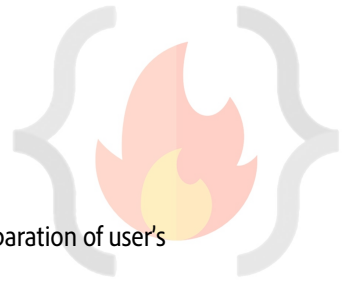
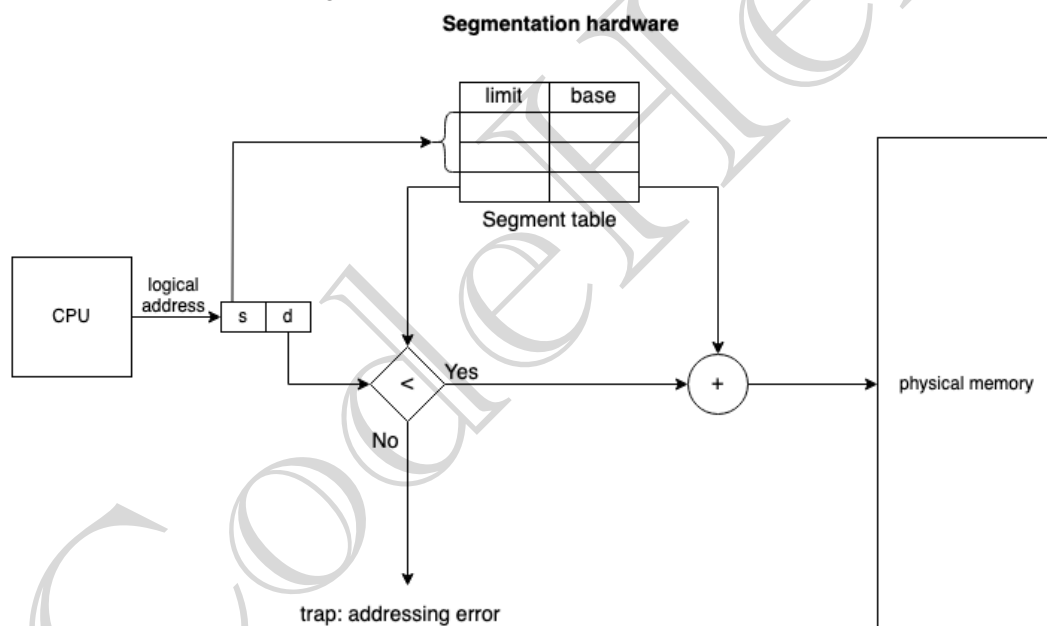


LEC-27: Segmentation | Non-Contiguous Memory Allocation



1. An important aspect of memory management that become unavoidable with paging is separation of user's view of memory from the actual physical memory.
2. Segmentation is memory management technique that supports the **user view of memory**.
3. A logical address space is a collection of segments, these segments are based on **user view** of logical memory.
4. Each segment has **segment number and offset**, defining a segment.
<segment-number, offset> {s,d}
5. Process is divided into **variable segments based on user view**.
6. **Paging** is closer to the Operating system rather than the **User**. It divides all the processes into the form of pages although a process can have some relative parts of functions which need to be loaded in the same page.
7. Operating system doesn't care about the **User's view** of the process. It may **divide the same function into different pages** and those **pages may or may not be loaded at the same time into the memory**. It decreases the efficiency of the system.
8. It is better to have segmentation which divides the process into the segments. Each segment contains the same type of functions such as the main function can be included in one segment and the library functions can be included in the other segment.



- 9.
10. **Advantages:**
 - a. No internal fragmentation.
 - b. One segment has a contiguous allocation, hence efficient working within segment.
 - c. The size of segment table is generally less than the size of page table.
 - d. It results in a more efficient system because the compiler keeps the same type of functions in one segment.
11. **Disadvantages:**
 - a. External fragmentation.
 - b. The different size of segment is not good that the time of swapping.
12. Modern System architecture provides both segmentation and paging implemented in some hybrid approach.