CIS3110 Lecture Summary - February 6, 2025

Virtual Memory and Page Replacement Algorithms

Virtual Memory Address Translation (Main Memory 9.1-9.5, Virtual Memory 10.1-10.6, 10.8)

The lecture began with a detailed walkthrough of virtual memory address translation. Key points included:

1. **Page Offset Basics**:
   - Page offset is the number of bits making up the page size
   - Example: 512 byte page size = 9 bits of offset; 2048 byte (2KB) page size = 11 bits of offset (0x800)
   - Page size is fixed by CPU manufacturer and is the same for all programs on that CPU
2. **Virtual-to-Physical Address Translation Process**:
   - When a program accesses memory at virtual address (e.g., 0x77FF), the system divides by page size
   - The division separates the address into page number and offset
   - The page number is used as an index into the page table
   - The page table provides the frame number where the page is stored in physical memory
   - The offset is combined with the frame number to produce the final physical address
3. **Page Table Entry Details**:
   - Contains several bits that track page status:
     - Valid bit: indicates if page is in memory
     - Read-only bit: determines if writes are allowed
     - Dirty bit: set when page is modified, indicating it needs to be saved to disk when removed
     - Use bit: set whenever page is accessed (read or write)
   - Page table maintains the mapping between virtual pages and physical frames
4. **Memory Organization**:
   - Main memory contains frames from multiple processes
   - Each process has its own page table with mappings to its allocated frames
   - Virtual memory presents continuous memory space to programs, physical memory can be fragmented

- Walking from one virtual page to the next doesn't mean walking through contiguous physical memory

Page Replacement Algorithms (Virtual Memory 10.1-10.8)

The second part of the lecture focused on page replacement algorithms, which are critical when a process needs a page that isn't in memory:

1. **Page Fault Handling**:
   - When a process needs a page not in memory (valid bit = 0), a page fault occurs
   - Page replacement algorithms determine which page to remove to make room
   - The choice of algorithm significantly impacts performance by reducing the number of page faults
2. **Working Set vs. Resident Set**:
   - Working set: pages a process needs at a given time
   - Resident set: pages a process actually has in memory
   - Ideally, resident set = working set, but this isn't always possible due to memory constraints
3. **Page Replacement Algorithms Compared**:
   - **Belady's Optimal Algorithm**:
     - Theoretical algorithm that looks into future to know which page won't be needed for longest time
     - Provides optimal (minimum) page fault count but impossible to implement in practice
     - Used as benchmark to evaluate other algorithms
   - **FIFO (First-In, First-Out)**:
     - Simplest algorithm - replace the oldest page
     - Doesn't consider usage patterns
     - Performs poorly compared to other algorithms
   - **LFU (Least Frequently Used)**:
     - Replaces the page that has been accessed the least number of times
     - Requires counters for each page, increasing memory overhead
     - Can be problematic when pages used heavily initially are never replaced
     - Often implemented with periodic counter resets to avoid this issue
   - **LRU (Least Recently Used)**:
     - Replaces page that hasn't been used for the longest time
     - Better approximation of optimal algorithm as it assumes temporal locality

    - Often implemented using linked lists (moving recently used pages to front)
    - Provides good performance with reasonable overhead
  - **Clock Algorithm**:
    - Approximates LRU but with lower overhead
    - Uses the hardware-managed use bit
    - Visualized as a clock hand moving around frames, clearing use bits
    - When page replacement needed, selects first page with clear use bit
    - Enhanced version considers both use and dirty bits to minimize disk writes

4. **Sample Comparisons**:
   - Using same reference string and 3 frames:
     - Belady's optimal: 11 page faults
     - LRU: 12 page faults
     - LFU: 13 page faults
     - FIFO: 15 page faults (mentioned but not demonstrated)
   - Adding one more frame significantly reduced faults for Belady's optimal

The lecture concluded with a reminder about the upcoming midterm exam on Tuesday and instructions regarding calculators and marking tools for the exam.