OPERATING SYSTEMS

Topic - *Segmentation*

Segmentation

- Like Paging, Segmentation is another non-contiguous memory allocation technique.
- In paging the user's view of the memory is not same as the actual physical memory.
- In segmentation, process is not divided blindly into fixed size pages.
- The user/programmer views memory as a collection of variable-sized segments with necessary ordering among the segments.

Programmer's view of a program

- Normally, when a program is compiled, the compiler automatically constructs segments reflecting the input program
- A *C* compiler might create following separate segments:
    1. The code
    2. Global variables
    3. The heap, from which memory is allocated
    4. The stacks used by each thread
    5. The standard C library
- Libraries that are linked in during compile time might be assigned separate segments. The loader would take all these segments and assign them segment numbers.

Segmentation Principle

- Segmentation is a memory-management scheme that supports this programmer view of memory.
- In segmentation, the logical address space is a collection of segments
- Each segment has a name and a length.
- The logical addresses specify both the segment name and the offset within the segment.
- For simplicity of implementation, segments are numbered and are referred to by a segment number, rather than by a segment name. Thus, a logical address consists of a *two-tuple:*

<segment-number, offset>.

Address Translation

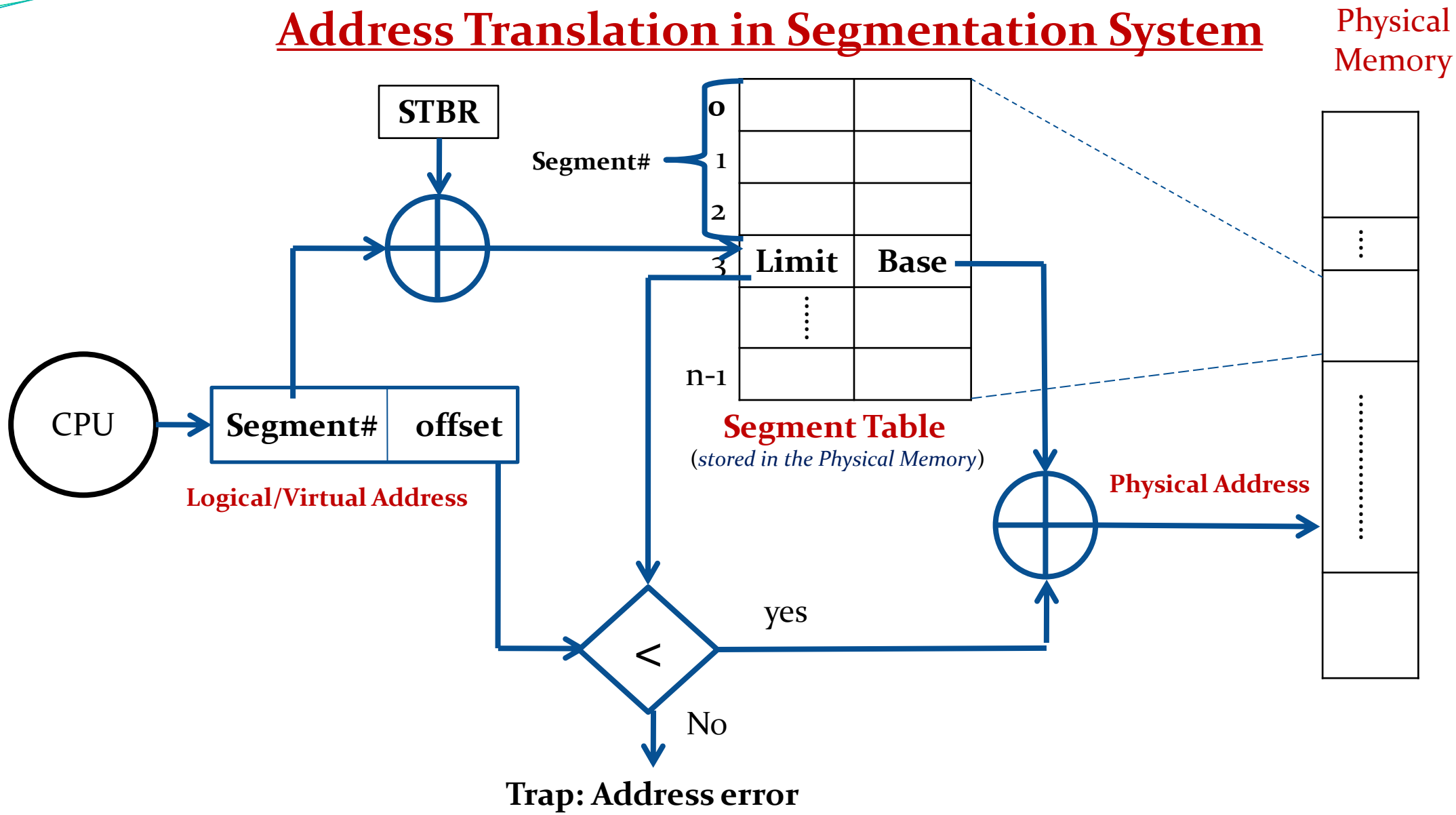
Although the programmer can now refer to objects in the program by a two-dimensional address, the actual physical memory is still a one dimensional sequence of bytes.

So there is need to map two-dimensional user-defined addresses into one-dimensional physical address.

This mapping is effected by a segment table.

- Segment Table 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.

Address Translation in Segmentation System



Address Translation Steps in Segmentation

- CPU generates a logical address consisting of two parts-

  Segment Number: Segment Number specifies the specific segment of the process from which CPU wants to read the data.

  Segment Offset: Segment Offset specifies the specific word in the segment that CPU wants to read.

- For the generated segment number, the corresponding entry is located in the segment table and the segment offset must always lie in the range [0, limit-1],
- The segment offset is compared with the limit (size) of the segment.

  If (Segment Offset >= Limit), then a trap is generated.

  Else If (Segment Offset < Limit), then

  Request is treated as a valid request.

  Segment Offset is added with the base address of the segment.

  The result obtained after addition is the physical address of the memory location storing the required word.

Advantages and Disadvantages

The advantages of segmentation are:

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

The disadvantages of segmentation are:

- 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.
- 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.

Problem

Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

a. 0, 430
b. 1, 10
c. 2, 500
d. 3, 400
e. 4, 112

Answer

a. 219 + 430 = 649
b. 2300 + 10 = 2310
c. illegal reference, trap to operating system
d. 1327 + 400 = 1727
e. illegal reference, trap to operating system