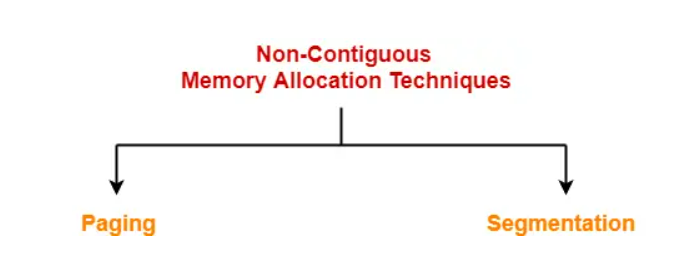
**SEGMENTATION**

There are two popular techniques used for non-contiguous memory allocation-

****

Segmentation is a memory management technique in which a process is divided into modules known as Segments. The size of each segment is not necessarily the same which is different from paging.

**Basic Method:**

Each segment has a name and a length. The addresses specify both the segment

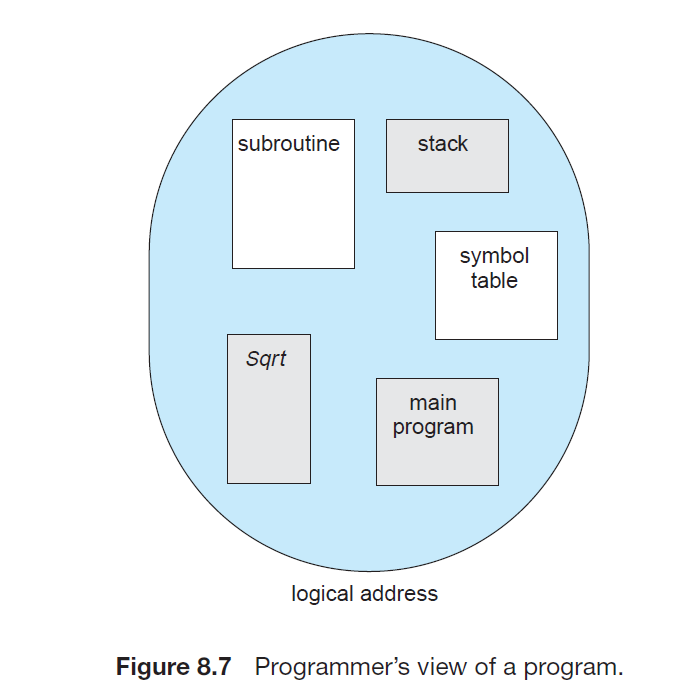
name and the offset within the segment. The programmer therefore specifies each address by two quantities: a segment name and an offset. For simplicity of implementation, segments are numbered and are referred to by a segment number, rather than by a segment name. Thus, a logical address consists of a two tuple:

<segment-number, offset>.

Normally, when a program is compiled, the compiler automatically constructs

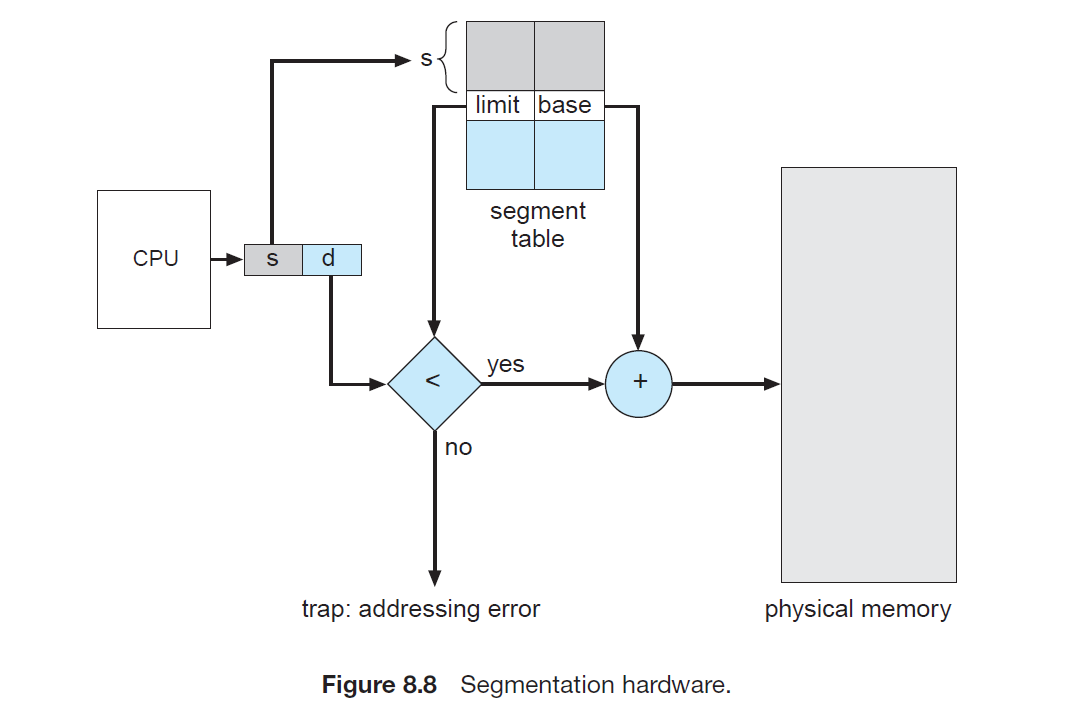
segments reflecting the input program. And a segment is a logical unit such as:

* main program
* procedure
* function
* method
* object
* local variable and global variables.
* symbol table
* common block
* stack
* arrays

****

**Segmentation Hardware:**

Given below figure shows the segmentation hardware:

****

CPU generates a logical address consisting of two parts-

1. Segment Number
2. Segment Offset

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. 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-

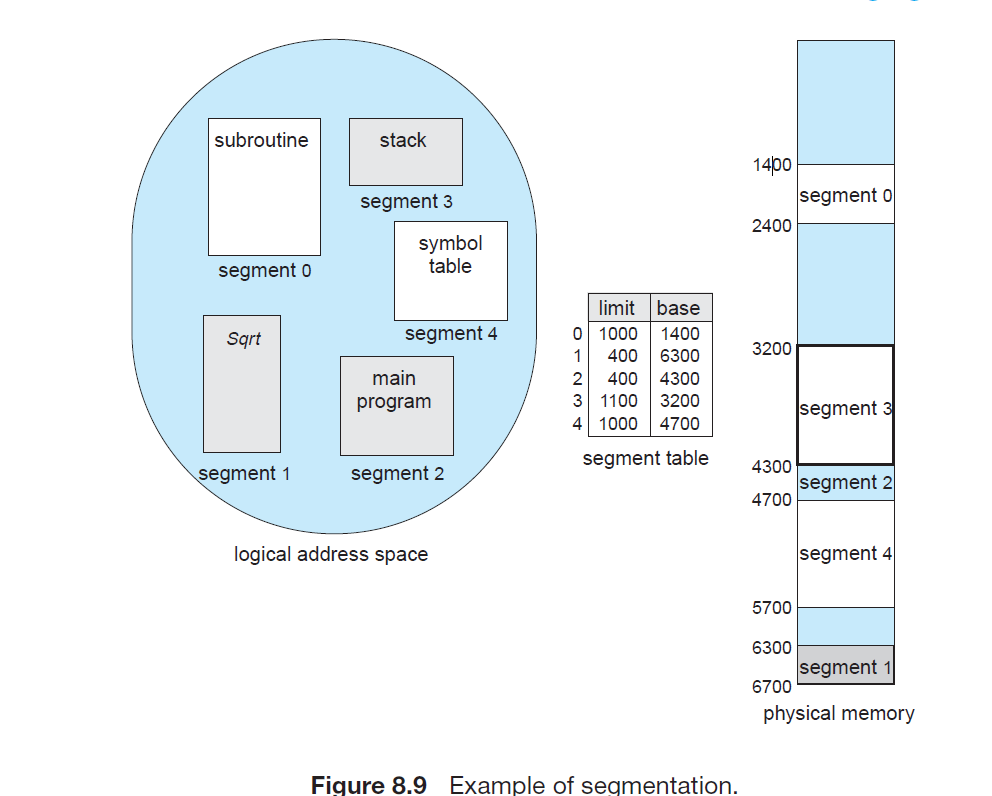
### Case-01: Segment Offset >= Limit

* 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.

**Example:**



We have five segments numbered from 0 through 4. The segments are stored in physical memory as shown. The segment table has a separate entry for each segment, giving the beginning address of the segment in physical memory (or base) and the length of that segment (or limit). For example, segment 2 is 400 bytes long and begins at location 4300. Thus, a reference to byte 53 of segment 2 is mapped onto location 4300 + 53 = 4353. A reference to segment 3, byte 852, is mapped to 3200 (the base of segment 3) + 852 = 4052. A reference to byte 1222 of segment 0 would result in a trap to the operating system, as this segment is only 1,000 bytes long.

## **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.
* It helps in the mapping of the two-dimensional logical addresses to the physical addresses.
* Segment table base register (STBR) stores the base address of the segment table.