Unit 1 – Overview of OS

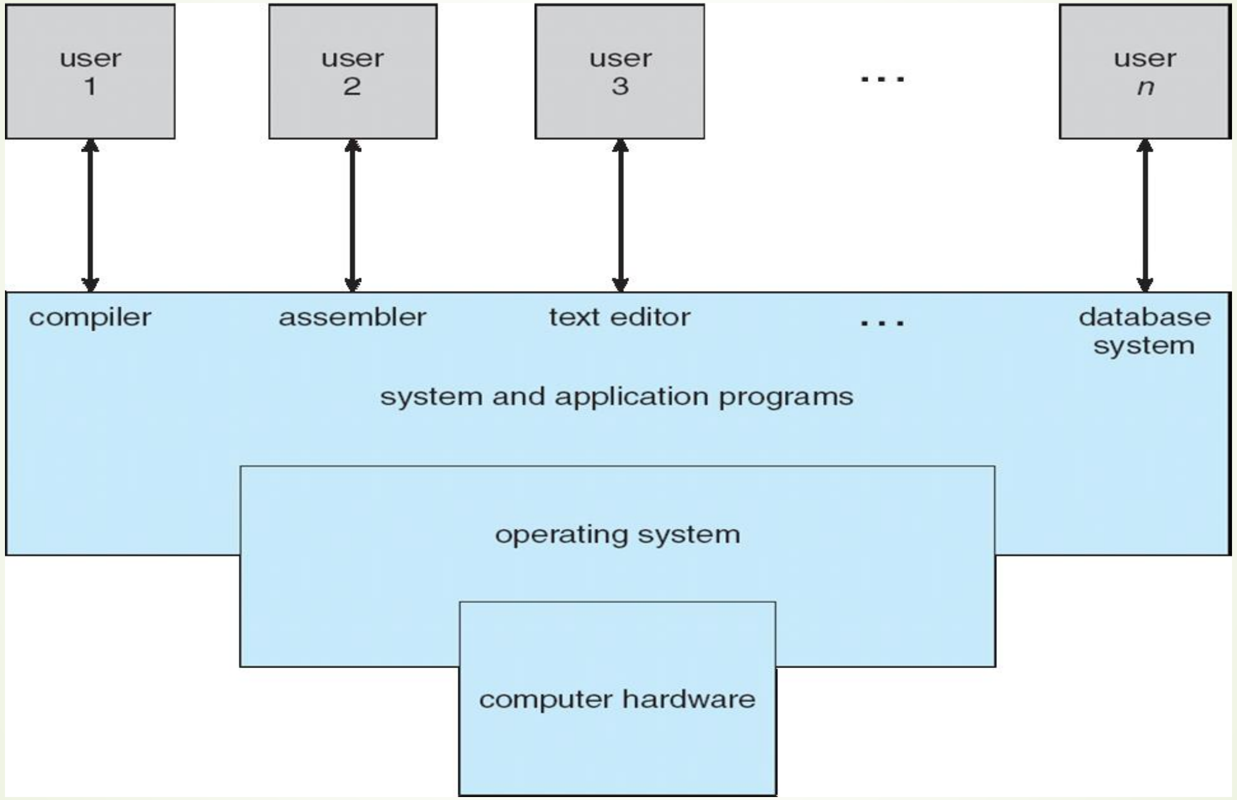
- Computer System
- Introduction to OS
- Two Views/aspects of OS
- Evolution/History of OS
- Types of OS
- Structure of OS
- System Calls
- Handling System Calls
- Types of System Calls
- The Shell
- Open Source OS

Computer System

Computer system can be divided into four components:

- Hardware – provides basic computing resources
  - CPU, memory, I/O devices
- Operating system
  - Controls and coordinates use of hardware among various applications and users
- Application programs – define the ways in which the system resources are used to solve the computing problems of the users
  - Word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

Computer System (contd.)



Four Components of a Computer System

Operating System

- A program that acts as an intermediary between a user of a computer and the computer hardware.

- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Aspects of OS

- OS is a **resource manager**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

- OS is a **extended machine**
  - Controls execution of programs to prevent errors and improper use of the computer.
  - Hides the complexity of the hardware and present simple and nice view to user.

Evolution of OS

- First Generation Computers (1945-55)
  - Vacuum-tubes as building machine
  - Single group of people designed, built, programmed, operated and maintained each machine
  - Machine Code is used
  - OS and programming language were unknown

- Second Generation Computers (1955-65)
  - Transistors as building machine
  - Batch System were used
  - FORTAN and assembly language were used

Evolution of OS (contd.)

- Third Generation Computers (1965-80)
  - Integrated Circuit (IC) as building machine
  - Multiprogramming came into existence
  - OS: OS/360, MULTICS, CTSS were used

- Fourth Generation of Computers (1980-Present)
  - LSI and VLSI as building machine
  - Microcomputers came into existence
  - IBM PC (MS-DOS), Apple Macintosh (Macintosh)

Types of OS

- Mainframe OS
- Server OS
- Multiprocessor OS
- PC OS
- Real time OS

and so on ...

Types of OS (contd.)

Mainframe OS

- Capable of processing large amounts of information and supporting a great number of users
- Used generally in support of government and business projects

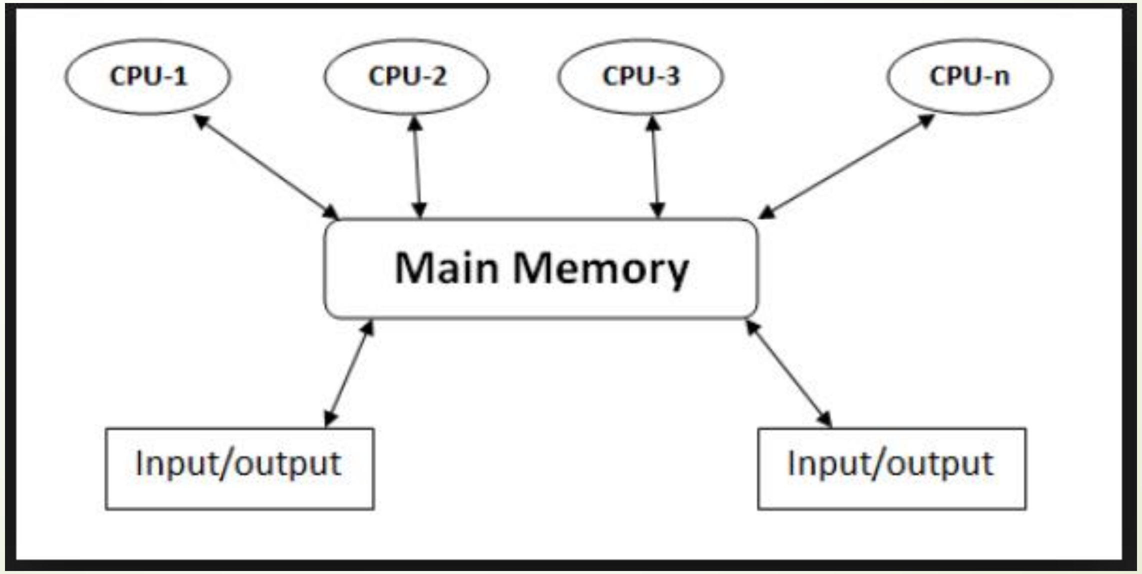
Server OS

- Developed for multi-user computer programs, programs important for business computing and networked application

Types of OS (contd.)

Multiprocessor OS

- Within a single computer system, the usage of multiple central processing units (CPUs) is known as multiprocessor operating system. Such systems are used to process large volumes of data at a very high speed.



Types of OS (contd.)

Personal Computer (PC) OS

- An operating system serving the general purpose (word processing, spreadsheets, web browser, digital media and others) of the end user is known as a personal computer operating system

Real Time OS

- The operating systems for serving real time application requests is known as real time operating systems
- The real time operating system should be capable of responding and serving requests as and when they come in

Types of OS (contd.)

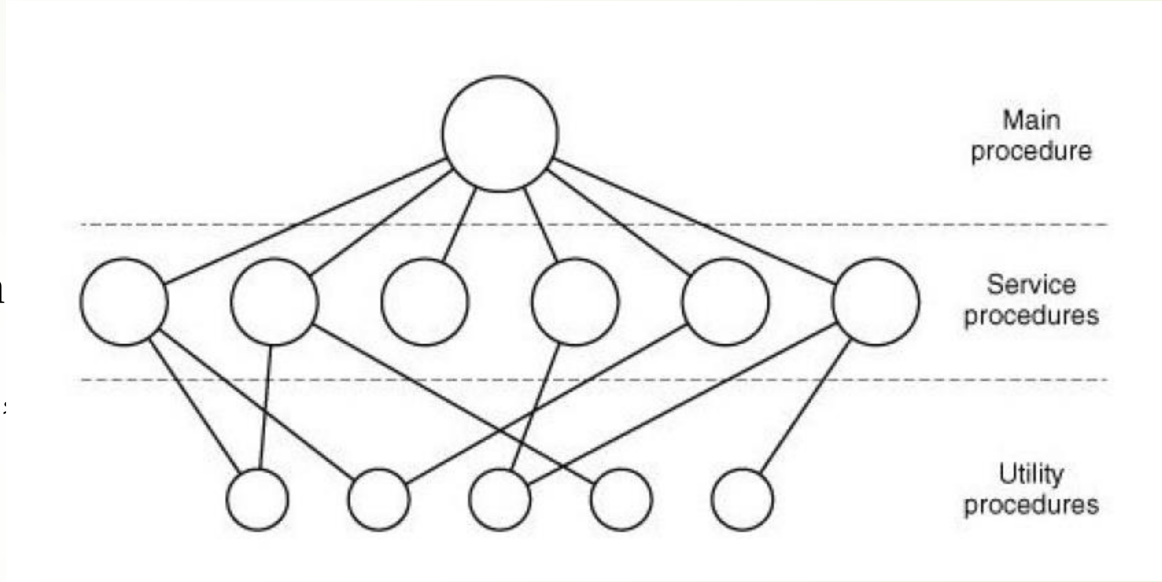
- **Multiprogramming** (**Batch system**) needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via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

- **Timesharing** (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory ⇨**process**
  - If several jobs ready to run at the same time ⇨ **CPU scheduling**
  - If processes don′t fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory

Structure of OS (contd.)

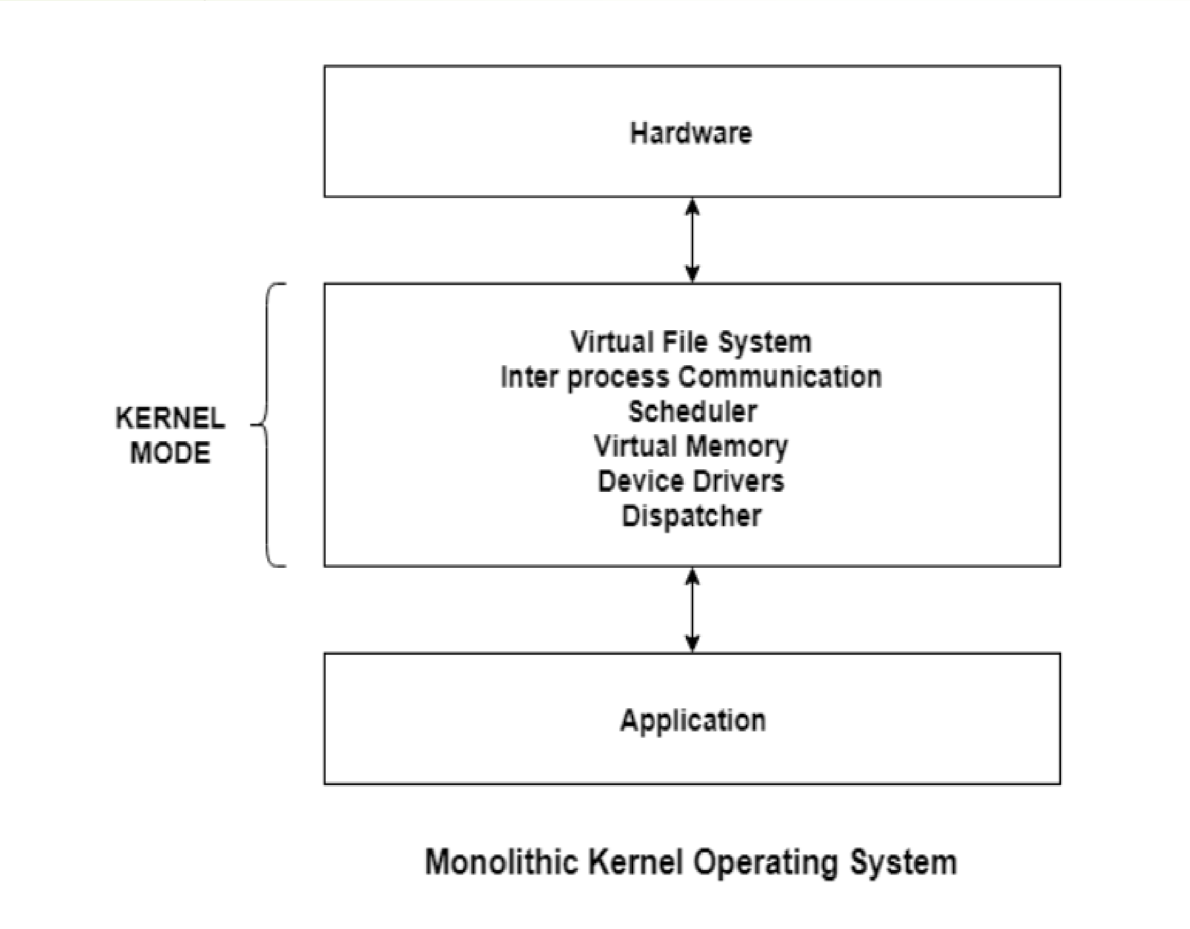
- **Monolithic Structure**

  - The components of monolithic operating system are organized haphazardly and any module can call any other module without any reservation.
  - Similar to the other operating systems, applications in monolithic OS are separated from the operating system itself. That is, the operating system code runs in a privileged processor mode (referred to as kernel mode), with access to system data and to the hardware.
  - applications run in a non-privileged processor mode (called the user mode), with a limited set of interfaces available and with limited access to system data.
  - The monolithic operating system structure with separate user and kernel processor mode is shown in Figure.
  - The operating system is written as a collection of procedures, each of which can call any of the other ones whenever it needs to.
  - Examples: **CP/M and MS-DOS**



Structure of OS (contd.)

- **Monolithic Structure**



Monolithic Kernel Operating System

Structure of OS (contd.)

Layered Systems (Simple Structure)

Layer	Functions
5	The operator
4	User Programs
3	Input/Output Management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

- The layered approach consists of breaking the operating system into the number of layers(level), each built on the top of lower layers. The bottom layer (layer 0) is the hardware layer; the highest layer is theuser interface.
- The main advantages of the layered approach is modularity.
- Example Systems: VAX/VMS, Multics, UNIX

Structure of OS (contd.)

- Microkernels
- Client-Server Model
- Virtual Machines
- Exo-Kernels: The exokernel architecture is designed to separate resource protection from management to facilitate application-specific customization.

System Calls

- A way of programs to interact with the OS
- Programmatic way to request a service from the kernel of the OS it is executed on

- Services Provided by System Calls :
  - Process creation and management
  - Main memory management
  - File Access, Directory and File system management
  - Device handling(I/O)
  - Protection
  - Networking, etc.

System Calls (contd.)

TYPES	WINDOWS	UNIX

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

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

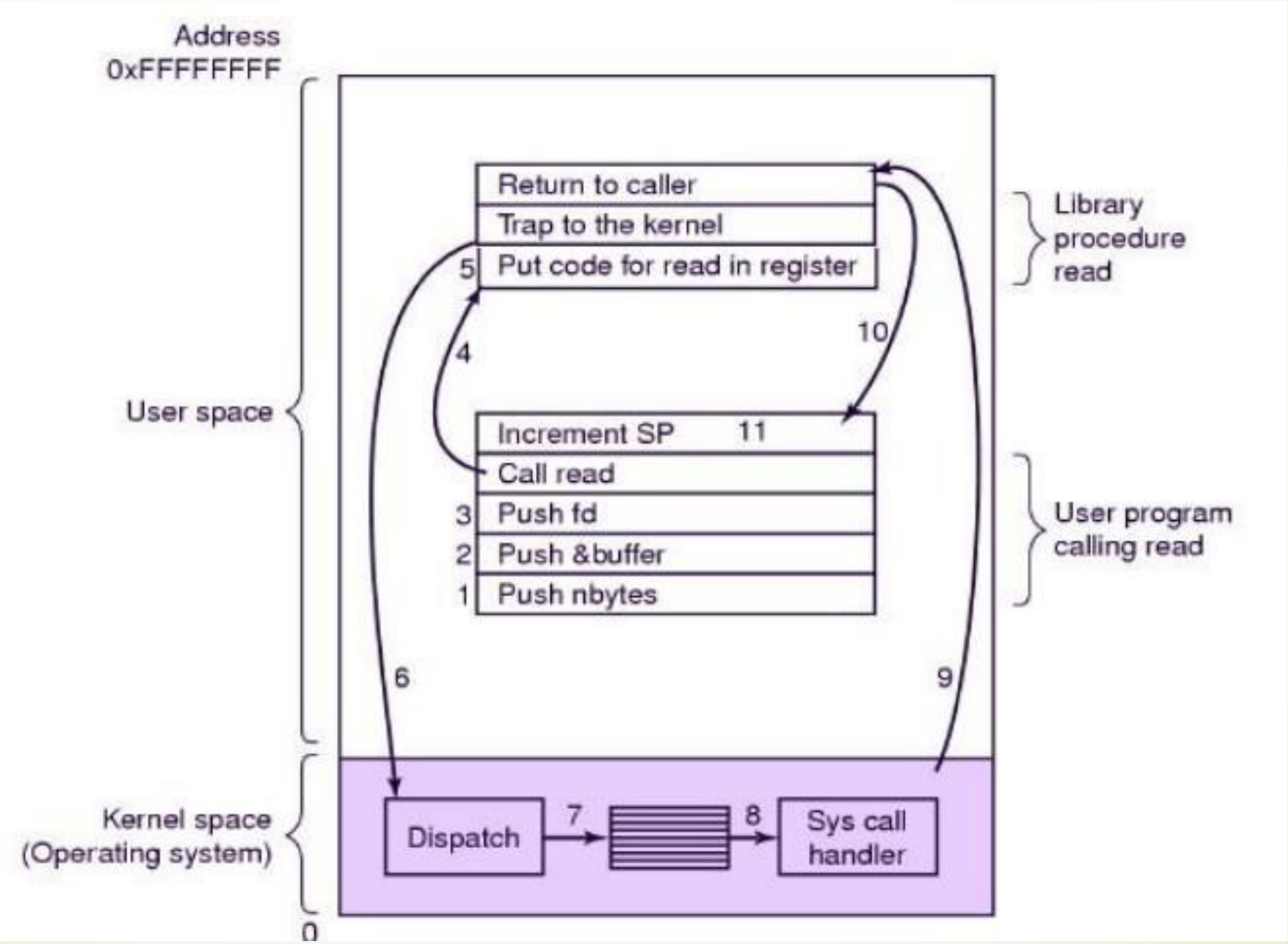
Device Manipulation	SetConsoleMode()
	ReadConsole()
	WriteConsole()
		ioctl()
		read()
		write()

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

Communication	CreatePipe()
	CreateFileMapping()
	MapViewOfFile()
		CreatePipe()
		CreateFileMapping()
		MapViewOfFile()

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

Handling System Calls



Handling System Calls (contd.)

- System calls performed in a series of steps. Most of the system calls are invoked as the following example system call: read

  count = read(fd, buffer, nbytes)

- Push parameters into the stack (1-3)
- Calls library procedure (4)
- Pass parameters in registers.(5)
- Switch from user mode to kernel mode and start to execute.(6)
- Examines the system call number and then dispatches to the correct system call handler via a table of pointer.(7)
- Runs system call handlers (8).
- Once the system call handler completed its work, control return to the library procedure.(9)
- This procedure then return to the user program in the usual way. (10)
- Increments Stack Pointer (SP) before call to finish the job. (11)

The Shell

- The outermost layer in an operating system
- A computer application software that provides the services of an operating system to an external user or another program
- Shell may be off either CLI or GUI
- Executes programs based on the input provided by the user

Open Source Operating System

- Operating systems made available in source-code format rather than just binary closed-source
- Counter to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has "copyleft" GNU Public License (GPL)
- Examples include GNU/Linux and BSD(Berkeley Software Distribution) UNIX (including core of Mac OS X), and many more
- Can use VMM like VMware Player (Free on Windows), VirtualBox (open source and free on many platforms - http://www.virtualbox.com)
- Use to run guest operating systems for exploration

What OS Do

- Depends on the point of view
- Users want convenience, ease of use, and good performance
  - Don't care about resource utilization
- But shared computers such as mainframe or minicomputers must keep all users happy
- Users of dedicated systems such as workstations have dedicated resources but frequently use shared resources from servers
- Handheld computers are resource-poor, optimized for usability and battery life
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

System Program

- System programs provide a convenient environment for program development and execution. They can be divided into:
  - File manipulation
  - Status information is sometimes stored in a File modification
  - Programming language support
  - Program loading and execution
  - Communications
  - Background services
  - Application programs

Common OS Concepts

- Process Management
- Memory Management
- File System Management

Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a ***passive entity***, process is an ***active entity***.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

File-System Management

- Files usually organized into directories
- Access control on most systems to determine who can access what
- OS activities include
  - Creating and deleting files and directories
  - Primitives to manipulate files and directories
  - Mapping files onto secondary storage
  - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)