

# Segmentation

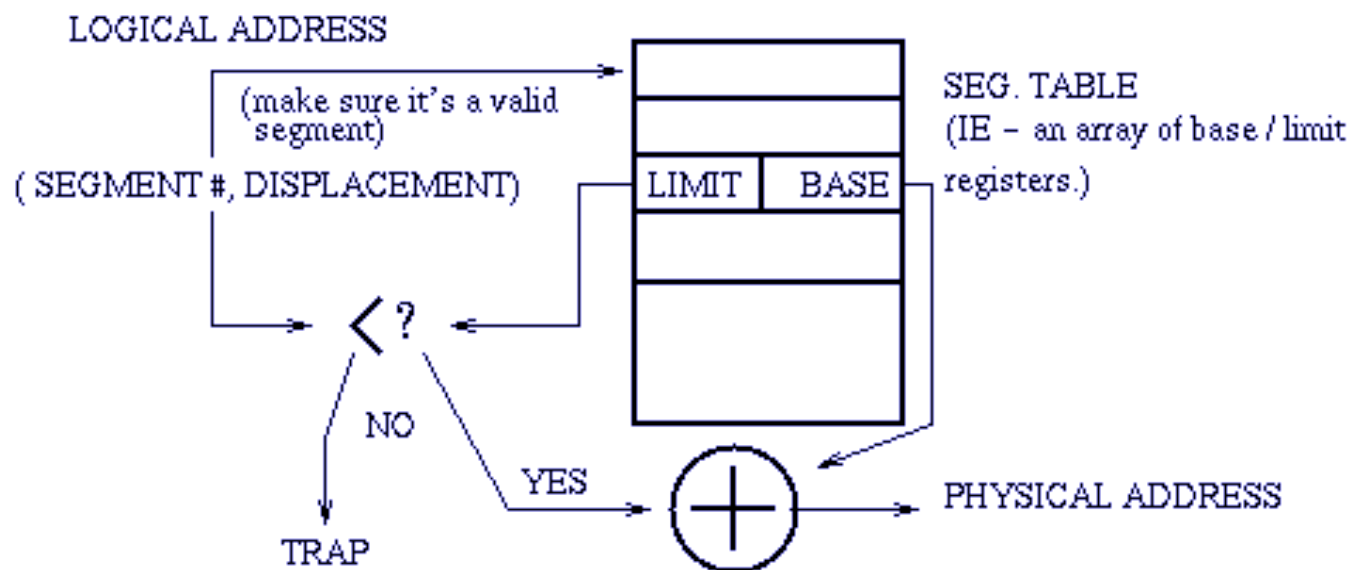
- Paging broke up our program into pages of a fixed size, another way to do things is to break our program into logical segments
  - E.g a subroutine, data structure, etc.
  - segment sizes vary!
- Main memory is no longer broken up into fixed-size chunks → just like MVT! The difference is:
  -
- Logically, a users program will consist of a # of segments
  - Each segment is contiguous, but the group is not contiguous

# Hardware Support for Segmentation

- Logical addresses for paging had a page number and displacement, logical addresses for segmentation have:
  -
- Instead of a page table we have a segment table. It contains the following:
  - 
  -
- Generation of logical addresses for segmentation must be done by the compiler/linker. Why?
  -

# Segmentation

An example of how to generate memory addresses using the segment table:

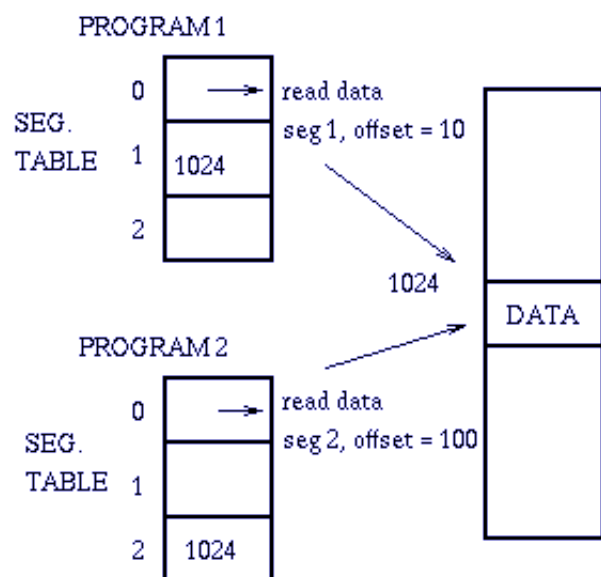


# Protection Made Easy

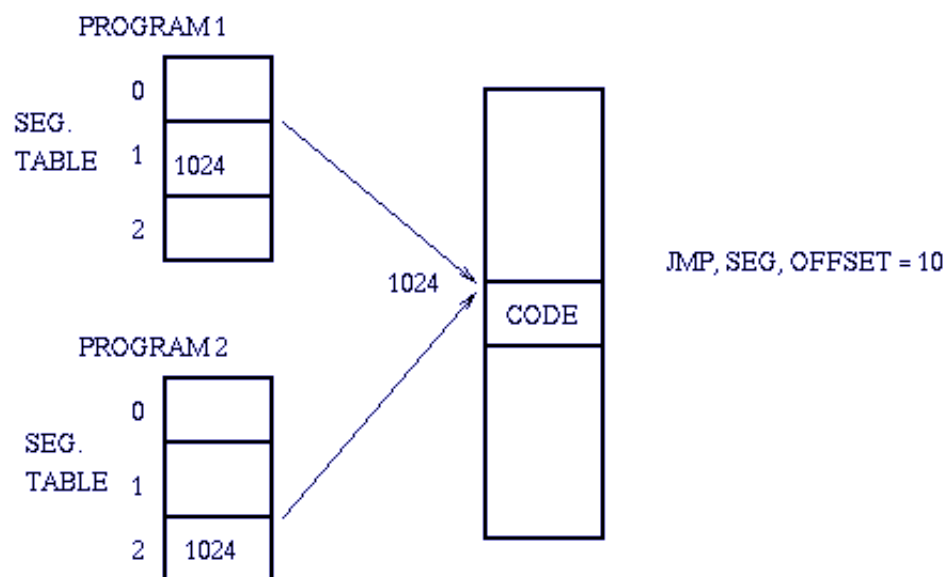
- Since each segment is a subroutine, data structure, etc., we can assign a protection mode to a segment
  - With paging, a page may be half subroutine and half data structure, so setting it to “read-only” for example would not make sense
  - With segmentation, it makes sense to protect these “logical” sections of the program
- Can represent these as follows:
  - 
  -

# Sharing: Data vs. Code

## Sharing Data



## Sharing Code



# Sharing

One solution to the sharing code problem is:

- If we want to share code, then that shared code segment must have the same number for each program using it
- For highly shared code we can reserve the segment numbers
- What about for paging, can we easily reserve page numbers for particular pages?
  -

# Fragmentation in Segmentation

- Since segments are of variable size, we will have fragmentation
  - Which kind?
- Degree of fragmentation depends on the segment size
  - If each job is one segment, then we have →
  - If the segments are smaller →
  - However, smaller segments →
- Compaction can reduce the problem
- We implement segment tables in the same way as page tables (i.e. cache/TLB)