Chapter 11 - Virtual Memory

- Fragmentation of contiguous allocation can be eliminated through compaction (Figure 11.1).
- Another technique is to avoid contiguous memory allocation.
- One way is to separate the program into different *segments*.
- A natural division is separating code from data. One could use two relocation registers (Figure 11.2).

- Rethinking Memory Allocation
  - Result is two separate physical address (Figure 11.3)
  - Why stop there? Why not 4 relocation (*a.k.a.* segment) registers? Figure 11.4 -- divide a portion of the 16-bit memory address into a 2-bit segment address and a 14-bit offset. Think of this as an array of base/bound relocation registers on a per-process basis.
  - Can now get more creative physical address assignments where the process appears to have contiguous memory (Figure 11.5).
  - Note that now there can be *holes* in the logical address space (Figure 11.6 & 11.7).

- Noncontiguous Logical Address Spaces
  - Varying the size of the segments can lead to misuse of the segment sizes, assuming that the number of segments is limited.
  - Why not just fix the size of the segments? Then, we can eliminate the "bound" column in the segment table and agree that all segments are of an equal, fixed size. We'll call these new fixed-size segments *pages*.
  - Page Table Entry - Pointer to physical memory frame.
  - Page table - collection of page table entries.
  - Again we divide the logical address into separate portions (4-bit page and 12-bit offset) - Figure 11.8.