Module 5

Virtual Memory

- ▼ Why use virtual memory?
  - In order to have more memory than physically needed, we use virtual memory, which is a combination of hard drive space and RAM.
  - It then acts like memory that processes can use. With virtual memory, a system can load larger programs, or multiple programs running at the same time, operating as if it has infinite memory, without having to upgrade any hardware.
- ▼ How does virtual memory transfer data?

  paging or segmentation
- ▼ How does virtual memory paging work?

  Virtual memory paging works similarly to simple paging of regular memory, with the big difference being demand paging, which means that not all pages of a process need to be in main memory frames for the process to run, but may be read in as needed.
- ▼ How does virtual memory segmentation work?

  it reads the segments in as needed, instead of loading them into main memory all at once
- ▼ What are the benefits of combined paging/segmentation systems?
  - The efficiency of paging with the protection and sharing capabilities of segmentation.
  - Each segment of a process is divided into pages, so the virtual address space of a process is logically divided into segments, but each segment does not need

to be contiguous in physical memory.

- This method has all of the advantages of both methods, with the exception of internal fragmentation, which can still happen to some degree.

▼ What is a page fault?

When a running program accesses a memory page that is not currently mapped into the virtual address space of the program, a page fault occurs, which adds a lot of overhead to the process.

▼ What is a fetch policy?

- Determines when a page should be brought into main memory.
- The typical methods are demand paging, where a page is brought into main memory when a reference is made to a location on that page, and prepaging, when other pages are brought in even if they are not referenced.

▼ What is a replacement policy?

- Decides which page in main memory to replace with the new page that must be brought in.
- Commonly used algorithms include First-in-first-out (FIFO), which has low overhead, is simple to implement, and is very intuitive, but does not have great performance, and Least recently used (LRU), which replaces the page in memory that has not been referenced for the longest time, so it performs very well, but is difficult to implement.

▼ What is a cleaning policy?

- Deals with when a modified page should be written out to secondary memory, so it is the opposite of a fetch policy.
- The typical methods are demand cleaning, where a page is written out only when it has been selected for replacement, and precleaning, where modified pages are updated on secondary memory before their page frames are needed so that pages can be written out in batches.

▼ How can memory be categorized?

1. Kernel memory, which the OS needs to manage itself.

2. Private memory, which is used by user programs.
3. Shared memory, which can be used by multiple programs simultaneously.

▼ Why use shared memory?

- Shared memory enables programmers to be efficient with their code, creating libraries of common functions that can be used by many programs.
- Only one copy of the common code needs to be loaded into memory, and all of the running programs can share that copy.
- In addition, processes that pass large amounts of data to one another can do it faster when it is in shared memory.

▼ What is shared virtual memory?

- A shared virtual memory is a single address space shared by a number of processors.
- It is an extra layer to the virtual memory paging model.
- Despite the additional overhead and design considerations, shared virtual memory makes it easy for processes to share memory by simply referencing the same physical pages.