Task 7: Segmentation and Paging

1. Introduction

Memory management is a core responsibility of an operating system. Segmentation and Paging are two fundamental memory management techniques used to efficiently utilize main memory and provide logical abstraction to processes.

2. Segmentation

Segmentation divides a process into logical units called segments such as code, data, and stack.

Logical Address Format:
<Segment Number, Offset>

Address Translation:
Segment Number is used as an index in the Segment Table.
Segment Table contains Base and Limit.
Physical Address = Base + Offset (if Offset < Limit).

3. Paging

Paging divides memory into fixed-size blocks called pages (logical memory) and frames (physical memory).

Logical Address Format:
<Page Number, Offset>

Address Translation:
Page Number indexes the Page Table.
Frame Number is obtained from Page Table.
Physical Address = Frame Number + Offset.

4. Segmentation with Paging

This hybrid approach combines segmentation and paging.
Each segment is divided into pages.
It provides both logical view and efficient memory utilization.

5. Example Address Translation

Example (Paging):
Logical Address: Page = 2, Offset = 100
Page Table[2] = Frame 5
Physical Address = Frame 5 + Offset 100

Example (Segmentation):
Logical Address: Segment = 1, Offset = 50
Segment Table[1]: Base = 1000, Limit = 500
Physical Address = 1000 + 50 = 1050

6. Advantages and Disadvantages

Segmentation Advantages:
Logical division, Protection
Segmentation Disadvantages:
External Fragmentation

Paging Advantages:
No external fragmentation
Paging Disadvantages:
Internal fragmentation

7. Conclusion

Segmentation and Paging are essential concepts in operating systems. Understanding address translation helps in grasping how programs are executed efficiently.