Chapter 16 Segmentation

- The simple approach of base and bounds pointers is wasteful.
- There are problems trying to run a program with larger address space than physical memory with base and bounds.

16.1 Segmentation: Generalized Base and Bounds

- Segmentation was born to solve the problems of base and bounds
- Instead of having one base and bounds pair in the MMU have a base and bounds pair for each logical segment of the address space.
    - A segment is a contiguous piece of address space of a particular length.
    - There are three logically different segments in our address space:
        - code
        - stack
        - heap
    - Segmentation allows the OS to avoid spaces of unused memory by placing the segments rather than the whole address space.
        - Using a base and bounds pair per segment.
    - Requires a set of three base and bounds pairs.
- Segmentation fault is when a program tries to read an address that is out of bounds of the segment.

16.2 Which Segment?

- The hardware uses segment registers during translation.
    - Needs to know the offset into a segment and which segment the address refers to.
- One approach is the explicit approach.
    - Chop up the address space into segments based on the top few bits of the virtual address.
    - The bottom bits then tell the offset in the segment.
- The Implicit approach:
    - The hardware determines which segment a particular address is in.
    - Determines based on how the address was formed.

16.3 The Stack

- The stack grows backwards.
- Instead of using base and bounds only we need the direction of growth.
    - uses a bit as a direction assignment in the hardware.

16.4 Sharing Support

- Sometimes it is useful to share certain memory segments between address spaces.
    - Code sharing is common.
- Protection bits support code sharing in the hardware.

16.5 Fine V Coarse-grained Segmentation

- Coarse grained segmentation chops up the address space into relatively large chunks.
- Larger numbers of smaller segments is fine-grained segmentation.

    - Requires a segmentation table.

16.6 OS Support

- Segmentation raises some new issues:
    - What should the OS do in a context switch?
        - The segmentation registers must be saved and restored.
    - Managing free space in physical memory.
        - When a new address is created the OS has to be able to find space in physical memory for the segments.
        - Each segment may be a different size.
        - External fragmentation is when the memory has little holes of free space that make it difficult to allocate new segments.
            - One solution to this is to compact physical memory by rearranging the existing segments.
            - Another solution is to use a free-list management algorithm that tries to keep large extents of memory available for allocation.