Chapter 9 - Binding

- Program is loaded to create a process
- Processes can occupy any available memory
- How are addresses generated?
  1. Compile time
  2. Runtime
  3. Execution time

Chapter 9 - Binding

- Dynamic loading of routines
- Dynamic binding
- Overlays

Chapter 9 - Virtual and Physical Addresses

- CPU generates logical addresses
- Memory unit generates physical addresses
- Logical addresses $\rightarrow$ physical addresses

Chapter 9 - Swapping

- Process must be in memory to execute
- Process can temporarily be moved to secondary memory
- What influences swapping
  1. Binding
  2. Amount of memory
  3. Incomplete system calls

Chapter 9 - Memory Allocation

- Single vs multiple partitions
- Allocation strategies: First fit, Best fit and Worst fit
- Internal and external fragmentation
- Compaction

Chapter 9 - Paging

- Possible solution for fragmentation
- Physical memory divided into frames
- Logical memory divided into pages
- Logical addresses divided into page number (p) and offset (d)
- p is used as an index into a page table
- Table contains start addresses of physical frames
- Start addresses of frames and offsets are combined to form a physical address which is transferred to the memory unit

Chapter 9 - Paging

- Size of frames depend on hardware
- Typical size expressed as power of 2
- Example: Logical address space of 2^m and page size of 2^n. Top $m-n$ bits of logical address yields p while bottom n bits specify the offset
- Internal fragmentation still exists
- Page tables must be kept for every process: Context switching more expensive

Chapter 9 - Hardware

- Implement page table as a set of registers. Not efficient for large tables
- Solution: Store table in memory. Single register points to the start of the table (PTBR)
- Disadvantage: Every memory references requires additional access to memory
- Solution: TLB
- TLB translates page number to frame reference
- Context switches flush the TLB
- Calculation of effective access time

Chapter 9 - Protection

- Every frame contains protection bits
- Control: Read, write, read/write
- Invalid operations will be trapped by hardware and operating system
- Valid bit: Controls access to a specific page

Chapter 9 - Multilevel Approach

- Logical address space too large: 2^{32}
- Example: 4K page size inside 32-bit address space requires 2^{20} entries. 4 bytes per entry implies that 2^{22} bytes are needed to store page table
- Solution: Use a separate table to locate page tables by dividing page number
- Disadvantage: Every level of indirection requires additional memory references, decreasing performance
- What effect does this have on context switching?

Chapter 9 - Segmentation

- Programmer's view vs operating system's view of memory
- Programs contain a logical structure with regards to code and data
- Segmentation supports this view
- Logical addresses divided into segments and offsets

Chapter 9 - Segmentation Hardware

- Logical address still maps to physical address
- Segmentation table: Entries describe attributes of segments
- Attributes: Base, limit and access rights
- Segmentation table base register (STBR)
- Segmentation table length register (STLR)