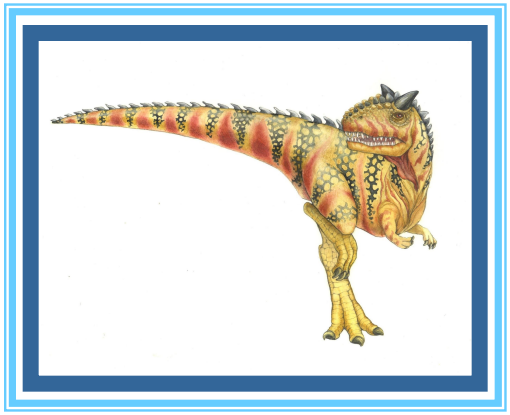
Lecture 1: Introduction

Lecture : 04



Recap

- What is an Operating System
- What Operating Systems Do
- Computer-System Organization
- Operating-System Structure
- Storage Structure
- Operating System Operations
- Operating System architecture
- Distributed Systems
- Process Management
- Memory Management
- Storage Management
- Protection and Security

Objectives

- Hardware Protection
  - I/O Protection
  - CPU Protection
  - Memory Protection
- Operating System Structure
- Computing Environments

Hardware Protection

- Dual-Mode operation
- I/O Protection
- Memory Protection
- CPU Protection

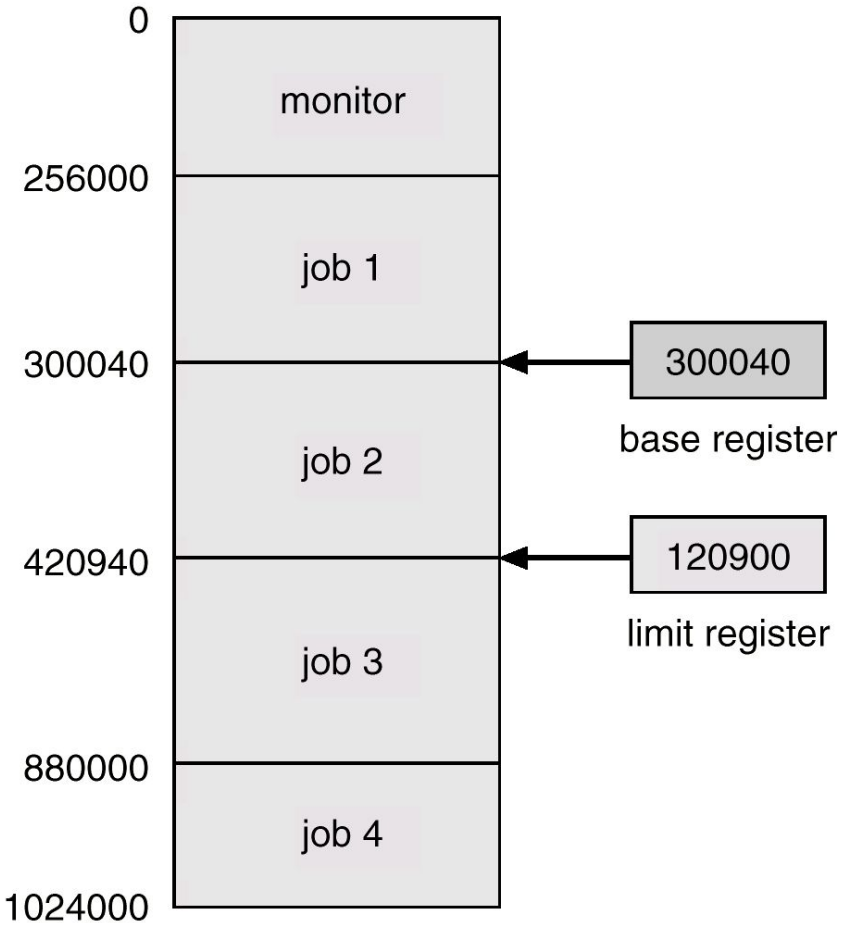
I/O Protection

- All I/O instructions are privileged instructions.
- **A user process might disrupt normal operation of the system by issuing the illegal I/O instructions, by accessing memory locations and addresses within the operating system itself, or by** refusing to surrender CPU. We can use of several mechanisms to ensure that such disruptions should not take place in the system.
- To prevent the users from performing the illegal I/O, we define all the I/O instructions to be as privileged instructions. Therefore users cannot issue the I/O instructions directly; they should do it by making use of the operating system.
- For I/O protection and security to be complete, we should be sure that the user program can never gain control of the computer in the monitor mode. If it could, then the I/O protection could be compromised.

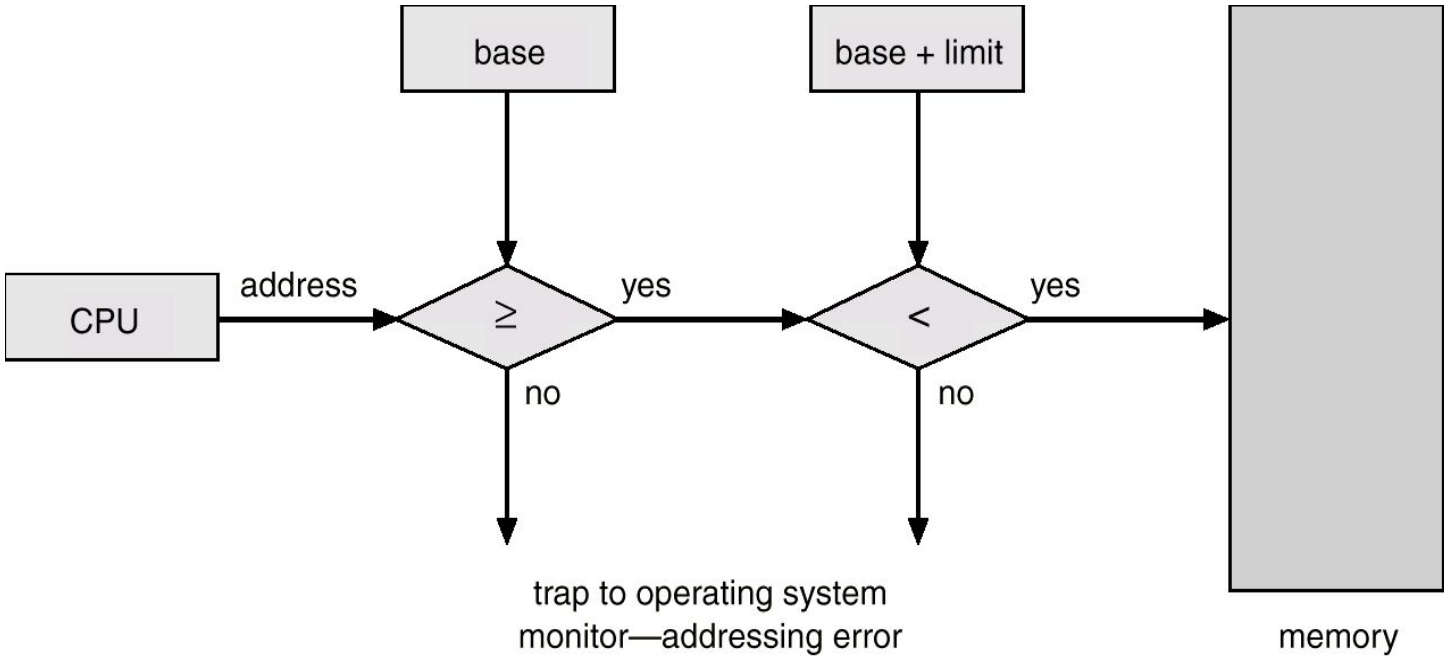
Memory Protection

- Must provide memory protection at least for the interrupt vector and the interrupt service routines.
- Memory protection is a way to control memory access rights on a computer, and is a part of most modern operating systems.
- The main purpose of memory protection is to prevent a process from accessing memory that has not been allocated to it. This prevents a bug within a process from affecting other processes, or the operating system itself
- In order to have memory protection, add two registers that determine the range of legal addresses a program may access:
  - **base register** – holds the smallest legal physical memory address.
  - **Limit register** – contains the size of the range
- Memory outside the defined range is protected.
- **Base and limit registers** are special hardware **registers**. When a process is run, the **base register** is loaded with the physical location where the process begins in memory. The **limit register** is loaded with the length of the process. In other words, they define the logical address space.

A Base And A limit Register Define A Logical Address Space



Protection Hardware



- When executing in monitor mode, the operating system has unrestricted access to both monitor and user's memory.
- The load instructions for the *base* and *limit* registers are privileged instructions.

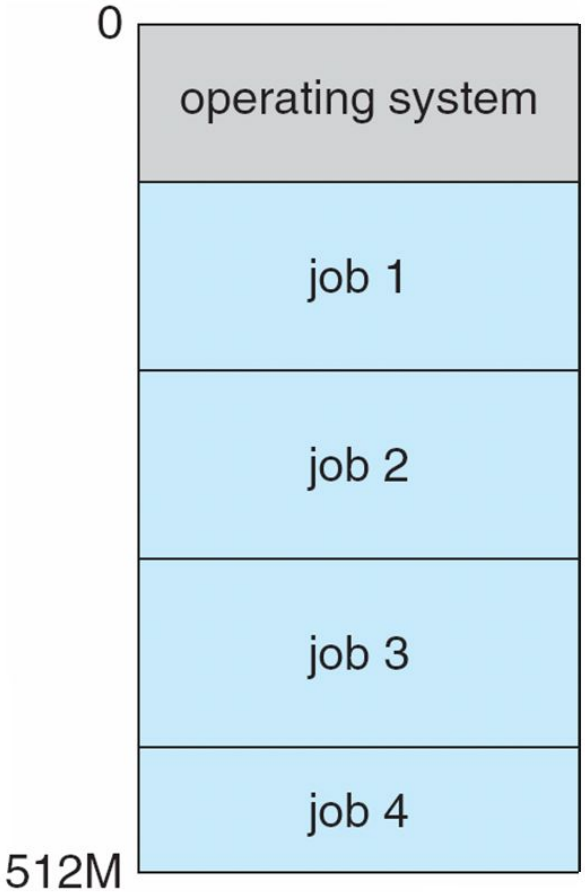
CPU Protection

- *Timer* – interrupts computer after specified period to ensure operating system maintains control.
  - Timer is decremented every clock tick.
  - When timer reaches the value 0, an interrupt occurs.
- CPU protection is needed to prevent a user program from getting stuck in an infinite loop and never returning control to the O/S.
- A timer is used to prevent this. The timer is set to interrupt, say every *N* msecs. The O/S then switches the CPU to another process in a multitasking O/S.
- Loading/setting a timer is a privileged instruction.

Operating System Structure

- **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

Memory Layout for Multiprogrammed System



Computing Environments - Mobile

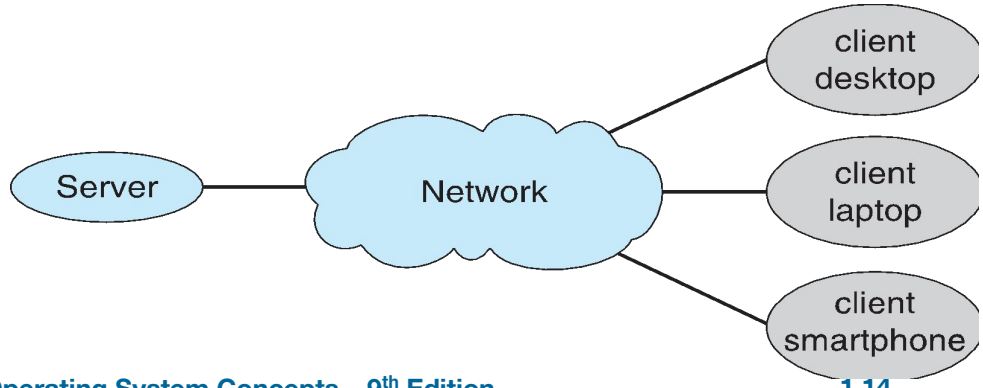
- Handheld smartphones, tablets, etc
- A technology that is capable of providing an environment which enables users to transmit data from one device to other device without the use of any physical link/cables is known as Mobile Computing.
- It means, data transmission is done wireless-ly with the help of wireless devices such as mobiles, laptops etc.
- Whenever any device is connected to a network without being connected physically over a link or cable, data transmission such as messages, voice recording, videos etc. can be done be done by using the concept of mobile computing.
- Mobile Computing technology helps users to access and transmit data from any remote locations without being present there physically.
- Thus, having such a big coverage diameter, it is one of the fastest and most reliable sectors of computing technology field.

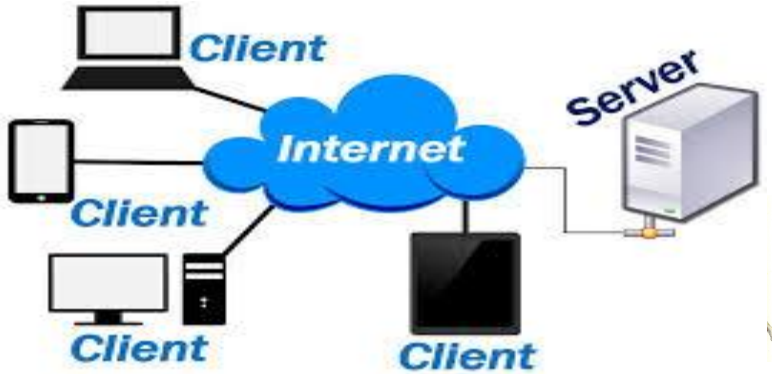
Computing Environments – Distributed

- Distributed computing
  - Collection of separate, possibly heterogeneous, systems networked together
- A distributed computing environment contains multiple nodes that are physically separate but linked together using the network. All the nodes in this system communicate with each other and handle processes in cycle. Each of these nodes contains a small part of the distributed operating system software.
    - **Network** is a communications path, **TCP/IP** most common
      - **Local Area Network** (**LAN**)
      - **Wide Area Network** (**WAN**)
      - **Metropolitan Area Network** (**MAN**)
      - **Personal Area Network** (**PAN**)
  - **Network Operating System** provides features between systems across network
    - Communication scheme allows systems to exchange messages
    - Illusion of a single system

Computing Environments – Client-Server

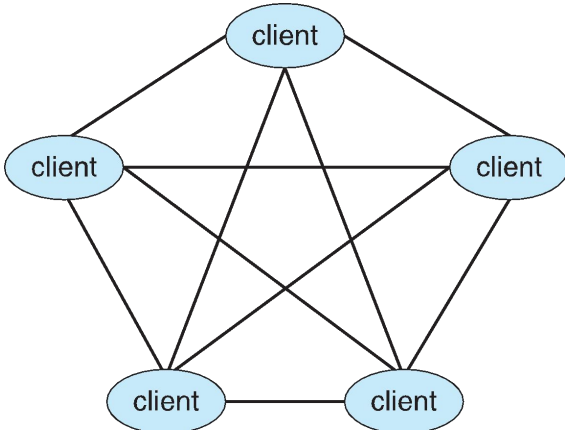
- Client-Server Computing
- A Computer networking model where one or more powerful computers (servers) provide the different computer network services and all other user'of computer network (clients) access those services to perform user's tasks is known as client/server computer networking model.
- In such networks, there exists a central controller called server. A server is a specialized computer that controls the network resources and provides services to other computers in the network.
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files





Computing Environments - Peer-to-Peer

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
  - Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype



Computing Environments - Virtualization

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
- **Virtualization** is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware. It refers to running multiple operating systems on a computer system simultaneously. Apple laptop running Mac OS X host, Windows as a guest

Computing Environments - Virtualization

Virtual Memory is a space where large programs can store themselves in form of pages while their execution and only the required pages or portions of processes are loaded into the main memory. This technique is useful as large virtual memory is provided for user programs when a very small physical memory is there.

- Developing apps for multiple OSes without having multiple systems
- QA testing applications without having multiple systems
- Executing and managing compute environments within data centers

Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for its functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor), software that's available via a third-party over the internet.
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server), hardware and software tools available over the internet.
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use). cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.

SAAS (Software as a Service)

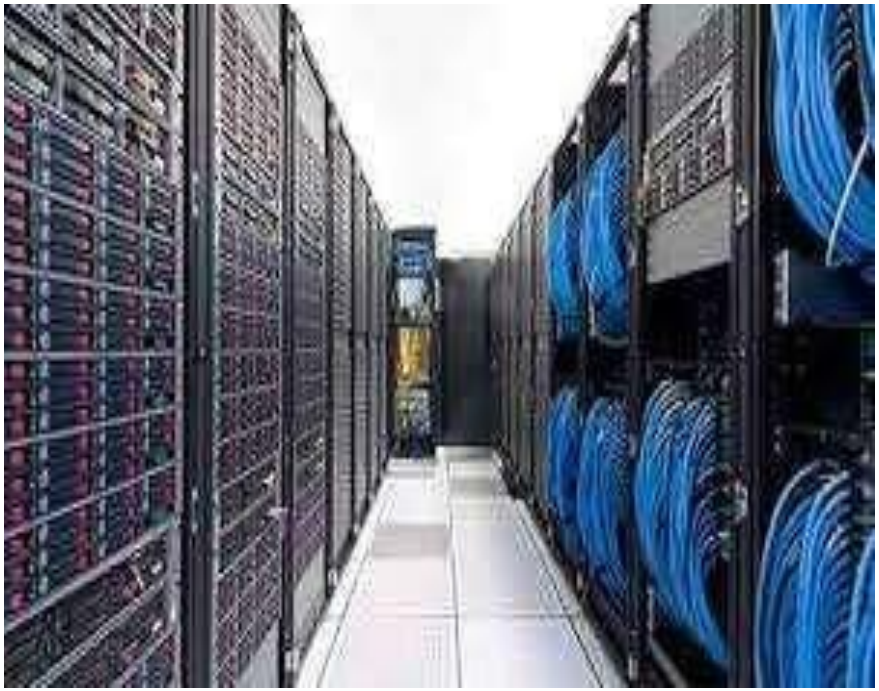
- Examples
  - Google Slides
  - Google Docs
  - Google Sheets
  - Calendar

PAAS (Platform as a Service)

IAAS (Infrastructure as a Service)

Dropbox

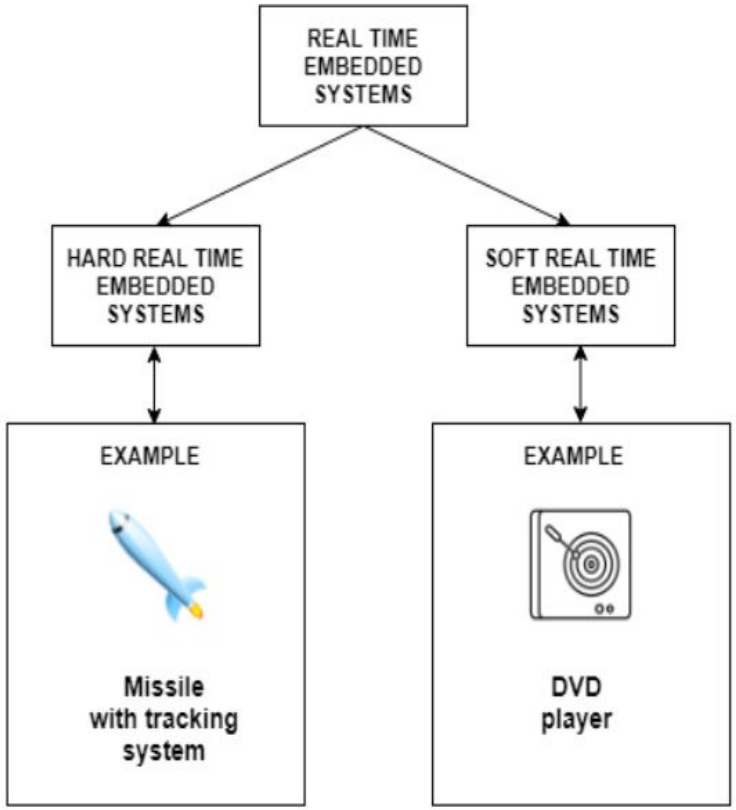
Google Drive



Computing Environments – Real-Time Embedded Systems

- Real-time embedded systems most dominant form of computers
- An Embedded System is more of an application oriented system i.e. it is dedicated to perform a single task (or a limited number of tasks, but all working for a single main aim).
  - Vary considerable, special purpose, limited purpose OS, **real-time OS**
- A Real Time Embedded System is a type of computer system with timing constraints i.e. a system which responds to external events or input stimuli in a timely fashion (within finite and specified time).
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met

Computing Environments – Real-Time Embedded Systems



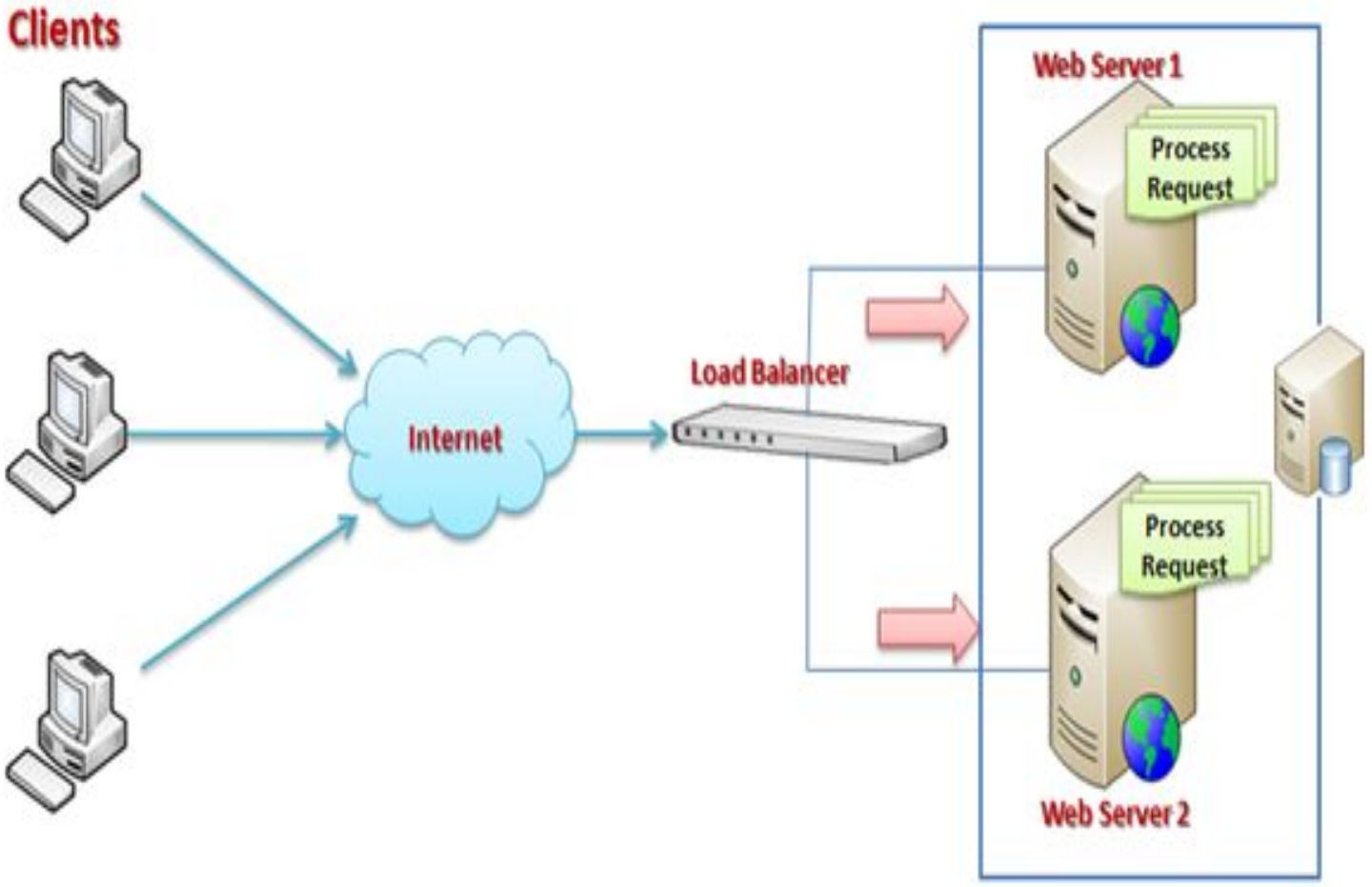
Open-Source Operating Systems

- Open Source operating systems are released under a license where the copyright holder allows others to study, change as well as distribute the software to other people. This can be done for any reason.
- 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**), and many more

Web-Based Computing

- Web based computing puts the program on web site like- google documents.
- Web has become global. PCs used to be more dominant devices but now mobile devices (e.g. smart phones and tablets) are more prevalent modes of access
- In brief, Web-based computing is an environment that consists of ultra-thin clients networked over the Internet or intranet. Applications in this environment consist of code on the servers distributed to thin clients containing a browser, such as Netscape Communicator or Internet Explorer; the browser completely defines the user interface
- Now **load balancers** are used to manage web traffic among similar servers
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers

Web-Based Computing



Open-Source Operating Systems

- The term open source refers to something people can modify and share because its design is publicly accessible. Open source software is software with source code that anyone can inspect, modify, and enhance.
- Operating systems made available in source-code format rather than just binary closed-source
- Started by Free Software Foundation (FSF), which has “copyleft” GNU General Public License (GPL)
- Examples include GNU/Linux, BSD UNIX (including core of Mac OS X), and Sun Solaris

End of Chapter 1