

Segmentation (Ch 16)

Discussion Questions

- What is a sparse address space?
 - how do they affect simple (non-segmented) base and bounds virtual memory systems?
- Why does base and bound fail for processes with address spaces larger than physical memory?
- What is a segment?
- Why can't we simply add the base physical address of the heap segment to an address in the heap?
 - because heap VA space doesn't start at 0
- What is the difference between explicit and implicit segment references?
 - how do we get the offset for explicit?
 - segments laid out in VA so leading bits determine segment (segment slots)

- remaining bits are offset
- Why is the stack difficult to segment?
 - grows backwards (base is actually largest address)
 - explicit bits give us start of segment "slot", not base address
 - offset must be recalculated from end of segment slot
- Why would we ever want to share a segment across processes?
- Why do we need protection bits for shared segments?
 - why are protection bits only necessary if we have shared segments?
 - a process cannot access another processes segments without sharing
 - how does the OS prevent this?

- Where are protection bits for segments stored?
 - in address?
 - in OS?
 - in hardware?
 - why?
- What provides support for multiple segments OS or hardware?
 - both, hardware provides registers, OS does bookkeeping
- What is the difference between internal and external fragmentation?
- What happens if a segment needs to grow?
 - What if there is no room (contiguously)?
 - can this happen in simple b&b?