OPERATING SYSTEM

Overlay in Operating Systems

Definition

Overlay is a memory management technique used to **run programs that are larger than the available physical memory** by **loading only the required parts of the program into memory at a time**.

Unused parts are kept on disk and loaded **only when needed**.

Why Overlay is Needed

Early systems had:

- Very **limited main memory**
- Large programs that **could not fit entirely** into RAM

👉 Overlay allows such programs to execute **without increasing memory size**.

Basic Idea

- A program is divided into **modules**
- Only **one module** (overlay) is loaded into memory at a time
- When another module is needed, it **replaces the current one**

Overlay Diagram

```
Program on Disk
+----------------------+
| Main Program         |
| Overlay A            |
| Overlay B            |
| Overlay C            |
+----------------------+

Main Memory
+----------------------+
| Main Program         |
| Overlay Area         |
|   (A or B or C)      |
+----------------------+
```

Only **one overlay** is present in memory at any time.

How Overlay Works

- Main program is loaded into memory

- Required overlay is loaded into the overlay area
- When another overlay is needed:
    - Current overlay is removed
    - New overlay is loaded from disk
- Execution continues

Advantages of Overlay

- Allows execution of **large programs** in small memory
- Efficient use of limited memory
- Reduces memory requirements

Disadvantages of Overlay

- Programmer must manage overlays manually
- Complex program design
- Slower execution due to disk access
- Largely replaced by **virtual memory**

Overlay vs Paging

Feature	Overlay	Paging
Loading	Manual	Automatic
Programmer Involvement	Required	Not required
Memory Management	Program-controlled	OS-controlled
Used In	Older systems	Modern systems

Real-World Status

- Rarely used in modern OS
- Important **theoretical concept**
- Foundation idea behind **virtual memory**

Overlay is a memory management technique where a program is divided into parts, and only the required part is loaded into memory at a time, allowing execution of programs larger than physical memory.