Ahsanullah University of Science and Technology

Department of Computer Science & Engineering

Course Name: Operating System

Course No: CSE3213

Assignment

Multilevel Paging & Memory Segmentation

Submitted by

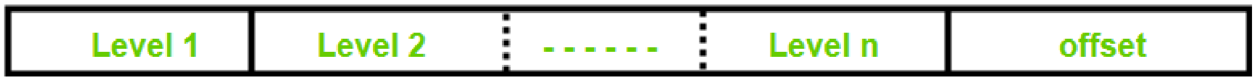
Name	**:** S. M. Tasnimul Hasan Samit
ID	**:** 18.02.04.142
Semester	**:** 3.2
Section	**:** B

Topic 01: Multilevel Paging

Multilevel Paging:

Multilevel Paging is a paging scheme which consist of two or more levels of page tables in a hierarchical manner. It is also known as hierarchical paging. The entries of the level 1 page table are pointers to a level 2 page table and entries of the level 2 page tables are pointers to a level 3 page table and so on. The entries of the last level page table are storing actual frame information. Level 1 contain single page table and address of that table is stored in PTBR (Page Table Base Register).

Virtual Address:



In multilevel paging whatever may be levels of paging all the page tables will be stored in main memory. So, it requires more than one memory access to get the physical address of page frame. One access for each level needed. Each page table entry **except** the last level page table entry contains base address of the next level page table.

The need for Multilevel Paging:

- To avoid the overhead of bringing large size page table in to memory, the multilevel paging will be implemented.
- In the multilevel paging, paging will be applied on the page tables and instead of bringing the entire page tables in to memory, the pages of page table will be brought in to main memory.

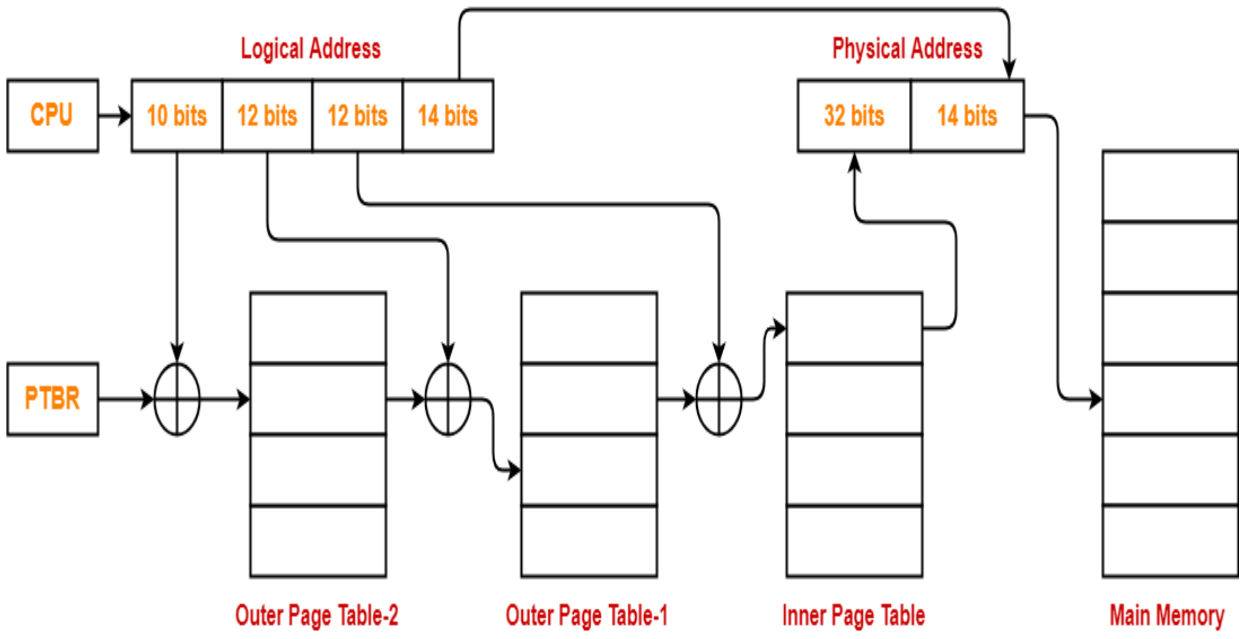


Figure: Three level page table diagram

Reference to actual page frame:

- ❖ Reference to PTE in level 1 page table = PTBR value + Level 1 offset present in virtual address.
- ❖ Reference to PTE in level 2 page table = Base address (present in Level 1 PTE) + Level 2 offset (present in VA).
- ❖ Reference to PTE in level 3 page table= Base address (present in Level 2 PTE) + Level 3 offset (present in VA).
- ❖ Actual page frame address = PTE (present in level 3).

Formula:

- ❖ Number of entries in page table = (virtual address space size) / (page size)
- ❖ Virtual address space size = 2^n B
- ❖ Size of page table >= (number of entries in page table) * (size of PTE)

If page table size > desired size then creates 1 more level.

Problem:

A computer has a 32-bit virtual address space and 1024 Bytes pages. A page table entry is 4 bytes. A multilevel page table is used because each page must fit in a page. How many levels are required?

Solution:

Given,

Data page size = 1024 bytes

Virtual address = 32 bit

Page table entry size = 4 bytes

Size of Page Table = (virtual address/page size) * page table entry size

PTS1 = $(2^{32}/1024) * 4B=2^{24}$ cannot fit in a single page

PTS2 = $(2^{24}/1024) * 4B = 2^{16}$ cannot fit in a single page

PTS3 = $(2^{16}/1024) * 4B=2^{8}$ can be fit in a single page

Therefore, 3 levels are required to fit in a single page.

Advantages Of Multilevel Paging:

- External fragmentation is not present.
- The memory management algorithm is simple.
- Swapping is easy.

Disadvantages Of Multilevel Paging:

- Internal disintegration
- It’s likely that page tables would take up more memory.

Topic 02: Memory Segmentation

Segmentation:

In the case of Operating system, Segmentation is actually considered as a method or system through which memory is divided into several variable size parts, where each part can be called as a segment that can be allocated to a certain process. Segments are actually quite general solution to provide a machine with different independent address spaces, where each segment consists of linear sequence of addresses, begins from 0 and then going until to some maximum extent of value.

Different segments can have different length and they can change during execution. The length might increase whenever something is pushed on the stack. On another note, the length of segments might also decrease whenever something is popped off the stack.

Segmentation can be thought as a scheme which genuinely supports a programmer's view of memory. A programmer may think of a main program with a set of functions or methods, where data structures such as arrays, queues, stacks, variables may also reside. Elements inside a segment are generally identified by the fixed offset from the beginning of a segment.

A logical address can be called as a collection of segments, where every segment has a name and a length. The addresses specify both the name and offset of the segment. Therefore, a programmer needs to specify or fix the address of each segment by specifying the name and an offset for the segment. As a result, we can say, a logical address has two tuples such as:

Segment Number(s): Segment Number is used to represent the number of bits that are required to represent the segment.

Offset(d): Segment offset is used to represent the number of bits that are required to represent the size of the segment.

In general, during the compilation of a program, a compiler usually by default constructs separate segments according to the input program.

- The Code
- The Standard library
- The Stacks used by each thread
- Global variables
- The heap, from which memory is allocated

There can be potentially two types of Segmentation.

- **Virtual Memory Segmentation:** In this kind of segmentation method, every process is divided and segmented into n divisions but they are not segmented together all at the same time.

- **Simple Segmentation:** In this kind of segmentation method, every process is segmented into n parts but this time they are segmented together at once. Therefore, during runtime the processes can be non-contiguous.

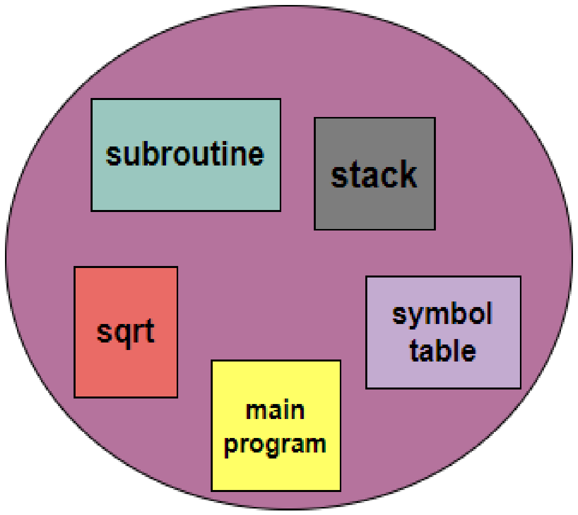


Figure: User's View of a Program

Segmentation Architecture:

In segmentation architecture there consists a Table that is necessary to store the details of all segments of processes. Ideally, there lies no simple relationship between physical and logical addresses.

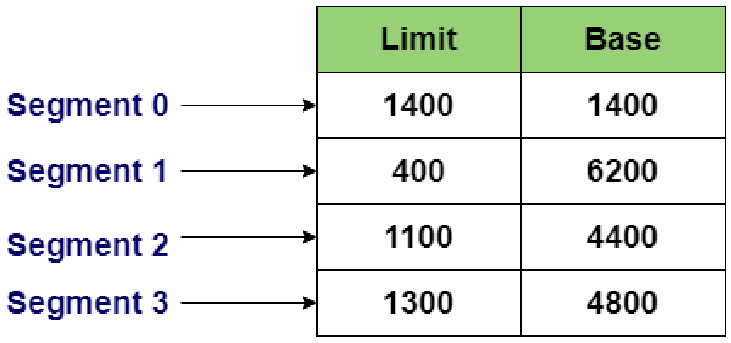
- ❖ The mapping of a two-dimensional Logical address into a one-dimensional Physical address is done using the segment table.
- ❖ This table is mainly stored as a separate segment in the main memory.
- ❖ The table that stores the base address of the segment table is commonly known as the Segment table base register (STBR)

In the segment table each entry has:

1. **Segment Base/base address:** The segment base mainly contain**s** the starting physical address where the segments reside in the memory.
2. **Segment Limit:** The segment limit is mainly used to specify the length of the segment.

Segment Table Base Register (STBR): The STBR register is used to point the segment table's location in the memory.

Segment Table Length Register (STLR): This register indicates the number of segments used by a program. The segment number s is legal if **s<STLR**



	Limit	Base
Segment 0 →	1400	1400
Segment 1 →	400	6200
Segment 2 →	1100	4400
Segment 3 →	1300	4800

Segment Table

Segmentation Hardware:

A logical address of a memory contains two parts such as; a segment number referred as s and an offset which is referred as d. That number is basically used as an index to the segment table. The offset has to be within the range of 0 and the maximum segment limit. If it fails to remain within them, a trap is used to trap to operating system. That trap generally refers to the attempt of beyond end of segment.
Thus, **correct offset + segment base= address in Physical memory**

Whenever, that offset is considered as legal, then it added after the segment base so that it can produce the address in the physical memory of the desired byte. If the offset is lesser than limit then, the offset is considered as a legal offset.

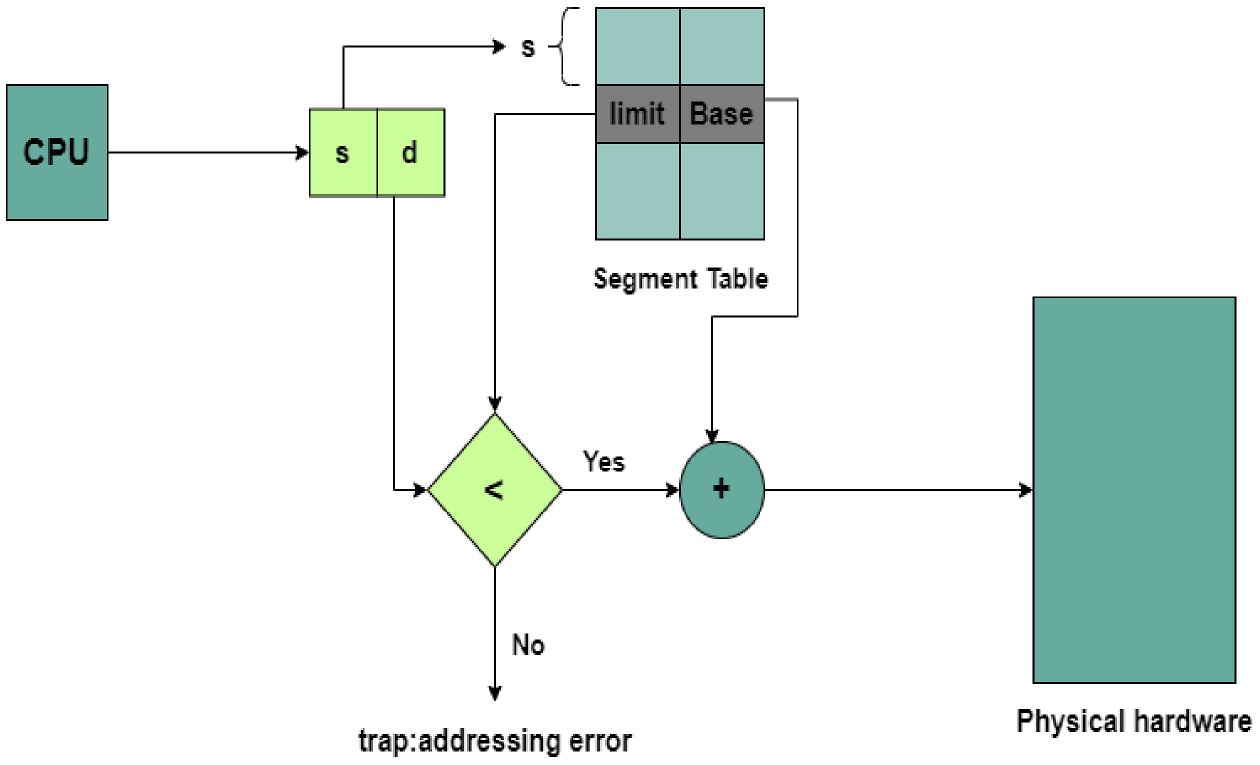


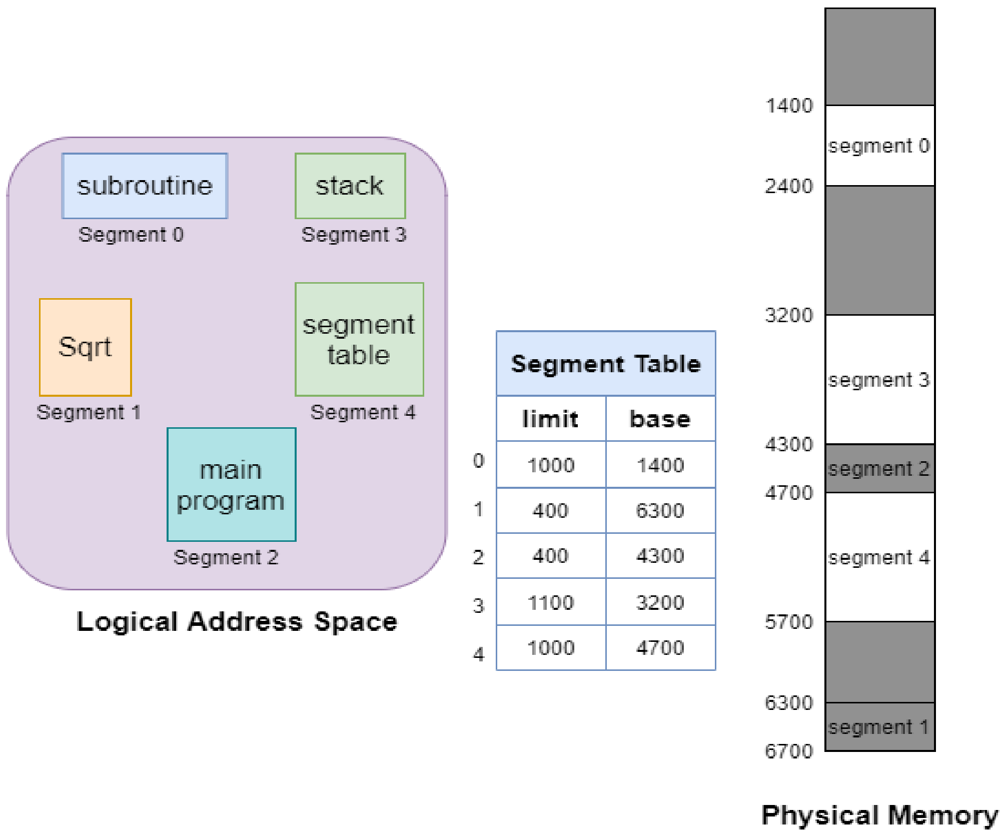
Figure: Segmentation Hardware

From this figure, we can have a general idea about the hardware's of segmentation. We can analyze that is a segment base which consists of the starting of the physical address and besides there is a limit which ensures the length of the segment.

There is also a trap to make sure that any offset larger than limit cannot get access otherwise there will occur an error. That trap makes sure that, if the offset is valid, it will be passed through or otherwise in negation it will fall into trap category.

Therefore, a segment table is indeed labeled as an array of base limit register pairs. Now, we can have further analysis about segmentation by taking a practical example about how segmentation works in memory.

Example Of Segmentation:



Segment Table

	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

Figure: Example Scenario of Segmentation

We can consider a situation according to the figure. We can see there are five segments numbered from 0 to 4. They are stored in physical memory as the figure shows above. There is a separate entry for each segment in the segment table which contains the beginning entry address of the segment in the physical memory (denoted as the base) and also contains the length of the segment (denoted as limit).

From this example, we can see Segment 2 is 400 bytes long and starts at location 4300. Thus, in this case a reference to byte 53 of segment 2 is mapped onto the location (4300+53) = 4353. Again, a reference to segment 3, byte 85 is mapped to (3200+852) = 4052.

But on another note, a reference to byte 750 of segment 1 would cause a trap to the operating system, as the segment 1 has a length of 400 bytes maximum. Therefore, the Trap hardware of segmentation will make sure this falls into trap addressing error category.

In general, from this example we have a genuine idea about how segmentation works in the operating system and the necessity of segmentation for logical and physical memory address.

Advantages Of Segmentation:

- ❖ In the Segmentation technique, the segment table is mainly used to keep the record of segments. Also, the segment table occupies less space as compared to the paging table.
- ❖ Segmentation generally allows us to divide the program into modules that provide better visualization.
- ❖ There is no Internal Fragmentation.
- ❖ Segments are of variable size.
- ❖ Less overhead.
- ❖ The segment table is of lesser size as compared to the page table in paging.

Disadvantages Of Segmentation:

- It can have external fragmentation.
- It is difficult to allocate contiguous memory to variable sized partition.
- Segmentation contains memory management algorithms that can potentially have higher costs which is a problematic thing.
- If a scenario occurs when a process is loaded or removed from the main memory, then the remaining free spaces are broken into smaller portions and this situation causes external fragmentation in Operating System.
- Segments are of unequal size in segmentation and thus are not suitable for swapping.

This was all about Memory Segmentation. All the concepts related to Memory Segmentation have been described shortly.