System Components – Process Management

- A **process** is a program in execution. It is a unit of work within the system. A program is a *passive entity*, a process is an *active entity*.
- A process needs resources to accomplish its task
  - E.g. CPU, memory, I/O devices, files
- Process termination requires reclaiming of any reusable resources
- A *single-threaded process* has one program counter specifying location of next instruction to execute
  - A process generally executes instructions sequentially until completion
- A *multi-threaded process* has one program counter per thread
- Typically a system has many processes, some user, some operating system running concurrently on one or more CPUs

System Components – Process Management

- The operating system is responsible for the following activities in connection with process management.
  - Process creation and deletion
  - process suspension and resumption
  - Provision of mechanisms for:
    - process synchronization
    - process communication
    - deadlock handling

System Components – Main Memory Management

- Memory is a large array of words or bytes, each with its own address.
- Main memory is a *volatile* storage device.
- The operating system is responsible for the following activities in connection with memory management:
  - Keep track of which parts of memory are currently being used and by whom.
  - Decide which processes to load when memory space becomes available.
  - Allocate and deallocate memory space as needed

System Components – File Management

- A file is a collection of related information defined by its creator.
- The operating system is responsible for the following activities in connection with file management:
  - File/directory creation and deletion
  - Support of primitives for manipulating files and directories
  - Access control available on most systems
  - Mapping files onto secondary storage
  - File backup on stable storage media

System Components – Secondary Storage Management

- Since main memory (*primary storage*) is volatile and too small to accommodate all data and programs permanently, the computer system must provide *secondary storage* to back up main memory.
- Most modern computer systems use disks as the principle storage medium, for both programs and data.
- The operating system is responsible for the following activities in connection with disk management:
  - Free space management
  - Storage allocation
  - Disk scheduling

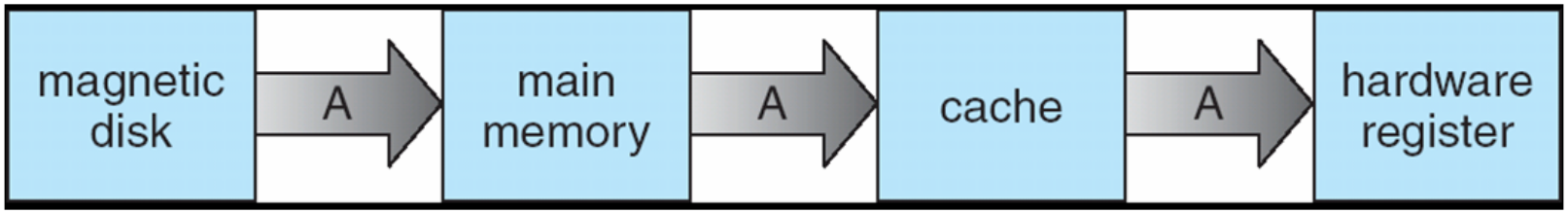
Performance of Various Levels of Storage

- Movement between levels of storage hierarchy can be explicit or implicit

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

Migration of Integer A from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies of a datum can exist

System Components – I/O System Management

- One purpose of OS is to hide peculiarities of hardware devices from the user

The I/O system consists of:

- A buffer-caching system
- A general device-driver interface
  - a set of interrupt handlers for a device controller
- Drivers for specific hardware devices

Protection and Security

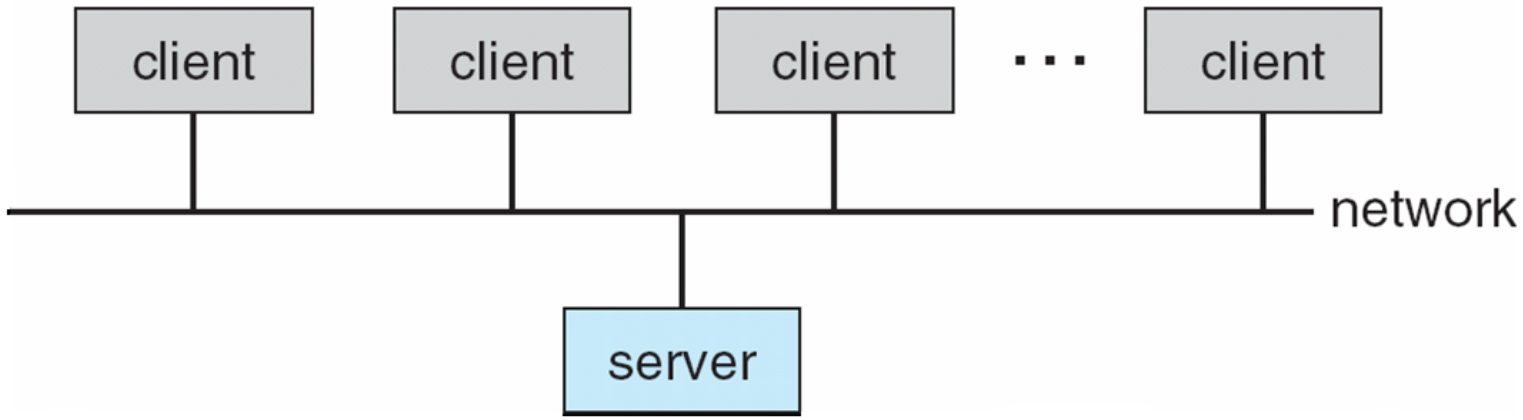
- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - E.g denial of service, viruses, identity theft

Protection and Security

- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

Computing Environments

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server:** provides interface for client to access certain services (e.g database)
    - **File-server:** provides interface for clients to store and retrieve files



Peer-to-Peer Computing

- Another model for a distributed system
- P2P does not distinguish clients and servers
  - all nodes are considered peers
  - May each act as a client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via **discovery protocol**
  - Examples include *Napster* and *Gnutella*

Web-Based Computing

- Web has become ubiquitous
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers: **load balancers**
- Use of operating systems like Windows 95, client-side, have evolved into Linux, Mac OS/X and Windows 7/10, which can be clients and servers

Open-Source Operating Systems

- 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 UNIX* (including core of Mac OS X)