OPERATING SYSTEM

Segmentation vs Paging

Both **Segmentation** and **Paging** are **non-contiguous memory management techniques**, but they differ in **how memory is divided and viewed**.

Paging

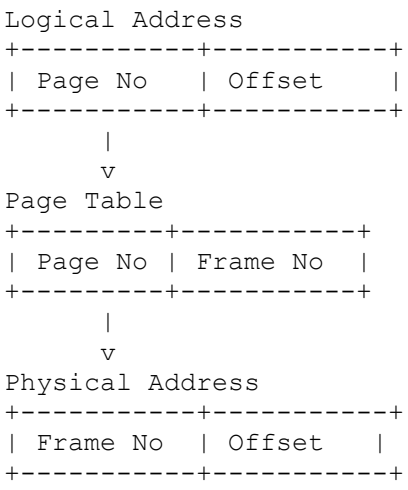
Definition

Paging divides:

- **Logical memory** → fixed-size blocks called **pages**
- **Physical memory** → fixed-size blocks called **frames**

Pages are placed in **any available frame**.

Paging Diagram



Characteristics of Paging

- Fixed-size blocks
- No external fragmentation
- Internal fragmentation possible
- Transparent to programmer

Segmentation

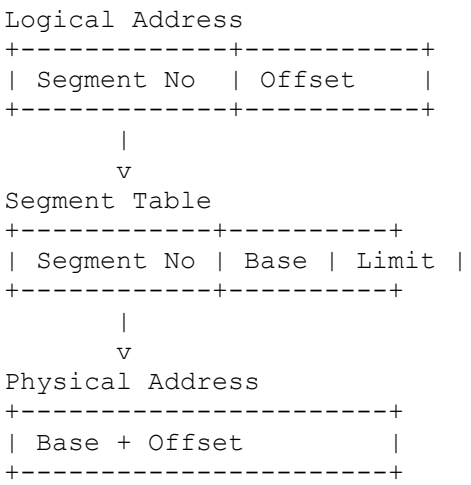
Definition

Segmentation divides a program into **logical units called segments**, such as:

- Code
- Data
- Stack
- Heap

Each segment can be of **different size**.

Segmentation Diagram



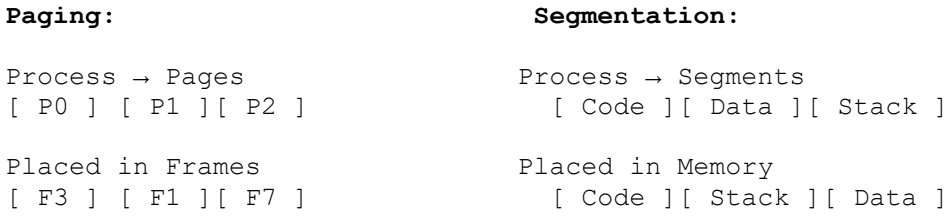
Characteristics of Segmentation

- Variable-size segments
- Matches programmer’s view
- External fragmentation possible
- Supports protection and sharing

Key Differences: Segmentation vs Paging

Feature	Paging	Segmentation
Memory Division	Fixed-size pages	Variable-size segments
Programmer View	Invisible	Visible
Fragmentation	Internal	External
Table Used	Page Table	Segment Table
Address Format	Page No + Offset	Segment No + Offset
Protection	Page-level	Segment-level
Sharing	Difficult	Easy

Paging vs Segmentation (Side-by-Side Diagram)



When to Use Which

- **Paging** → Better for efficient memory utilization
- **Segmentation** → Better for logical program structure
- **Modern OS** → Uses **Segmentation + Paging**

Paging divides memory into fixed-size pages and frames, eliminating external fragmentation. Segmentation divides programs into logical variable-size segments, matching the programmer's view of memory.