

# Operating Systems - Chapter 16 Summary

## Segmentation

- Base and bounds wastes memory when entire address space is relocated (unused gaps like stack–heap).
- Crux: How to support large, sparse address spaces efficiently.
- Solution: segmentation — multiple base/bounds pairs, one per logical segment (code, heap, stack).

## Segmentation: Generalized Base/Bounds

- Each segment (code, heap, stack) has its own base and bounds.
- Allows placing each segment independently in physical memory.
- Supports sparse address spaces, avoids wasting memory between segments.

## Example Translation

- Virtual addresses divided into segment + offset.
- Hardware picks correct segment using top bits, adds offset to base, checks against bounds.
- Out-of-bounds = segmentation fault (origin of the term).

## Stack Growth

- Some segments grow negatively (stack).
- Hardware tracks growth direction and computes negative offsets.

## Support for Sharing

- Protection bits allow read, write, execute permissions per segment.
- Enables safe code sharing across processes (e.g., shared libraries).

## Fine-grained vs. Coarse-grained Segmentation

- Coarse-grained: few large segments (code, heap, stack).
- Fine-grained: many small segments, requires segment tables (e.g., Multics).

## OS Responsibilities

- Save/restore segment registers on context switch.
- Grow/shrink segments when needed (e.g., sbrk system call).
- Manage free memory: external fragmentation is a challenge.
- Solutions: compaction (expensive) or free-list allocation algorithms (best-fit, first-fit, buddy).

## Downsides of Segmentation

- External fragmentation: free memory chopped into small holes.
- Still not fully flexible for sparse address spaces (large unused heap must still reside fully).

## Summary

Segmentation improves on base-and-bounds by dividing address space into segments (code, heap, stack), each with its own base and bounds. It saves memory, supports sparse address spaces, and enables code sharing. Hardware enforces protection and supports negative growth for stacks. However, segmentation introduces external fragmentation and still lacks full flexibility, leading to the need for paging.