Module-01 **RBT Level-**L1, L2

Introduction to system software:

Introduction to system software definition, system software, Machine structure, components of a programming system, Assemblers, Linkers, Loaders, Compilers, Macros. Relocating loaders and dynamic linking loader designs.

System software refers to the low-level software that manages and controls a computer's hardware and provides basic services to higher-level software. System software refers to the collection of programs and software components that enable a computer or computing device to function properly. It acts as an intermediary between the user and the computer hardware, allowing the user to interact with the hardware and use various applications and programs.

System software is a type of computer program that is designed to run a computer's hardware and application programs and examples of system software include operating systems (OS) (like macOS, Linux, Android, and Microsoft Windows), game engines, search engines (like google, Bing, Yahoo!), industrial automation, computational science software, and (SaS)software as a service application.

Operating systems (OS): Windows, Linux, macOS, etc.

Device drivers: software that enables the communication between hardware and OS.

Firmware: pre-installed low-level software that controls a device's basic functions.

Utility software: tools for system maintenance and optimization.

Boot loaders: software that initializes the OS during startup.

Why use System Software?

Here are some reasons why system software is necessary:

Hardware Communication: System software serves as an interface between the hardware and software components of a computer, enabling them to communicate and work together.

Resource Management: System software manages computer resources such as memory, CPU usage, and storage, optimizing their utilization and ensuring that the system operates efficiently.

Security: System software provides security measures such as firewalls, antivirus software, and encryption, protecting the system and its data from malware, viruses, and other security threats.

User Interface: System software provides a user interface that allows users to interact with the computer or computing device and perform various tasks.

Application Support: System software supports the installation and running of applications and software on the system.

Customization: System software allows for customization of the system settings and configuration, giving users greater control over their computing environment.

Features of System Software:

The most important features of system software are as below.

Memory Management: operating system performs memory management. The OS keeps track of the primary memory and allocates the memory when a process requests it.

Processor Management: OS performs process management. Allocates the main memory (RAM) to a process and de-allocates it when it is no longer required.

File Management: Operating system perform file management. Allocates and de-allocates the resources and decides who gets the resources.

Security: Prevents unauthorized access to programs and data using passwords.

Error-detecting Aids: Production of dumps, traces, error messages, and other debugging and error-detecting methods.

Scheduling: The OS schedules the process through its scheduling algorithms.

Advantages of System Software

Resource management: System software manages and allocates resources such as memory, CPU, and input/output devices to different programs.

Improved performance: System software optimizes the performance of the computer and reduces the workload on the user.

Security: System software provides security features such as firewalls, anti-virus protection, and access controls to protect the computer from malicious attacks.

Compatibility: System software ensures compatibility between different hardware and software components, making it easier for users to work with a wide range of devices and software.

Ease of use: System software provides a user-friendly interface and graphical environment, making it easier for users to interact with and control the computer.

Reliability: System software helps ensure the stability and reliability of the computer, reducing the risk of crashes and malfunctions.

Increased functionality: System software provides a range of tools and utilities for performing various tasks, increasing the functionality and versatility of the computer.

Disadvantages of System Software

Complexity: System software can be complex and difficult to understand, especially for non-technical users.

Cost: Some system software, such as operating systems and security software, can be expensive.

System Overhead: The use of system software can result in increased system overhead, which can slow down the performance of the computer and reduce its efficiency.

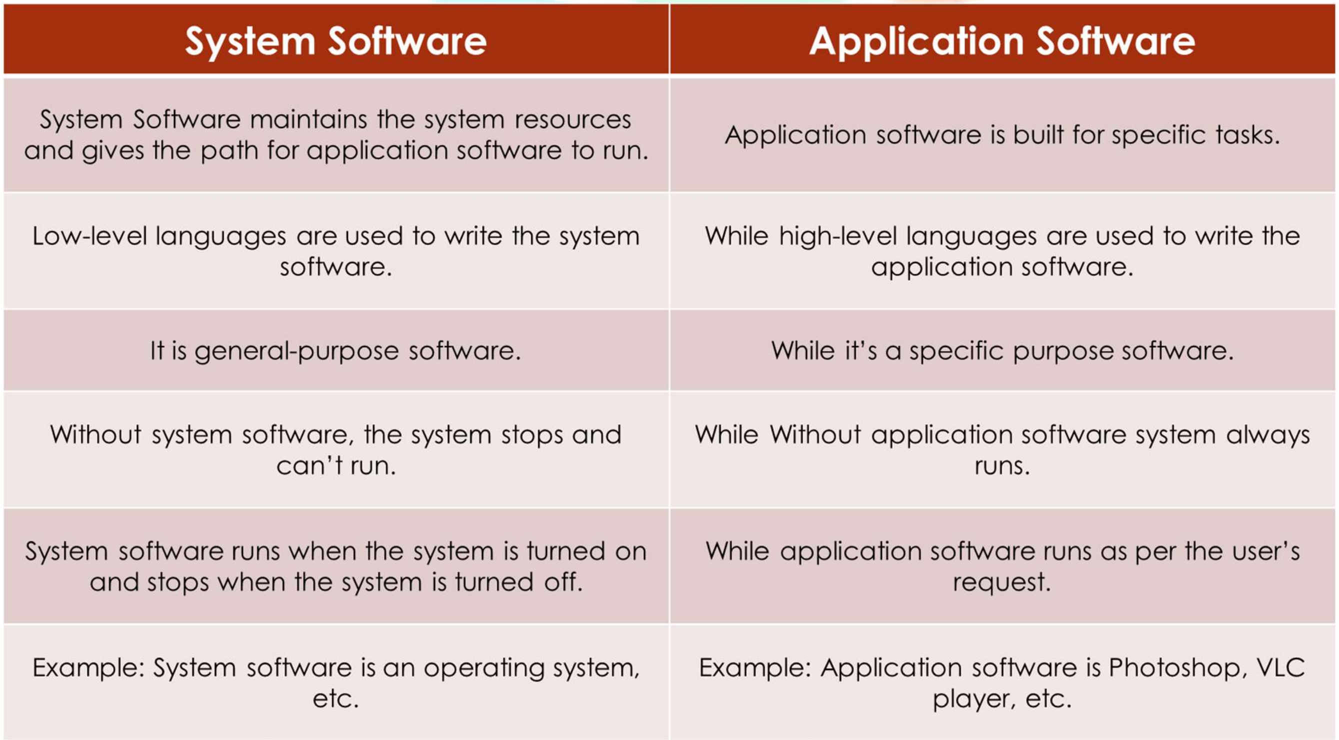
Vulnerability: System software, especially the operating system, can be vulnerable to security threats and viruses, which can compromise the security and stability of the computer.

Upgrades: Upgrading to a newer version of system software can be time-consuming and may cause compatibility issues with existing software and hardware.

Limited Customizability: Some system software may have limited options for customization, making it difficult for users to personalize their computing experience.

Dependency: Other software programs and devices may depend on the system software, making it difficult to replace or upgrade without disrupting other systems.

Difference between System Software and Application Software:



System Software	Application Software
System Software maintains the system resources and gives the path for application software to run.	Application software is built for specific tasks.
Low-level languages are used to write the system software.	While high-level languages are used to write the application software.
It is general-purpose software.	While it's a specific purpose software.
Without system software, the system stops and can't run.	While Without application software system always runs.
System software runs when the system is turned on and stops when the system is turned off.	While application software runs as per the user's request.
Example: System software is an operating system, etc.	Example: Application software is Photoshop, VLC player, etc.

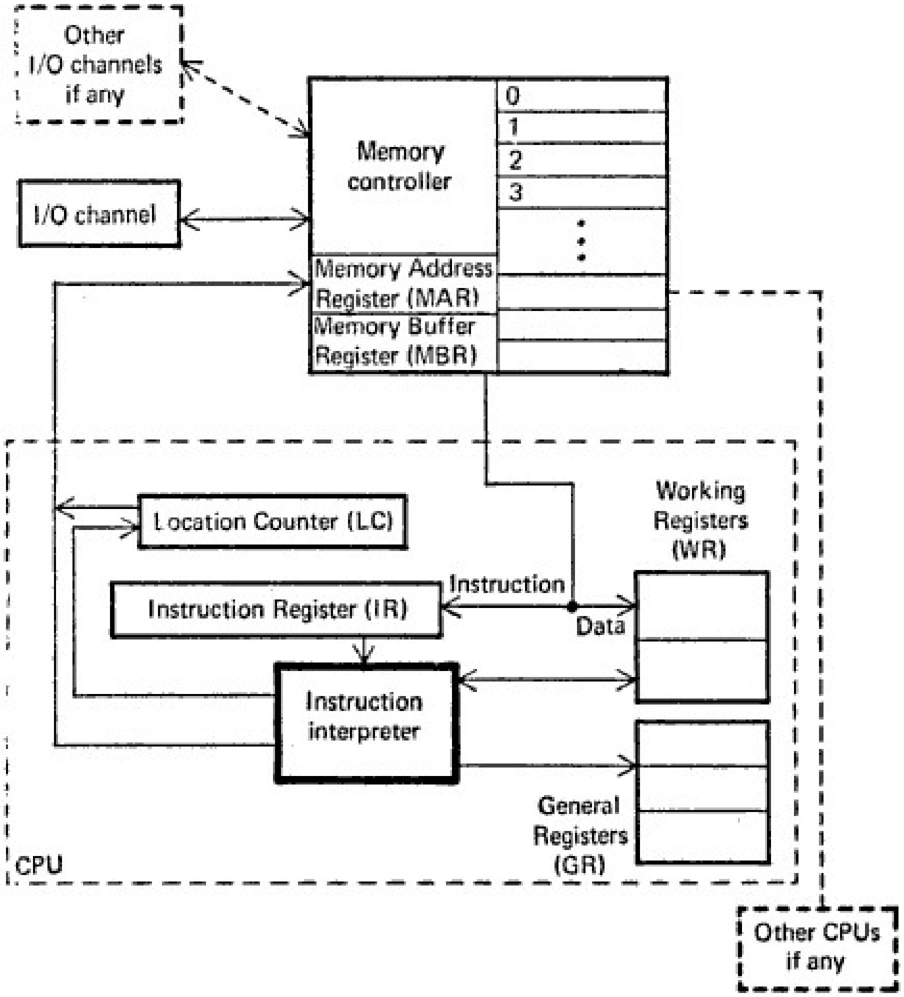
There are two main types of software:

1. Systems software
2. Application software.

System software: It is a type of computer program that manages a computer's hardware and application programs. It acts as an interface between the hardware and user applications.

Application software: It is a type of computer program that allows users to perform tasks such as creating documents, playing games, and browsing the web.

General Machine Structure - Von neumann Architecture



The structure above consists of -

1. Instruction Interpreter

2. Location Counter

3. Instruction Register

4. Working Registers

5. General Register

The **Instruction Interpreter** Hardware is basically a group of circuits that perform the operation specified by the instructions fetched from the memory.

The **Location Counter** can also be called as Program/Instruction Counter simply points to the current instruction being executed.

The **working registers** are often called as the "scratch pads" because they are used to store temporary values while calculation is in progress.

This CPU interfaces with Memory through MAR & MBR

MAR (Memory Address Register) - contains address of memory location (to be read from or stored into)

MBR (Memory Buffer Register) - contains copy of address specified by MAR

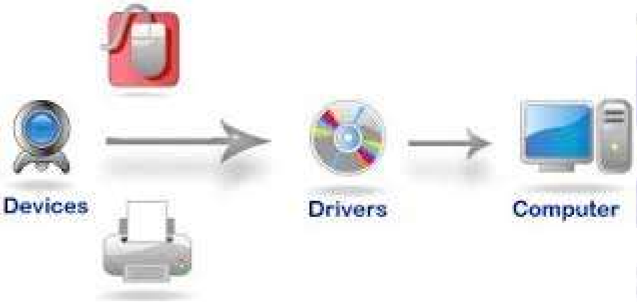
Memory controller is used to transfer data between MBR & the memory location specified by MAR

The role of I/O Channels is to input or output information from memory.

Components of a programming system

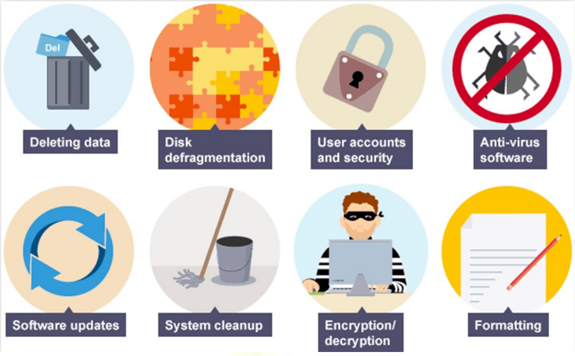
Here are some components of system software:

- **Device drivers:** Device drivers are software components that help the operating system and particular hardware devices communicate with one another.

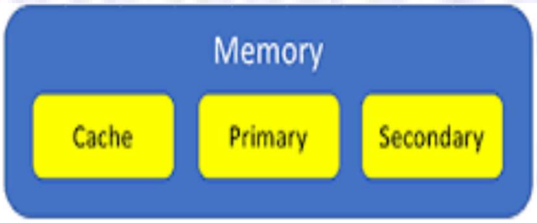


- **Firmware:** This is a system software embedded within the hardware device. It controls the device and performs functions that are essential for the computer system. Firmware is stored in ROM due to which it cannot be modified or deleted. In some devices, firmware can be updated using updates from USB or the manufacturer's device.

- **Utilities:** Utility software is a program specifically designed to help manage and tune system or application software. It is used to support the computer infrastructure - in contrast to application software, which is aimed at directly performing tasks that benefit ordinary users.

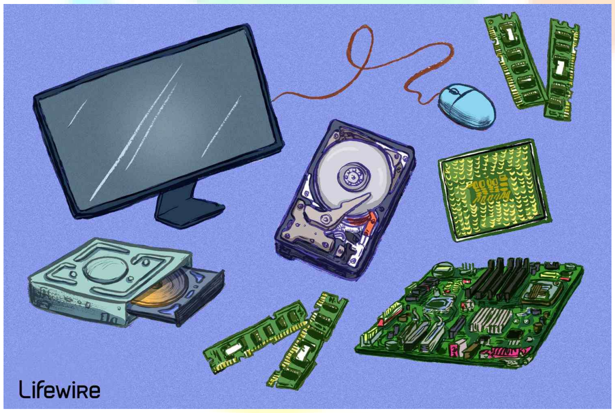


- **Memory management:** Memory management is a function of an operating system that controls and coordinates a computer's main memory. It ensures that blocks of memory space are allocated and managed properly so that the operating system, applications, and other running processes have the memory they need to carry out their operations.

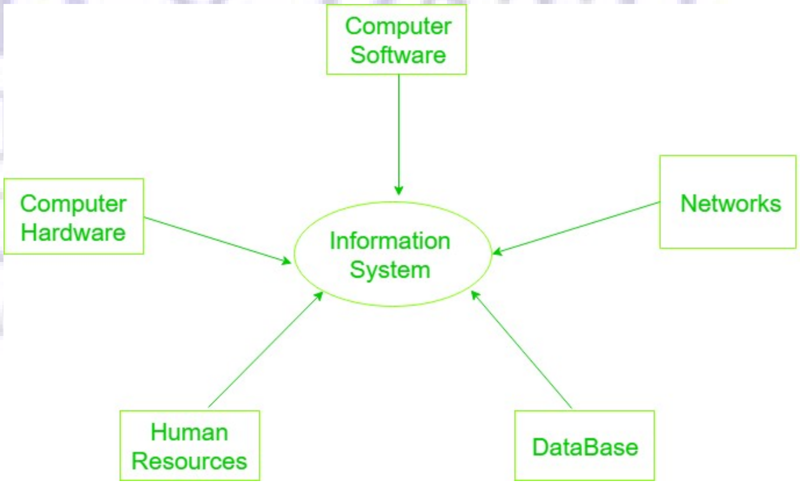


- **Data:** Data is a collection of facts or measurements that has been converted into a form that can be easily processed or moved. It can be in many forms, such as images, text, numbers, audio, or video.

- **Hardware:** Hardware is the physical components of a computer system that store and run instructions from software. Software is the instructions that allow the hardware to perform tasks.

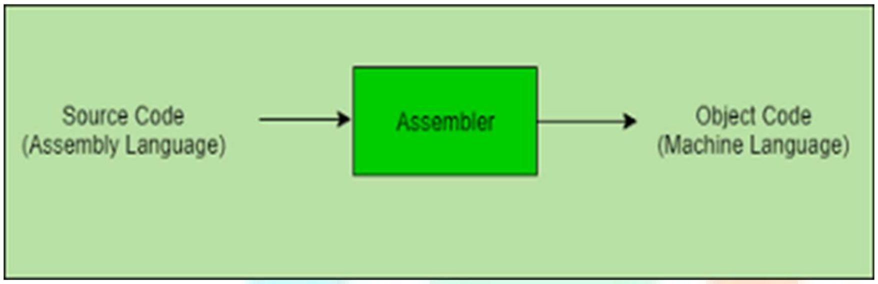


- **Information system:** An information system is software that organizes and analyzes data to help organizations make decisions. It's a combination of software, hardware, and telecommunication networks.

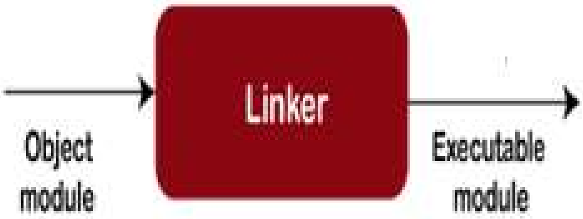


- **Macros**: A Macros is a piece of programming code that runs in excel environment, and it helps to automate routine tasks.
- **System services:** System software provides services to users and applications for using files.

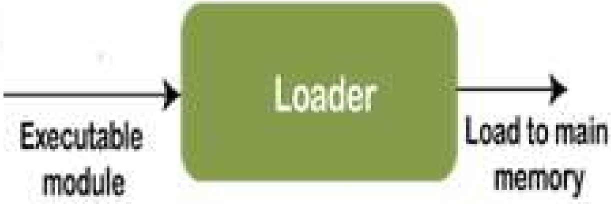
Assemblers: An assembler is a type of computer program that takes in basic instructions and converts them into a pattern of bits that the computer's processor can use to perform basic operations. The assembler's job is to convert assembler or assembly language code into machine code that the computer can then read and execute.



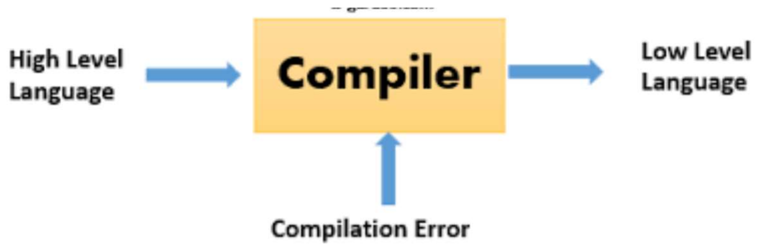
Linkers: A linker is a software tool that plays a crucial role in the compilation process of a program. It takes the object code generated by the compiler and combines it with other necessary libraries and modules to create an executable file.



Loaders: In computer systems a loader is the part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages in the process of starting a program, as it places programs into memory and prepares them for execution.

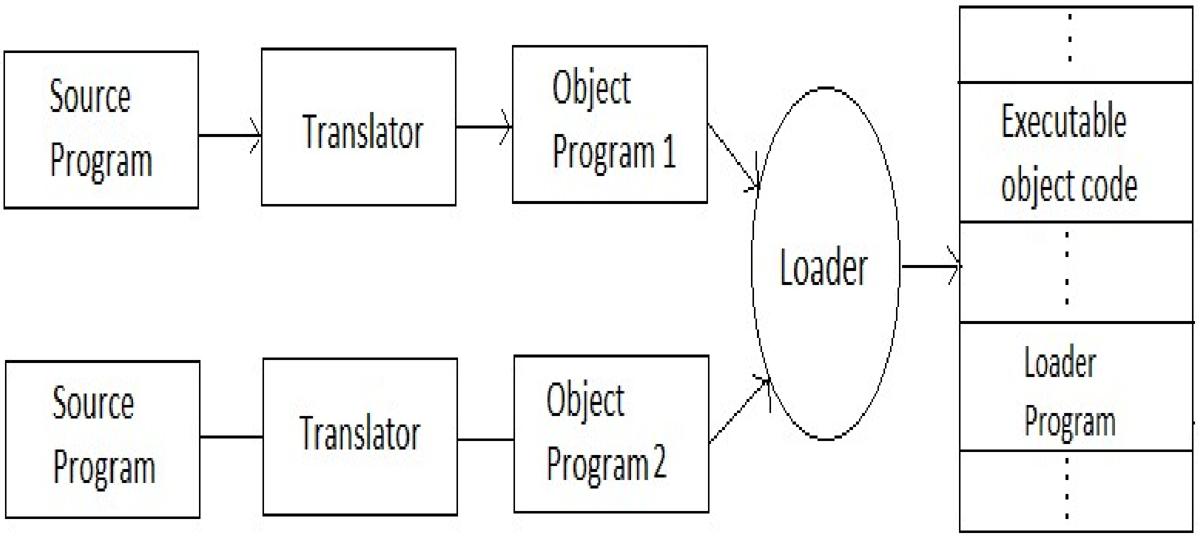


Compilers: A compiler is a software program that converts a computer program written in a high-level programming language into machine language.



Macros: Macros are used to make a sequence of computing instructions available to the programmer as a single program statement, making the programming task less tedious and less error-prone.

Relocating loaders and dynamic linking loader designs.



The basic architecture Loader:

Source program: This is a program written in a high-level programming language that needs to be executed.

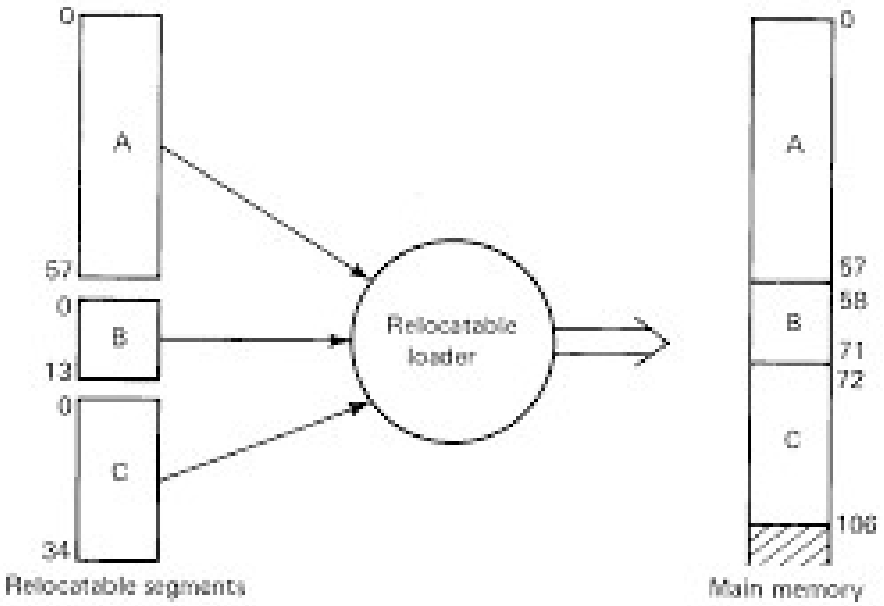
Translator: This component converts the source program into an object program.

Object program: This is the program in a machine-readable form, usually in binary, that contains both the instructions and data of the program.

Executable object code: This is the object program that has been processed by the loader and is ready to be executed.

Relocating loaders

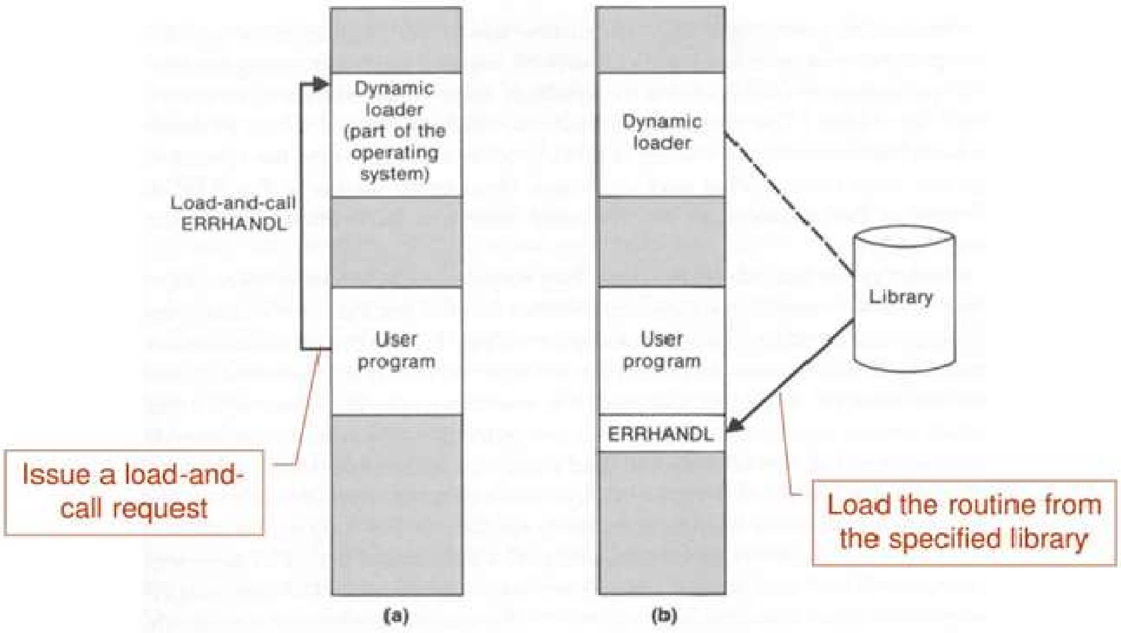
It loads a program at any memory location, and then adjusts all memory references in the program to reflect the new location. This allows the same program to be loaded into different memory locations without having to modify the program's object code



Advantages of relocating loaders include:

- They are simple to implement
- They allow multiple programs or source programs written in different languages
- The loader's task becomes simpler as it simply obeys the instruction regarding where to place the object code to the main memory
- The process of execution is efficient

Dynamic linking loader designs



Issue a load-and-call request

Load the routine from the specified library

- Dynamic loader loads only the required code a data by the program and leaves the rest part in the executable file
- Dynamic loader resolves all the external dependencies and the entire process is known as dynamic linking.
- Dynamic loading is a mechanism by which a computer program can, at run time, load a library into memory, retrieve the addresses of functions and variables contained in the library, execute those functions or access those variables, and unload the library from memory.
- Dynamic loading is utilized to ensure optimal memory consumption. In dynamic loading, a routine is not loaded until it is invoked.