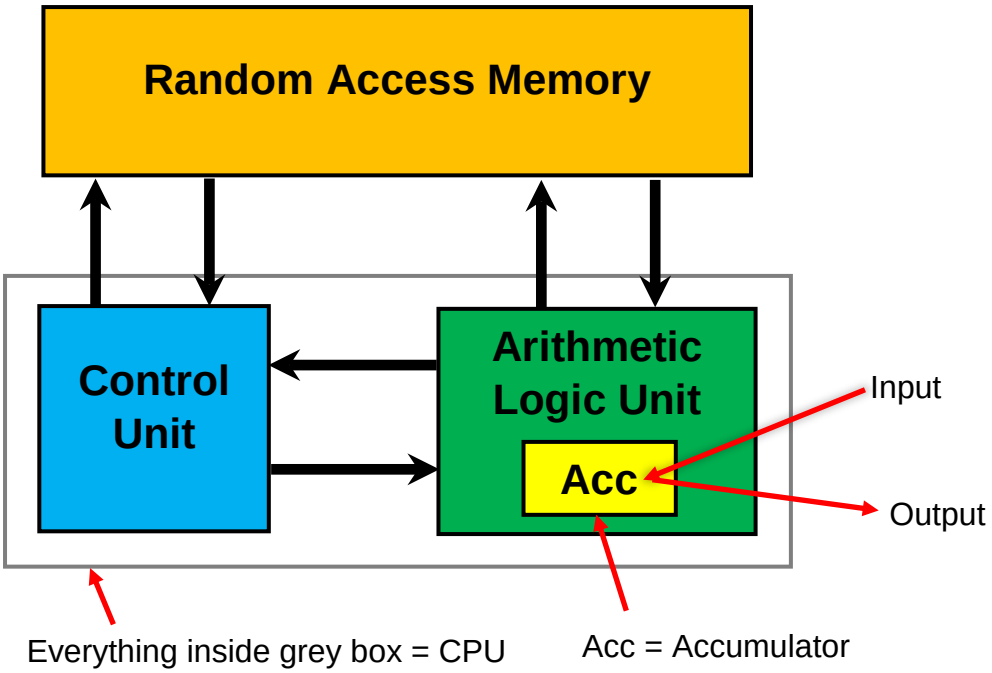
Von Neumann Architecture:



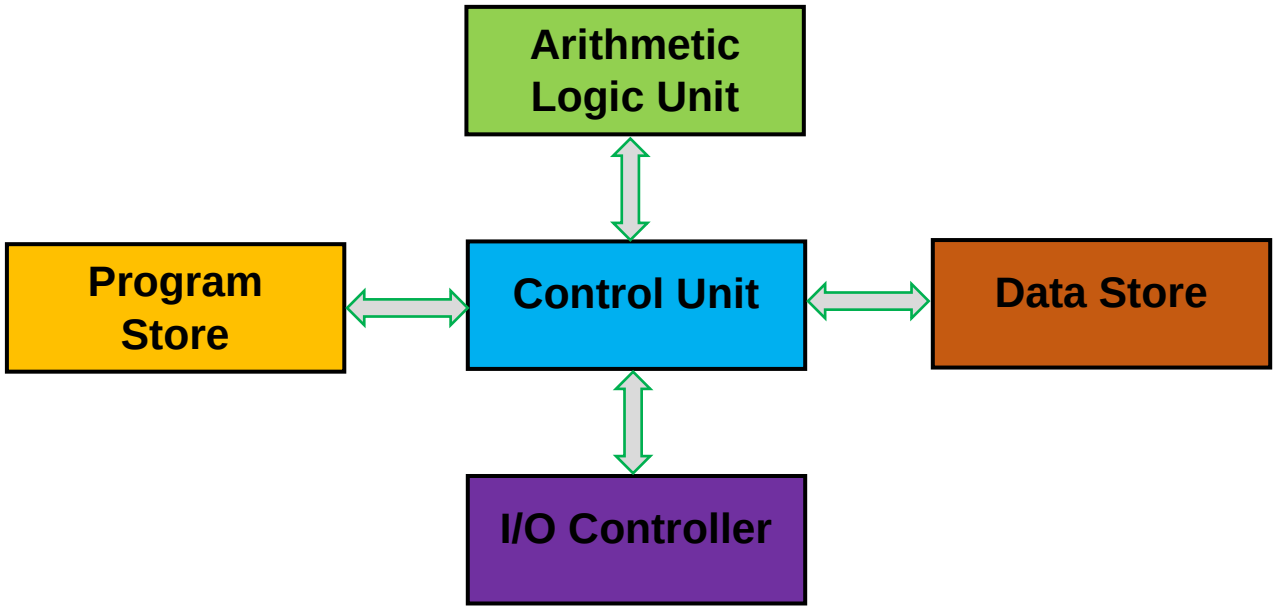
Limitations:

The Von Neumann Architecture uses 'Shared Store' meaning that both instructions and data must be the same length (in bits). It also means that it can only load instruction or data at once, not both simultaneously. This ultimately leads to a performance bottleneck.

Moore's Law:

Moore's law states that the number of transistors on an integrated circuit doubles every 2 years. So far, this prediction has held true.

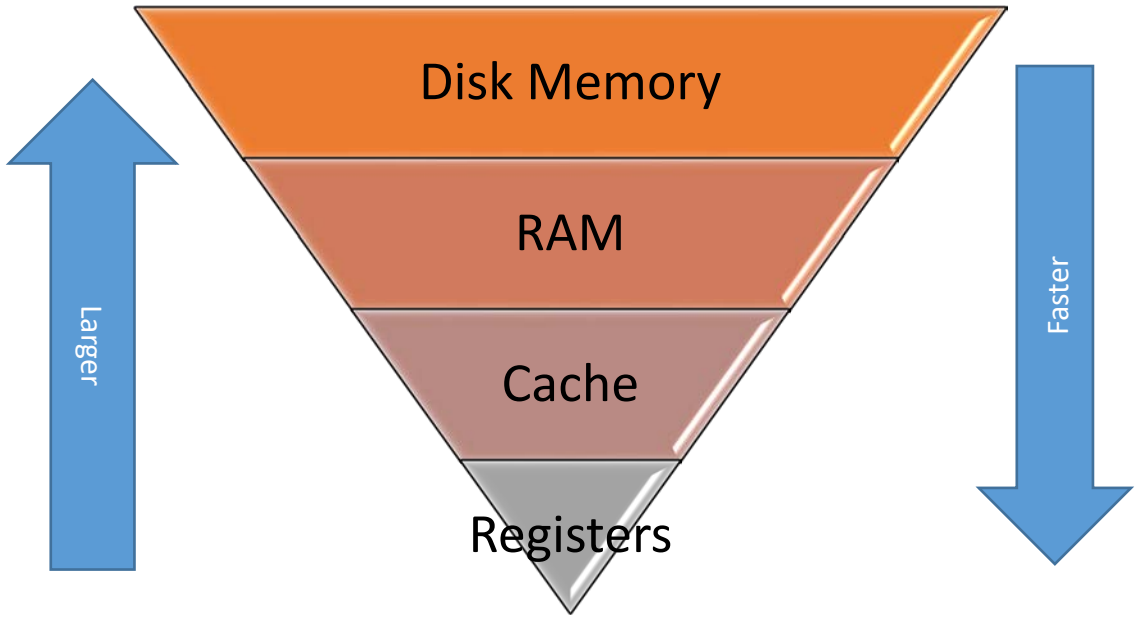
Harvard Architecture:



Modern computers tend to implement a modified version of Harvard Architecture. This includes an external storage (Disk/RAM) as a unified store. Within the processor there are separate pathways for instructions and data. Storage within the processor splits data and instructions, allow them to be loaded simultaneously.

Data Storage:

Memory Hierarchy



Registers:

- Registers are small blocks of memory within the Central Processing Unit (CPU).
- A CPU can have 8 or 16 registers per core.
- They store the data that the processer is currently working on. It will only perform operations on the data held in its registers.
- The CPU Transfers data in and out of the Random Access Memory (RAM).

Cache:

- Processers have on-board memory that contains recently used data and instructions.
- Many operations are repetitive loops. Therefore, the same data is often needed over and over again.
- Keeping a local copy of this data is faster than having to keep fetching the data from the RAM or Disk.

Cache Levels:

- There are 3 types of cache. Level 1 (L1), Level 2 (L2) & Level 3 (L3).
- **L1** is the ***fastest, most expensive*** and is very ***small.*** (L1 is closest to the CPU).
- **L2** is ***bigger, further away from the CPU***, and ***slower.*** (Relative to L1).
- **L3** is

Cache Synchronisation:

- If data is needed, the cache is checked first;
  - If present (***cache hit***), it is used straight away.
  - If absent (***cache miss***), it is loaded from the RAM.
- If the cache is full, some data currently stored will be written back into the RAM before new data can be loaded into the cache.

Multiple CPU Cores:

- Modern processors usually have multiple cores.
- These cores have a mix of dedicated cache for each core and shared L2/L3 Cache.
- They can potentially have problems if multiple cores are working on the same data.
- If different values are in separate caches, the data becomes inconsistent.
- A locking mechanism can implemented so the data is consistent.

Paging & Swapping:

- A PC Running complex tasks may require much more RAM than is installed.
- Slower 'backing store' can used (such as a HDD) to store some of the memory contents.
- Both paging and swapping sound similar; they use disk as extension to RAM.
- However, in practice, they are different.

Swap Files:

- Operating Systems make a distinction between active and idle processes.
- Data in ACTIVE processes needs to be in the RAM.
- Data in IDLE processes can be stored on Disk in the swap file until the process is re-activated.

Page Files:

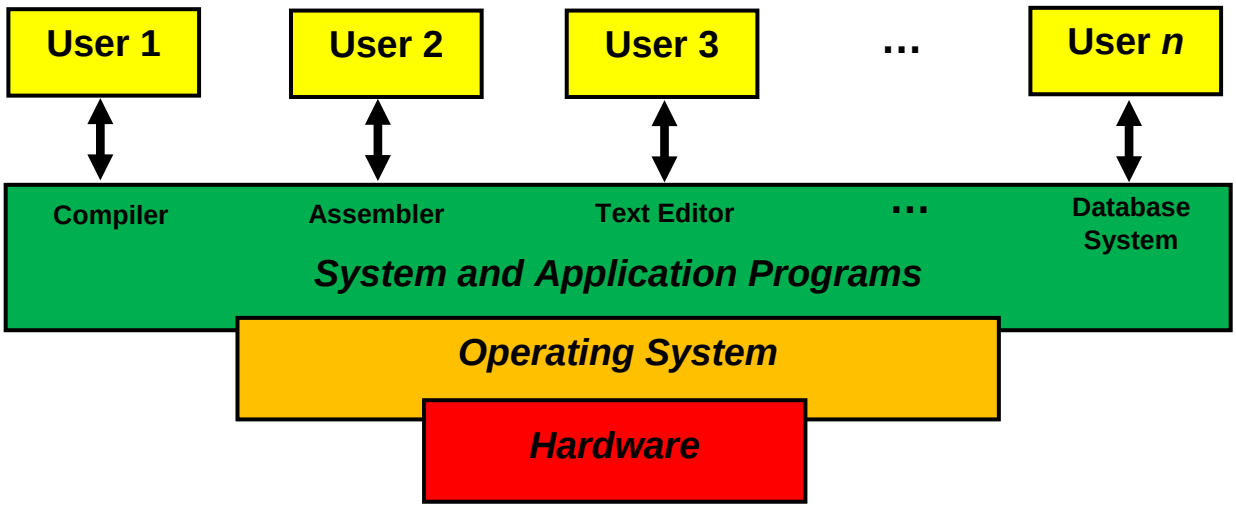
- Memory space is divided into fixed sized blocks called 'Frames'.
- Memory requirements for programs are divided into same size blocks known as 'Pages'.
- The total memory space (Logical Memory) consists of physical memory (Installed RAM) and the extended memory (Disk). I.e 32GB Installed + 8GB RAM Disk = 40GB Logical.
- Pages are copied into the RAM when needed and copied back to the disk when not.
- These 'Page Files' are transparent to the applications as they are managed by the OS.

<u>Operating Systems:</u>

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

What does an OS try to achieve?

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



What do Operating System do?

- Depends on the point of view;
- Users want ***convenience*** and ***ease of use***.
- ***NOT*** bothered about ***resource utilisation***.
- Shared computers such as a mainframe or a minicomputer must keep all users happy.
- Users of dedicated systems such as workstations have dedicated resources but frequently use shared resources from servers.
- Mobile/Handheld devices have poor resources as they are optimised for usability and battery life.
- Some computers have little or no user interface. I.e. embedded computers in cars.

Operating Systems Definition:

- An OS is a ***resource allocator*** that manages all resources and decides between conflicting requests for ***efficient*** and ***fair resource use***.
- An OS is a ***control program***. It controls execution of programs to prevent errors and improper use of the computer.
- No universally accepted definition.
- The ONE program that is running at all times is the '***kernel'.*** Everything else is either a system program that ships with the OS, or is an application program.

Computer Start-up:

- A ***bootstrap program*** is loaded at the power-up or reboot of the computer.
- It is typically stored in the **ROM** (Read-Only Memory) or the **EEPROM** (Electronically Executable Programmable Read-Only Memory). It is generally known as ***firmware.***
- It initialises all aspects of the system.
- Undertakes the **POST** (Power-On Self-Test. This checks all hardware is okay).
- Reads MBR (Master Boot Record) Code and Disk Partitions.
- Reads the boot config file (grub.conf.boot.ini)
- Loads the Kernel (Vmlinuz or ntoskrnl.exe)
- Jumps to kernel entry point and starts init.

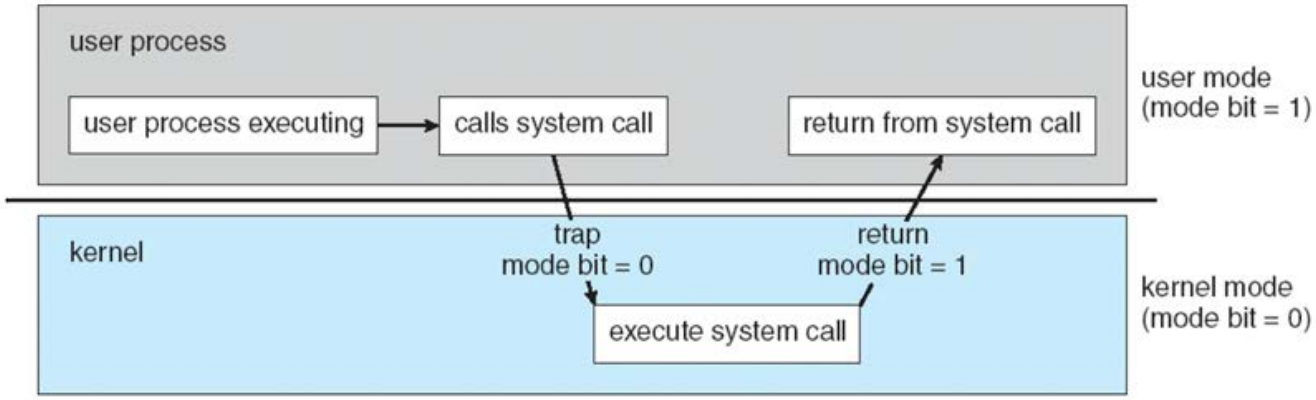
Computer System Operation:

- I/O Devices (Keyboard, Mouse etc.) can execute ***concurrently*** (Simultaneously).
- Each device controller is in charge of a particular ***device type.***
- Each device controller has a ***local buffer*** (A small portion of memory, set to one side).
- The CPU moves data to and from the main memory, to and from the buffers.
- The device controller informs the CPU that it has finished its operation by causing an ***interrupt.***

Common Functions of Interrupts:

- Operating systems are ***interrupt driven.***
- Interrupt transfers control to the interrupt service routine through the ***interrupt vector*****,** which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
- A ***trap*** or ***exception*** is a software generated interrupt caused either by an error or a user request. I.e. a division by 0, infinite loop etc…

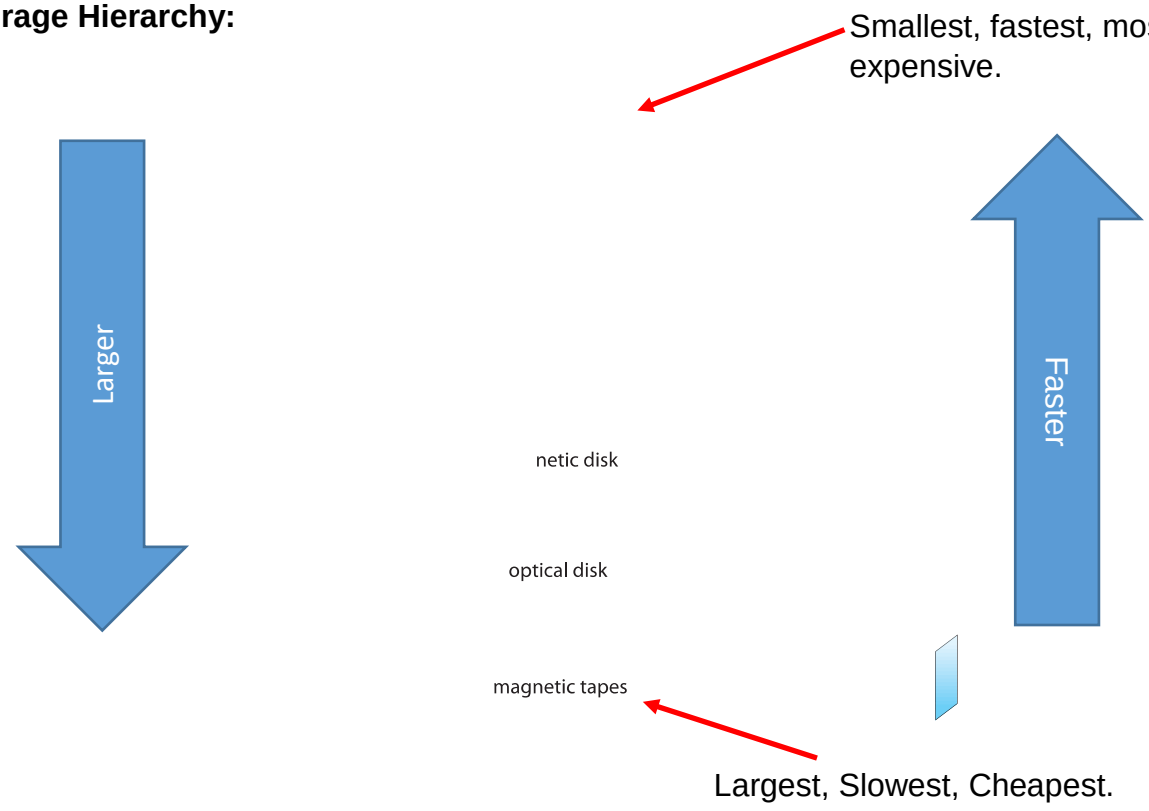
- ***Dual-Mode*** operation allows the OS to protect itself and other system components. These two modes are ***user mode*** and ***kernel mode.***
- A ***mode bit*** is provided by the hardware which distinguishes whether the system is running user code or kernel code.



Storage Structure:

- The main memory (Installed RAM) is the only large storage media that the CPU can access ***directly.***
- The main memory has ***random access*** and is ***volatile*** (It is cleared upon power-off).
- Secondary storage is an extension of the main memory that provides large ***non-volatile*** storage capacity (Meaning it retains its contents after power-off).
- Magnetic Disks are rigid metal or glass platters covered with magnetic recording material. The disk surface is logically divided into ***tracks*** which are sub-divided into ***sectors.***
- Solid state drives are ***faster*** than magnetic disks and are ***non-volatile.***

Storage Hierarchy:

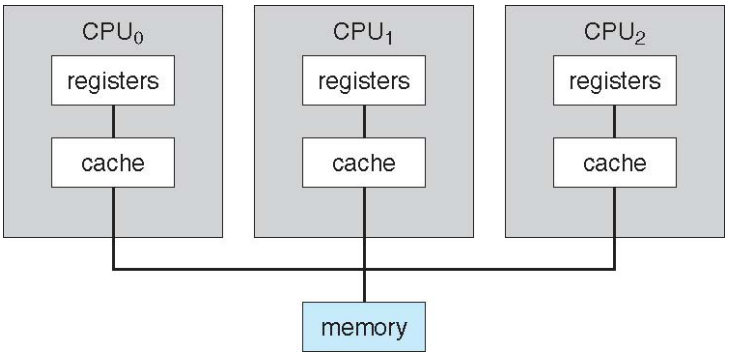


Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Multi-Processor Systems (Asymmetric & Symmetric Multiprocessing):

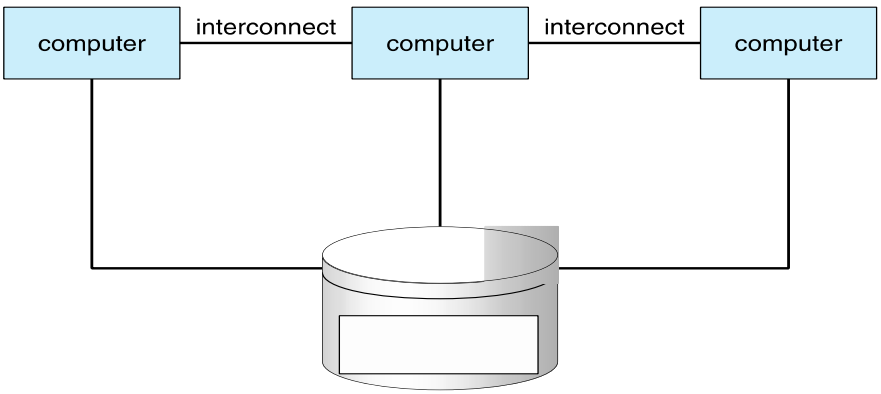
- Advantages include;
  - Increased throughput.
  - Economy of scale.
  - Increased reliability – graceful degradation or fault tolerance.

Symmetric Multiprocessing Architecture:



Clustered Systems:

- Similar to multi-processor systems, but using multiple computers to work together.
- These computers share storage via a SAN (Storage Area Network)
- In the event of a computer failing, the other(s) can continue without fail.
- ***Asymmetric clustering*** has one machine in hot-standby mode.
- ***Symmetric clustering*** has multiple nodes running applications and monitors each other.
- Some clusters are designed for high performance computing by combining the processing power of each node. Applications running on these setups must be written using ***parallelisation.***

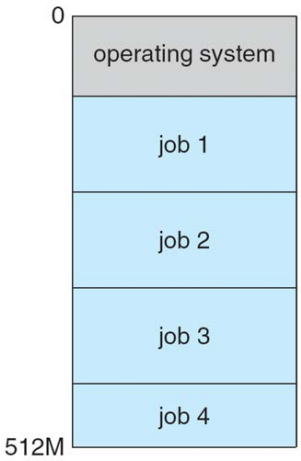


Operating System Structure:

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

Timesharing (Multi-Tasking):

- Timesharing is a logical extensions in which the CPU switches jobs so frequently that the users can interact with each job while it is running, creating ***interactive*** computing.
- Each user has at least one program executing in the memory, this is a process.
- If several jobs are ready to run at the same time, then the CPU schedules them.
- If all the scheduled processes don't fit in the available memory space, swapping moves them in and out in order to run them.
- Virtual memory allows execution of processes that are not completely in the memory.



Process Management:

- A ***process*** is a ***program in execution.***
- Therefore, a ***program*** is a ***passive entity***. And a ***process*** is an ***active entity.***
- Therefore, a process needs resources to accomplish its task (CPU, Memory, I/O, Files).
- Terminating a process will reclaim any re-usable resources.
- A Single-Threaded process has a ***program counter*** that specifies the location of the next instruction to execute. The process executes instruction sequentially, one at a time, until completion.
- Multi-Threaded processes have one program counter per thread.
- Typically, a system will have many processes. Some of which belong to the user, some to the operating system. These run concurrently on one or more CPU's.

Process Management Activities:

- The Operating System is responsible for the following activities;
  - Creating and deleting user and system processes.
  - Suspending and resuming processes.
  - Providing mechanisms for process synchronisation.
  - Providing mechanisms for process communication.
  - Providing mechanisms for deadlock handling.

Protection & Security:

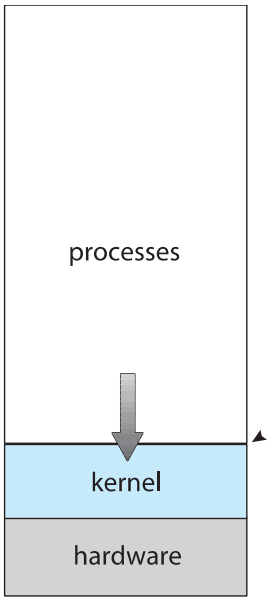
- **Protection:** Mechanisms for control the access of processes or users is defined by the Operating System.
- **Security:** Defence against internal or external attacks (DDOS, Worms, Viruses, Theft).
- Hierarchal system is put in place. User ID's & Password are assigned which determine what each user can manipulate (Files, Processes under control). Can also do this with a ***Group ID.***
- ***Privilege escalation*** allows a user to change to an ID with greater rights.

Computing Environments:

Peer-To-Peer & Client Server Networks (See lecture 3, slides 38 & 39).

Virtualisation Environment:

- Allows operating system to run applications within other Operating Systems.
- **Emulation** is used when the source CPU type is different from the target type. (I.e. Power PC to Intel x86)
- **Virtualisation** is when the Operating System is natively compiled for the CPU type and the running ***guest*** Operating Systems are also natively compiled. (I.e. consider a VMWare running WindowsXP guests, each running applications, all on native WinXP host Operating Systems).

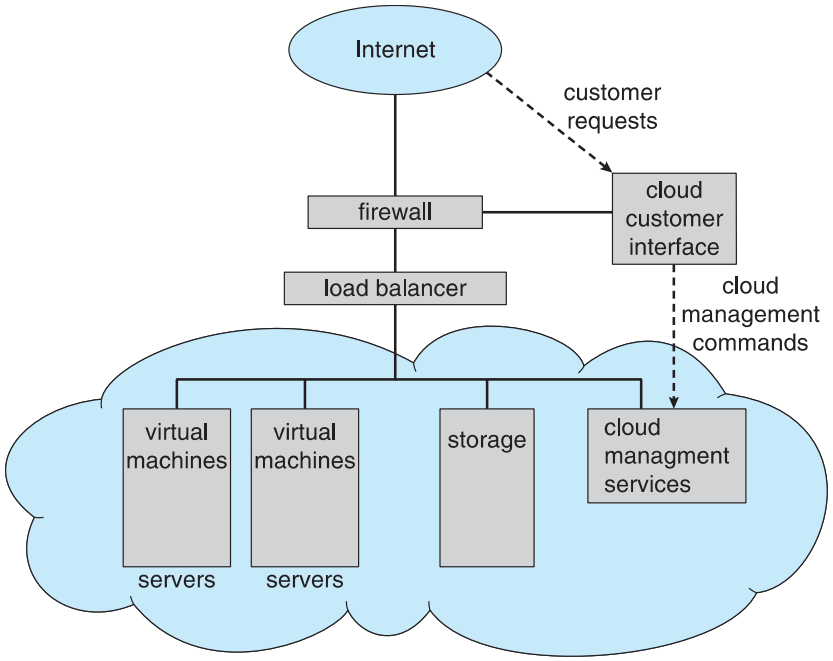


(a)

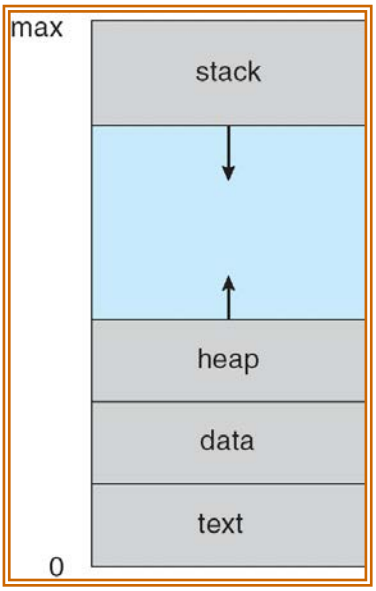
achine

Cloud Computing Environment:

- Can deliver computing, storage and applications as a service across a network.
- It is a logical extensions of virtualisation as based on virtualisation.
- Types: ***Public Cloud, Private Cloud, Hybrid Cloud*** (Both Public & Private)***.***



Processes & Threads:



Processes:

As stated previously, a ***process*** is a ***program in execution.***

A process includes;

- A program counter.
- A stack (Temporary values, function parameters).
- A heap (Dynamically allocated memory during run time).
- Data section (Global variables, text section which is the code).

A program ***IS NOT*** a process. A program ***becomes*** a process when its executable (program.exe) is loaded into the memory.

Process State:

As a process executes, it changes state;

- **New:** The process is being created.
- **Running:** Instructions are being executed.
- **Waiting:** The process is waiting for an event to occur.
- **Ready:** The process is waiting to be assigned to a processer.
- **Terminated:** The process has finished execution.

Only one process can be ***running*** on any processer at any instant. Many processes may be in the ***ready*** or ***waiting*** state.