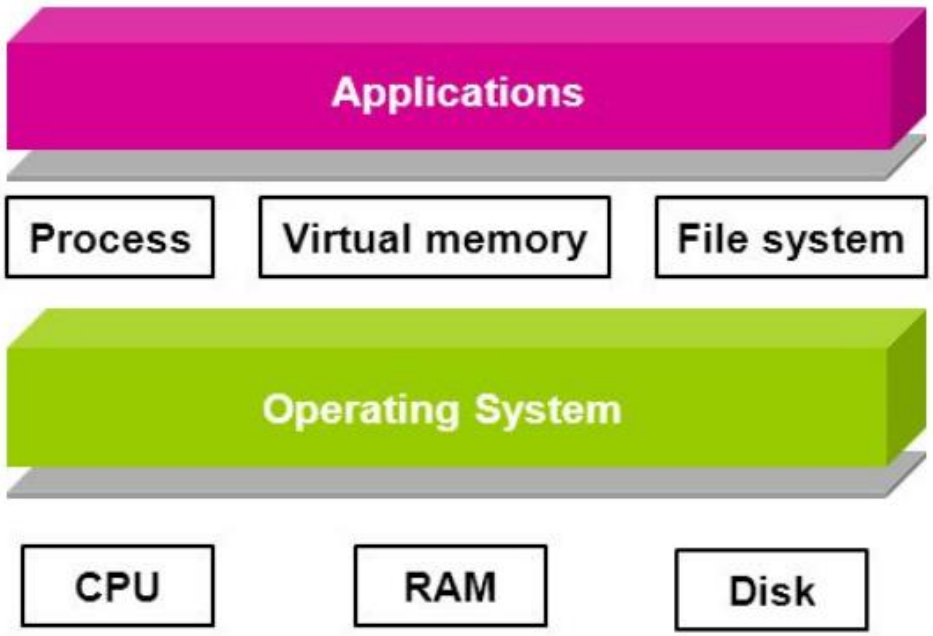
Abstraction

One of the major tasks of OS is to hide the hardware and present programs (and their programmers) with nice, clean, elegant, consistent, abstractions to work with.



Abstraction

- CPU - process
- Storage - files
- Memory - address space



Abstraction – real world example

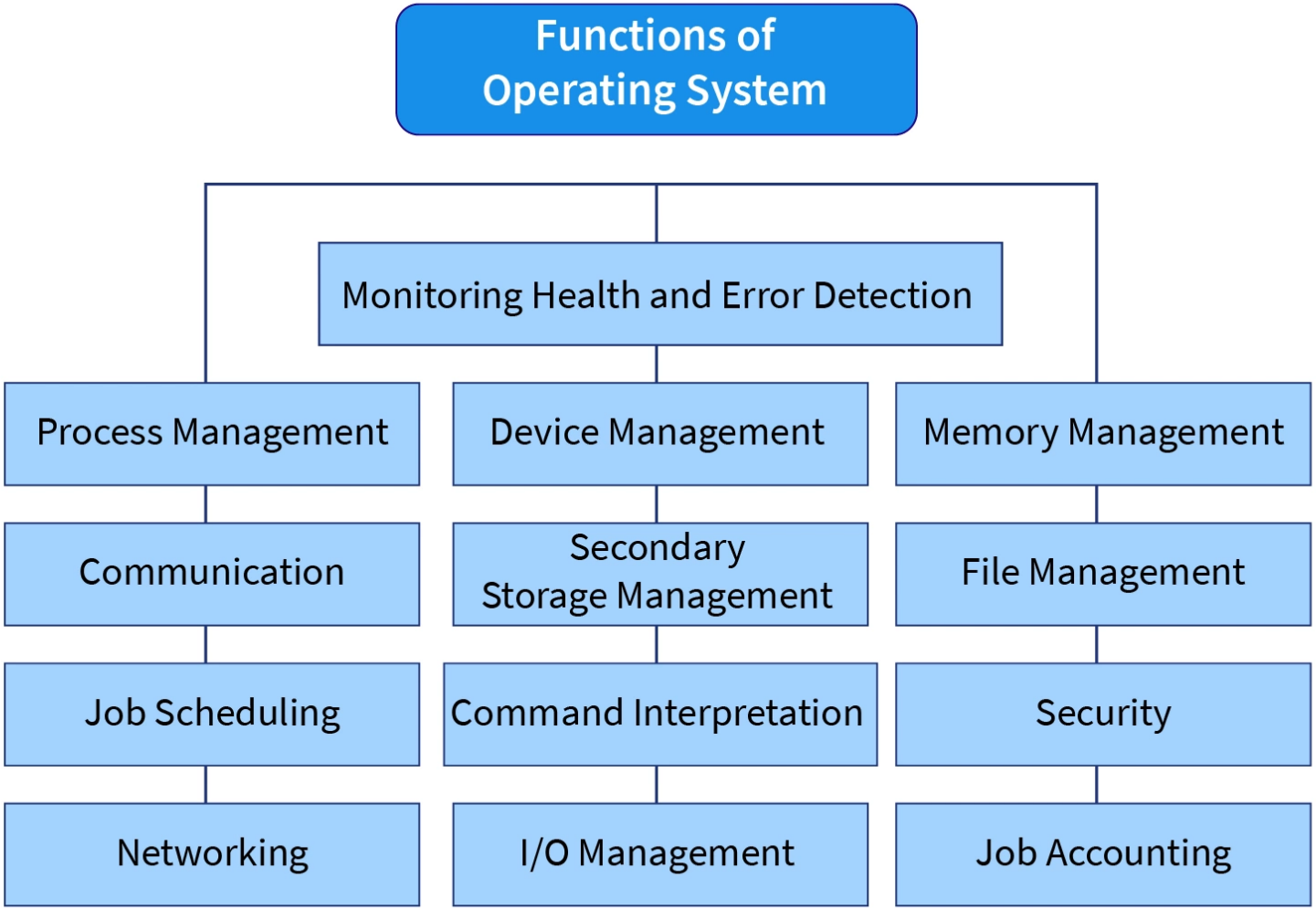
- Making coffee with a coffee machine is a good example of abstraction.
- You need to know how to use your coffee machine to make coffee. You need to provide water and coffee beans, switch it on and select the kind of coffee you want to get.
- The thing you don't need to know is how the coffee machine is working internally to brew a fresh cup of delicious coffee. You don't need to know the ideal temperature of the water or the amount of ground coffee you need to use.

Real Life Example of Abstraction

1.1 summary

- Operating system is a software
  - Is a complex software
  - Runs in kernel mode
  - Manages hardware
  - Provide a friendly interface for application programmer

1.2. Services provided by OS



1.2. Services provided by OS

1. Program development
   - provide facilities like editors and debuggers to assist programmers to create programs
2. Program execution
   - system capability to load a program into memory and to run it
   - OS handles scheduling duties for user
3. Access to I/O devices
   - The OS provides a uniform interface to the user that hides the details of each I/O device that requires its own set of instructions or control signals for operation.

Services provided by OS

4. Controlled access to files
   - Program capability to read, write, create and delete files.
   - The OS may provide protection mechanisms to control access to the files in the case of a system with multiple users.

5. System Access
   - In the case of a shared or public system, the OS provides protection of resources and data from unauthorised users and resolves conflicts for resource allocation.

6. Error detection and response
   - Ensure correct computing by detecting errors in the CPU and memory h/w, in I/O devices, or in user programs.
   - E.g.
     - Internal and external h/w errors like memory errors and device failures
     - Software errors like arithmetic overflow and access forbidden memory locations

Services provided by OS

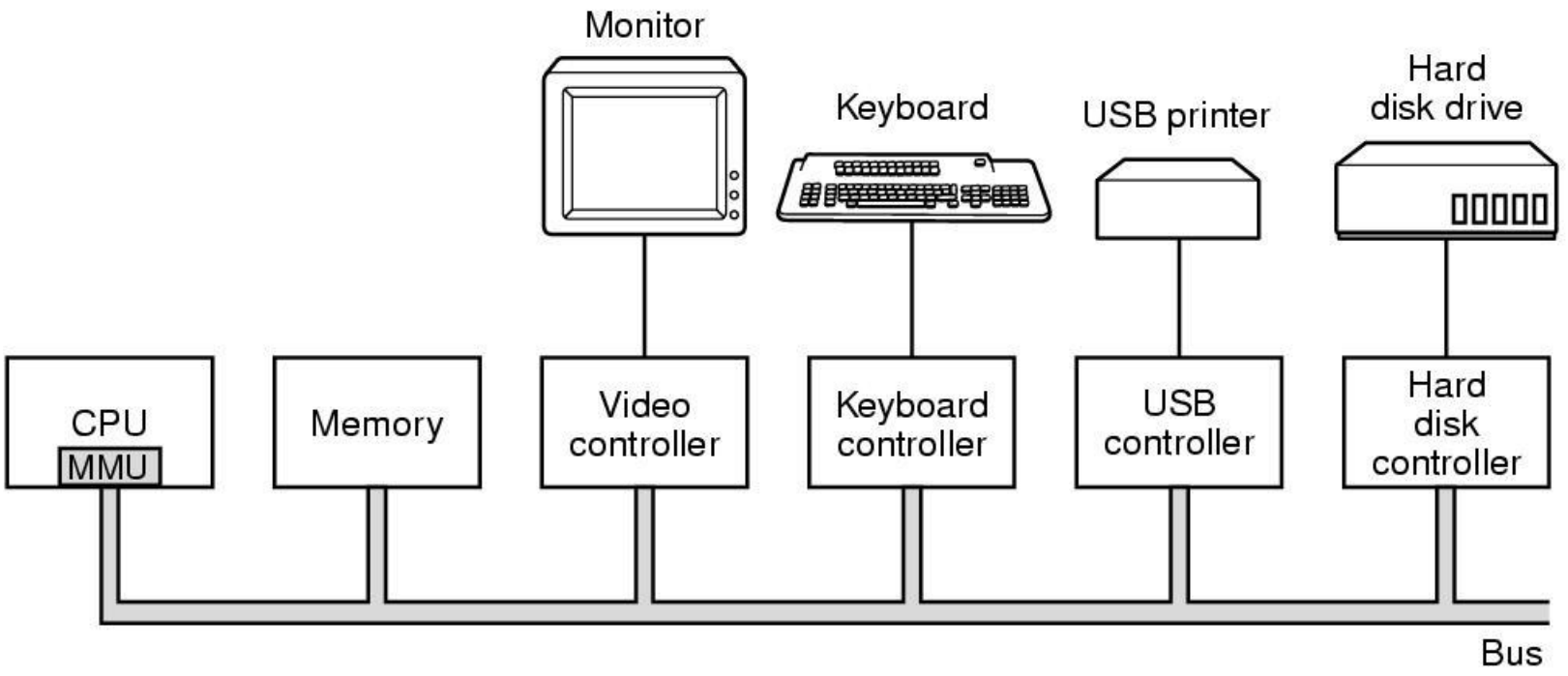
7. Accounting

   Keep track of and record which users use how much and what kinds of computer resources for account billing or for accumulating usage statistics and monitor performance.

8. Resource Allocation

   Allocating resources to multiple users or multiple jobs running at the same time

OS as a resource manager



Some of the components of a simple personal computer. OS is responsible for managing all of these.

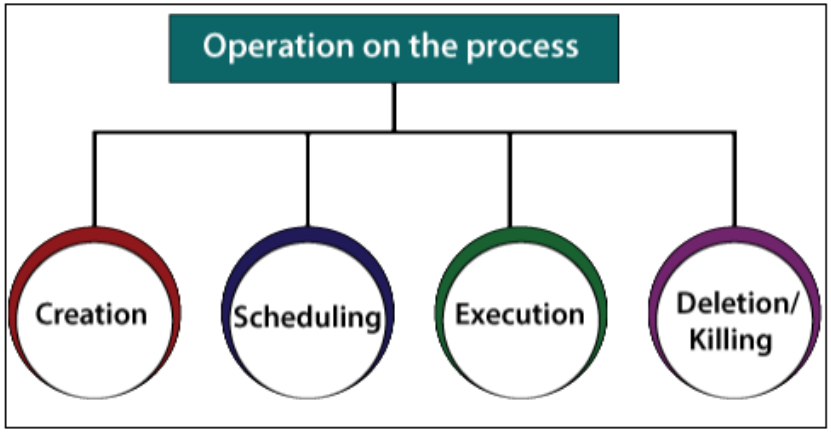
OS as a resource manager

OS is responsible for

- Process management
- Main memory management
- File management
- I/O system management
- Secondary storage management

Process management

- A process is a program in execution.
- OS is responsible for the following activities with process management.
  - Process creation and deletion
  - Process suspension and resumption
  - Provide mechanism for process synchronization, process communication.



Memory management

- Main memory is a volatile storage medium.
  - It loses its contents in the case of system failure.
- The OS is responsible for the following activities in connections with memory management.
  - Keep track of which parts of memory are currently being used and by whom.
  - Decide which processes are to be loaded into memory when memory space becomes available.
  - Allocate and de-allocates memory space as needed.

File management

- A file is a collection of related information defined by its creator.
- File is an abstraction of physical storage media.
- OS is responsible for the following activities in connections with file management.
  - The creation and deletion of files.
  - The creation and deletion of directories.
  - The mapping of files into secondary storage.
  - The backup of files on stable (non volatile) storage media.

Secondary storage management

- Since the main memory (primary storage) is volatile and too small to accommodate all data and programs permanently, the computer system must provide secondary storage to backup the main memory.
- Most modern computer systems use hard disks as the principal storage medium for both programs and data. Hence, the proper management of disk storage is of central importance to a computer system.
- The OS is responsible for the following activities in connection with disk management:
  - Free-space management.
  - Storage allocation.
  - Disk scheduling.