



**DEPARTMENT OF AIML**  
**23CST202- OPERATING SYSTEMS**  
**II YEAR IV SEM AIML-B**  
**UNIT 3-MEMORY MANAGEMENT**  
**TOPIC –SEGMENTATION WITH PAGING**

## **SEGMENTATION WITH PAGING**

Paged Segmentation and Segmented Paging are two different memory management techniques that combine the benefits of paging and segmentation.

1. Paged Segmentation is a memory management technique that divides a process's address space into segments and then divides each segment into pages. This allows for a flexible allocation of memory, where each segment can have a different size, and each page can have a different size within a segment.
2. Segmented Paging, on the other hand, is a memory management technique that divides the physical memory into pages, and then maps each logical address used by a process to a physical page. In this approach, segments are used to map virtual memory addresses to physical memory addresses, rather than dividing the virtual memory into pages.
3. Both Paged Segmentation and Segmented Paging provide the benefits of paging, such as improved memory utilization, reduced fragmentation, and increased performance. They also provide the benefits of segmentation, such as increased flexibility in memory allocation, improved protection and security, and reduced overhead in memory management.

However, both techniques can also introduce additional complexity and overhead in the memory management process. The choice between Paged Segmentation and Segmented Paging depends on the specific requirements and constraints of a system, and often requires trade-offs between flexibility, performance, and overhead.

### **Major Limitation of Single Level Paging**

A big challenge with single level paging is that if the logical address space is large, then the page table may take up a lot of space in main memory. For instance, consider that logical address is 32 bit and each page is 4 KB, the number of pages will be  $2^{20}$  pages. The page table without additional bits will be of the size 20 bits \* 220 or 2.5 MB. Since each process has its own page table, a lot of memory will be consumed when single level paging is used. For a system with 64-bit logical address even a page table of single process will not fit in main memory. For a process with a large logical address space, a lot of its page table entries are invalid as a lot of the logical address space goes unused.



Frame No.	Valid
12	Y
29	Y
-	N
-	N
-	N
36	Y
50	Y
-	N
-	N
-	N
-	N
-	N
-	N
14	Y
32	Y
43	Y

Page table with invalid entries

### Segmented Paging

A solution to the problem is to use segmentation along with paging to reduce the size of page table. Traditionally, a program is divided into four segments, namely code segment, data segment, stack segment and heap segment.

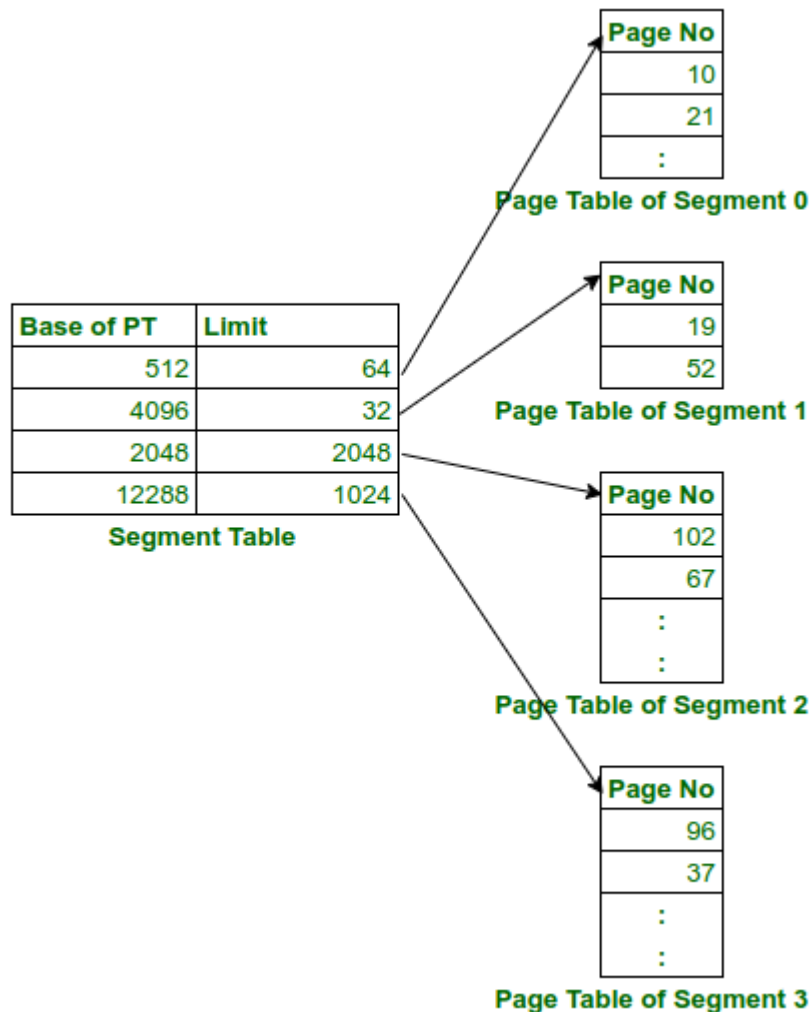
CODE
DATA
STACK
HEAP

Segments of a process

The size of the page table can be reduced by creating a page table for each segment. To accomplish this hardware support is required. The address provided by CPU will now be partitioned into segment no., page no. and offset.

31 30	29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Seg No	Page No	Offset

The memory management unit (MMU) will use the segment table which will contain the address of page table(base) and limit. The page table will point to the page frames of the segments in main memory.



## Segmented Paging

### Advantages of Segmented Paging

1. The page table size is reduced as pages are present only for data of segments, hence reducing the memory requirements.
2. Gives a programmers view along with the advantages of paging.
3. Reduces external fragmentation in comparison with segmentation.
4. Since the entire segment need not be swapped out, the swapping out into virtual memory becomes easier .

### Disadvantages of Segmented Paging

1. Internal fragmentation still exists in pages.
2. Extra hardware is required
3. Translation becomes more sequential increasing the memory access time.
4. External fragmentation occurs because of varying sizes of page tables and varying sizes of segment tables in today's systems.

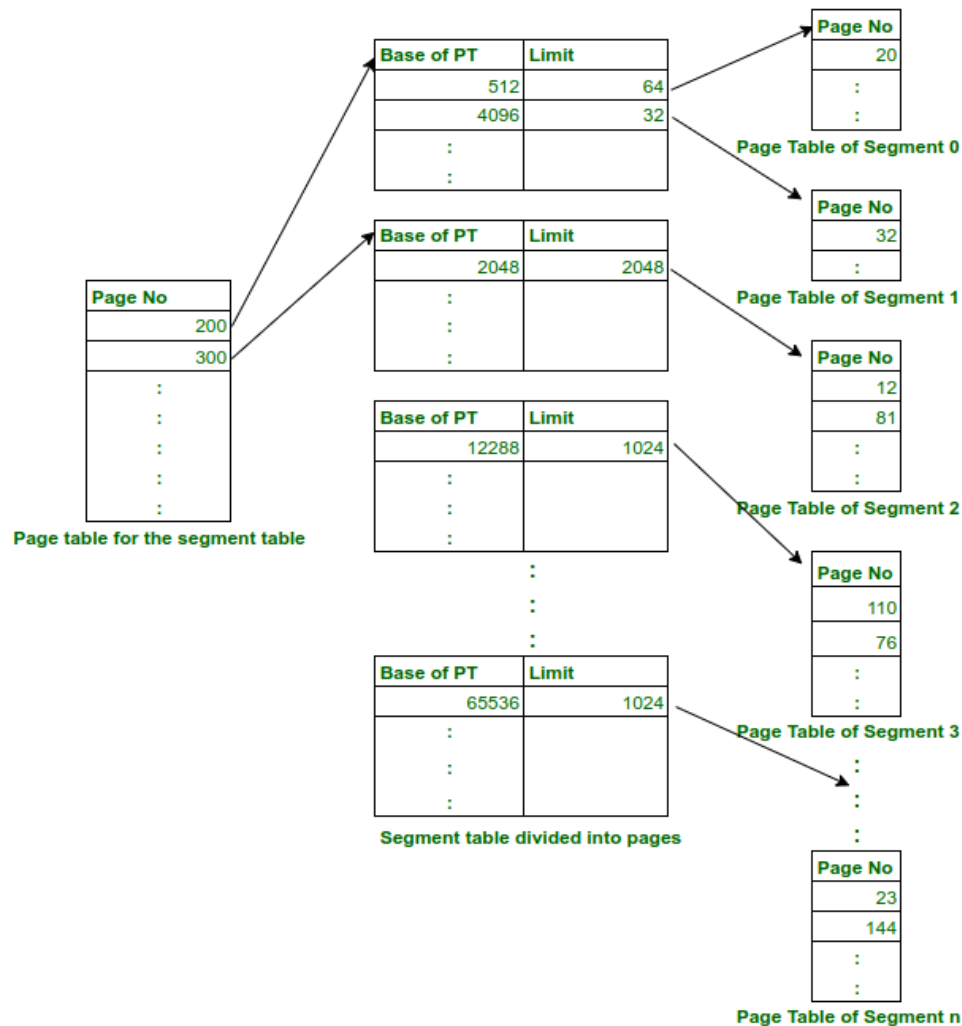
### Paged Segmentation

1. In segmented paging, not every process has the same number of segments and the segment tables can be large in size which will cause external fragmentation



due to the varying segment table sizes. To solve this problem, we use **paged segmentation** which requires the segment table to be paged. The logical address generated by the CPU will now consist of page no #1, segment no, page no #2 and offset.

2. The page table even with segmented paging can have a lot of invalid pages. Instead of using multi level paging along with segmented paging, the problem of larger page table can be solved by directly applying multi level paging instead of segmented paging.



## Paged Segmentation

### Advantages of Paged Segmentation

1. No external fragmentation
2. Reduced memory requirements as no. of pages limited to segment size.
3. Page table size is smaller just like segmented paging,
4. Similar to segmented paging, the entire segment need not be swapped out.
5. Increased flexibility in memory allocation: Paged Segmentation allows for a flexible allocation of memory, where each segment can have a different size, and each page can have a different size within a segment.
6. Improved protection and security: Paged Segmentation provides better protection and security by isolating each segment and its pages, preventing a



single segment from affecting the entire process's memory.

Increased program structure: Paged Segmentation provides a natural program structure, with each segment representing a different logical part of a program.

7. Improved error detection and recovery: Paged Segmentation enables the detection of memory errors and the recovery of individual segments, rather than the entire process's memory.
8. Reduced overhead in memory management: Paged Segmentation reduces the overhead in memory management by eliminating the need to maintain a single, large page table for the entire process's memory.
9. Improved memory utilization: Paged Segmentation can improve memory utilization by reducing fragmentation and allowing for the allocation of larger blocks of contiguous memory to each segment.

#### **Disadvantages of Paged Segmentation**

1. Internal fragmentation remains a problem.
2. Hardware is complex than segmented paging.
3. Extra level of paging at first stage adds to the delay in memory access.
4. Increased complexity in memory management: Paged Segmentation introduces additional complexity in the memory management process, as it requires the maintenance of multiple page tables for each segment, rather than a single page table for the entire process's memory.
5. Increased overhead in memory access: Paged Segmentation introduces additional overhead in memory access, as it requires multiple lookups in multiple page tables to access a single memory location.
6. Reduced performance: Paged Segmentation can result in reduced performance, as the additional overhead in memory management and access can slow down the overall process.
7. Increased storage overhead: Paged Segmentation requires additional storage overhead, as it requires additional data structures to store the multiple page tables for each segment.
8. Increased code size: Paged Segmentation can result in increased code size, as the additional code required to manage the multiple page tables can take up valuable memory space.
9. Reduced address space: Paged Segmentation can result in a reduced address space, as some of the available memory must be reserved for the storage of the multiple page tables.