Non-Contiguous Memory Allocation – Segmentation

By

Dr. Upasana Pandey

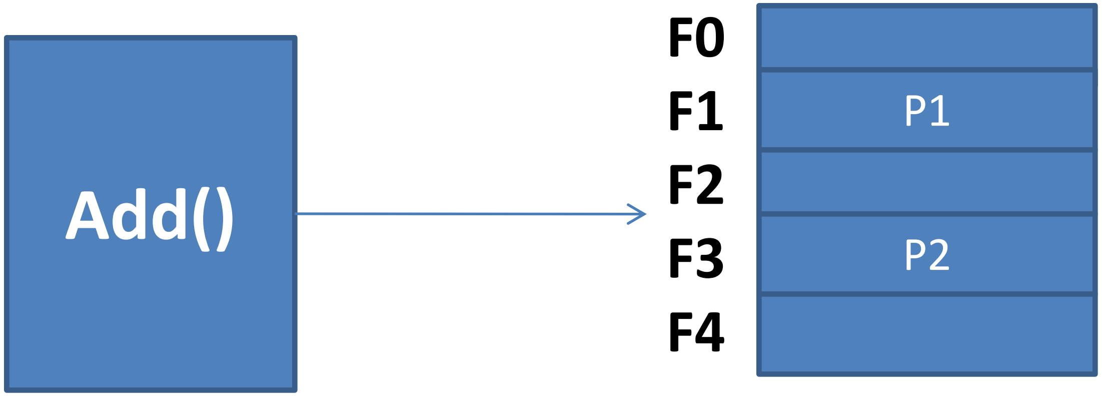
Department of Computer Science & Engineering

IMS Engineering College (College Code:143)

Recap

- Non-contiguous memory allocation is a memory allocation technique.
- It allows to store parts of a single process in a non-contiguous fashion.
- There are two popular techniques used for non-contiguous memory allocation-
  - Paging
  - Segmentation
- Paging: Paging is a fixed size partitioning scheme. In paging, secondary memory and main memory are divided into equal fixed size partitions. The pages of process are stored in the frames of main memory depending upon their availability.

Problem with Paging



1. Discussion: Add() divided into two pages P1 and P2 and stored into frame F1 and F3 in main memory respectively. If CPU generate logical address for P1 and we don't supply P2 just after P1, the generated output would be wrong. Because Add() is a function which correctly executed when both pages should be supplied to CPU. Page Fault may be another case, where P2 is not present in MM.
2. It suffers from Internal Fragmentation too.

Segmentation

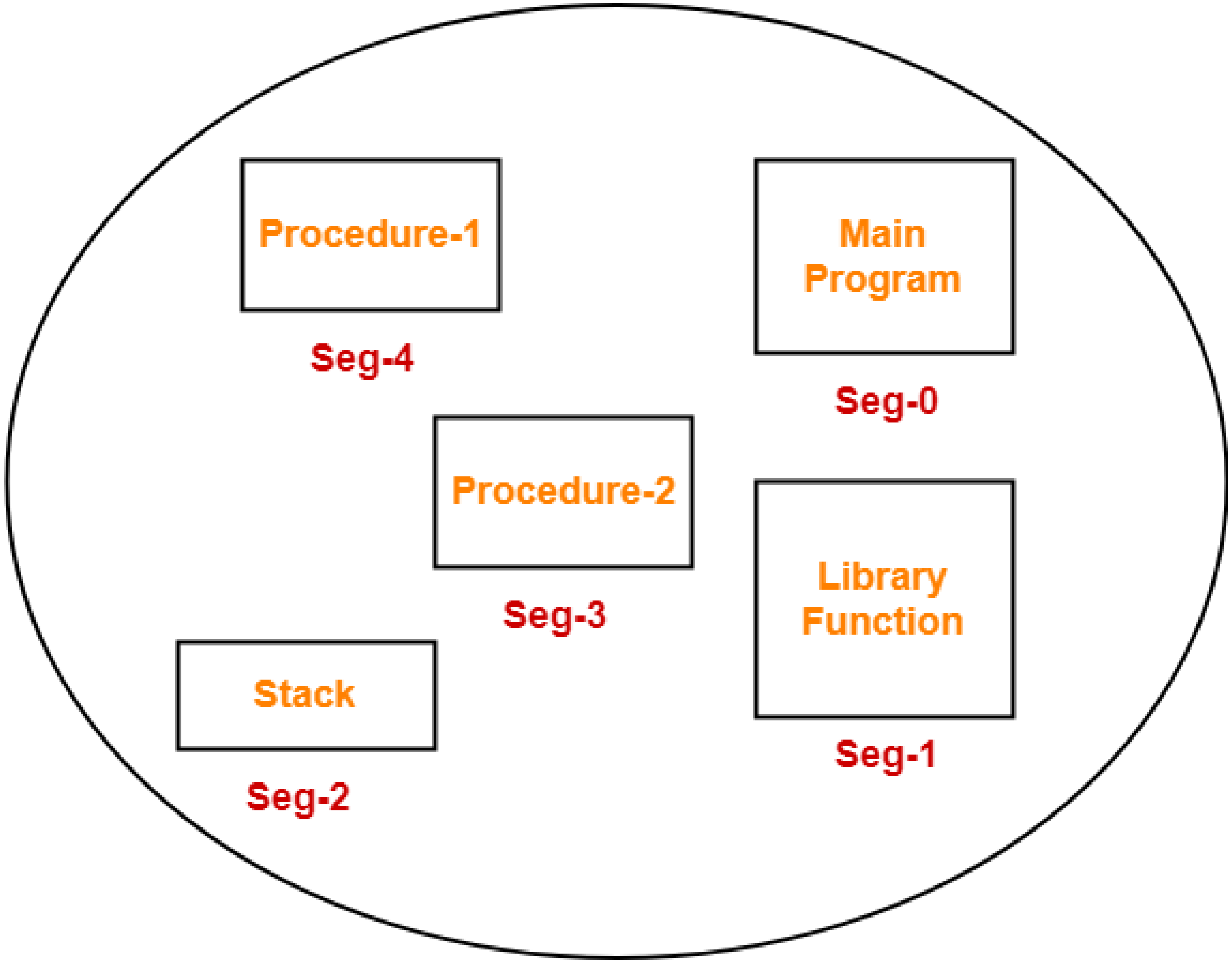
- Like Paging, Segmentation is another non-contiguous memory allocation technique.
- In segmentation, process is not divided blindly into fixed size pages.
- Rather, the process is divided into modules for better visualization.

Characteristics-

- Segmentation is a variable size partitioning scheme.
- In segmentation, secondary memory and main memory are divided into partitions of unequal size.
- The size of partitions depend on the length of modules.
- The partitions of secondary memory are called as **segments**.

Example

One program divided into 5 segment.



Segmentation

Segment Table-

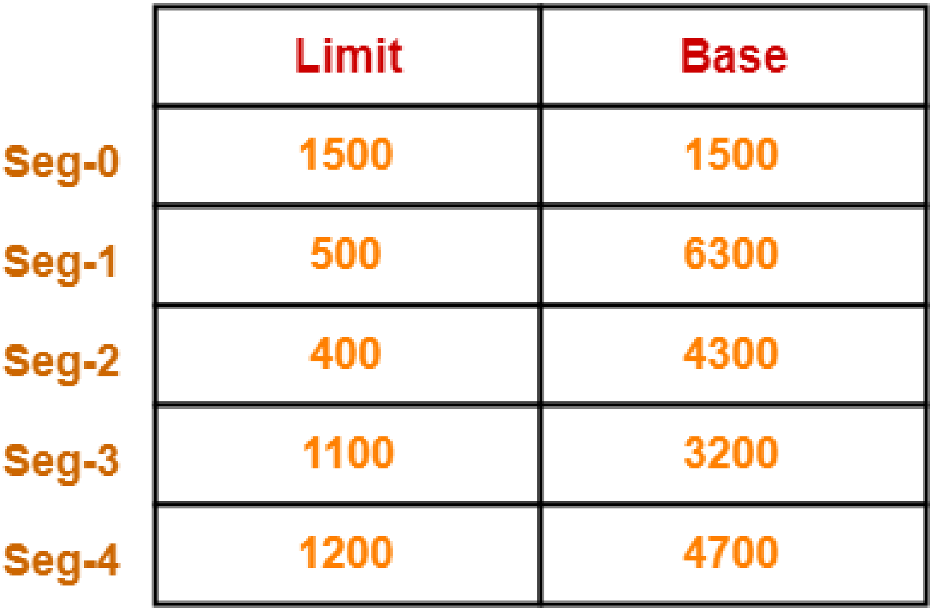
- Segment table is a table that stores the information about each segment of the process.
- It has two columns.
- First column stores the size or length of the segment.
- Second column stores the base address or starting address of the segment in the main memory.
- Segment table is stored as a separate segment in the main memory.
- Segment table base register (STBR) stores the base address of the segment table.

Segmentation

For the above illustration, consider the segment table is-

Here,

- Limit indicates the length or size of the segment.
- Base indicates the base address or starting address of the segment in the main memory.

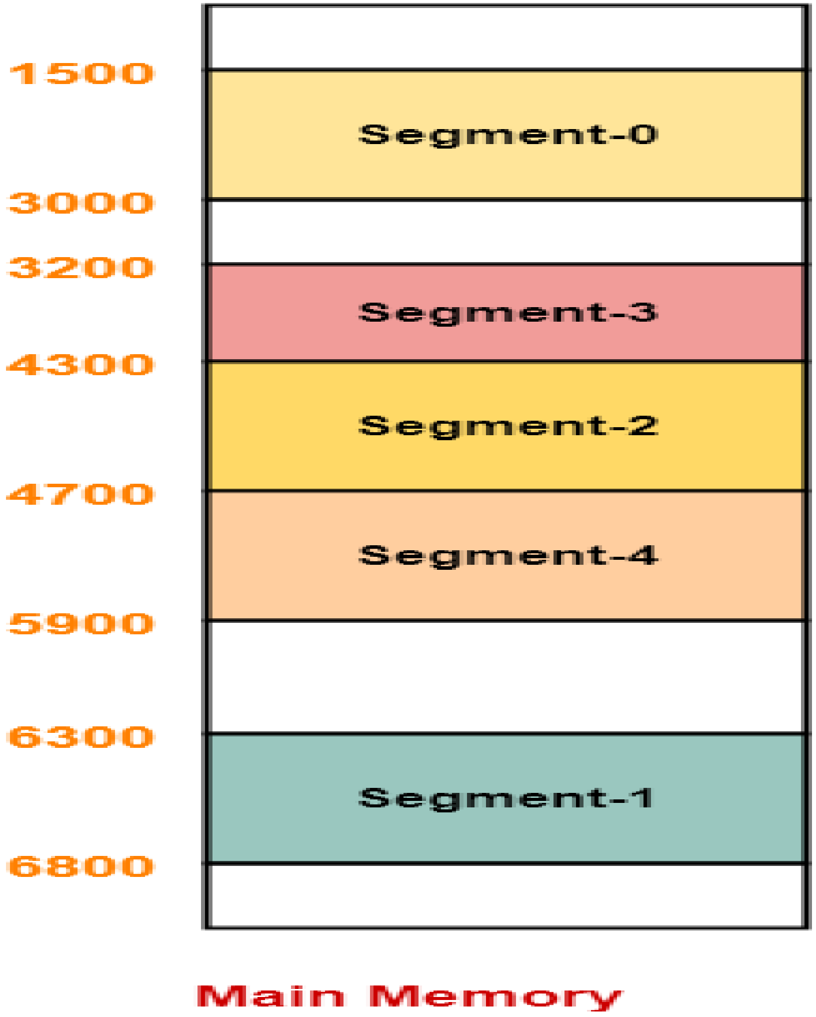


	Limit	Base
Seg-0	1500	1500
Seg-1	500	6300
Seg-2	400	4300
Seg-3	1100	3200
Seg-4	1200	4700

Segment Table

Segmentation

- In accordance to the above segment table, the segments are stored in the main memory as-



Address Translation

Step-01:

CPU generates a logical address consisting of two parts-

- Segment Number: specifies the specific segment of the process from which CPU wants to read the data.
- Segment Offset: specifies the specific word in the segment that CPU wants to read.

Address Translation

Step-02:

- For the generated segment number, corresponding entry is located in the segment table.
- Then, segment offset is compared with the limit (size) of the segment.

Now, two cases are possible-

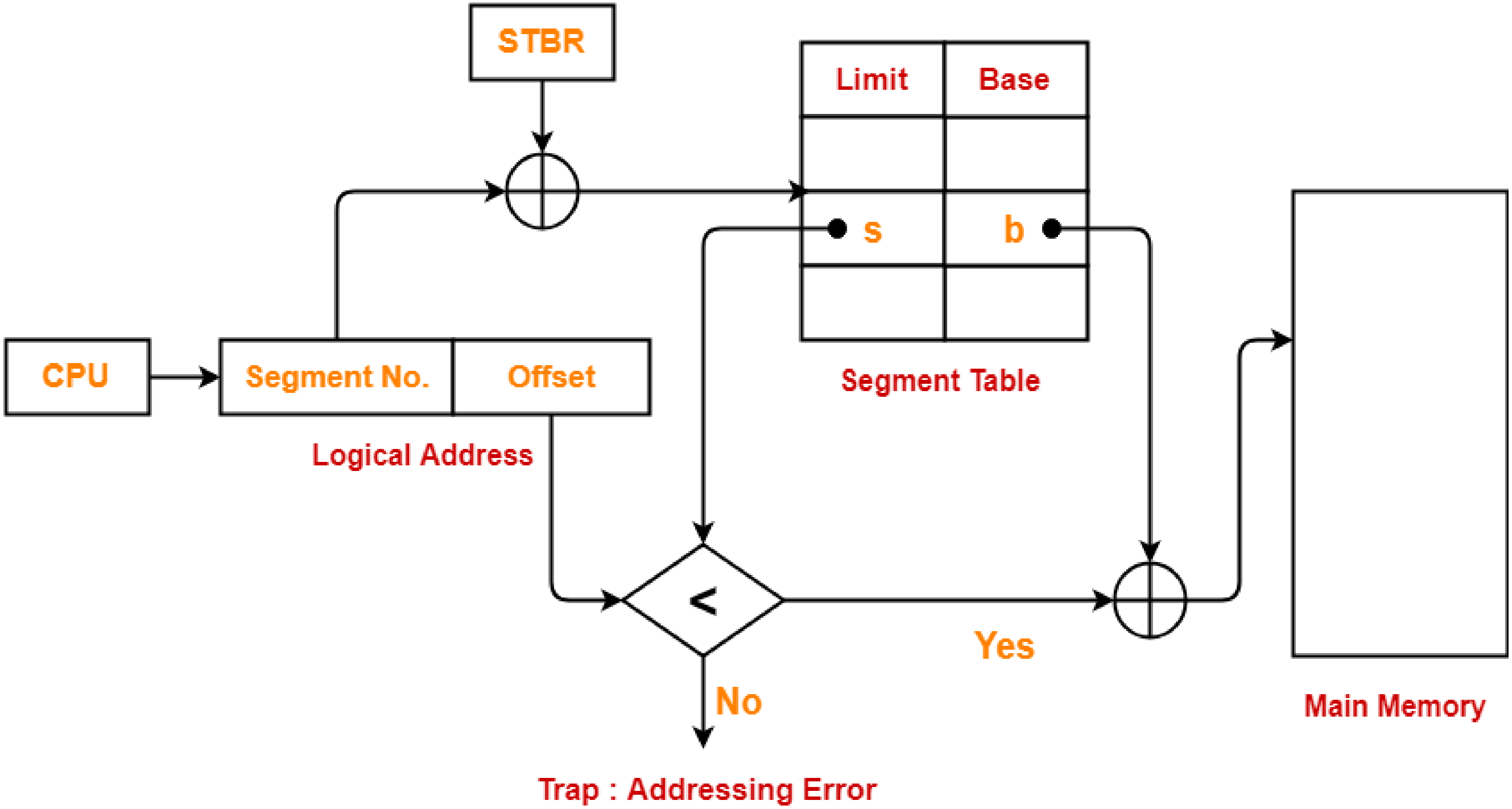
Case-01: Segment Offset >= Limit

If segment offset is found to be greater than or equal to the limit, a trap is generated.

Case-02: Segment Offset < Limit

- If segment offset is found to be smaller than the limit, then request is treated as a valid request.
- The segment offset must always lie in the range [0, limit-1],
- Then, segment offset is added with the base address of the segment.
- The result obtained after addition is the address of the memory location storing the required word.

Address Translation



Translating Logical Address into Physical Address

Segmentation

Advantages-

- It allows to divide the program into modules which provides better visualization.
- Segment table consumes less space as compared to Page Table in paging.
- It solves the problem of internal fragmentation.

Disadvantages-

- There is an overhead of maintaining a segment table for each process.
- The time taken to fetch the instruction increases since now two memory accesses are required.
- Segments of unequal size are not suited for swapping.
- It suffers from external fragmentation as the free space gets broken down into smaller pieces with the processes being loaded and removed from the main memory.

Numerical

Segment Offset must always lie in the range [0, limit-1].
If segment offset becomes greater than or equal to the limit of segment, then trap addressing error is produced.

Segment No	Base	Length	Request	Validity	Calculated address
0	1219	700	430	Yes	1649
1	2300	14	11	Yes	2311
2	90	100	100	No	Trap
3	1327	580	425	Yes	1752
4	1952	96	95	Yes	2047

Thank You