Name: Duc Anh Nguyen

Course: Operating System

Date: 3/5/2023

Learning Reflection 1

1. Reflection Prompt 1:
    - Operating systems use processes, context switching, and syscalls (traps) to manage resources and provide an environment that application can efficiently run.
    - Processes:
        - Processes is an abstraction that provided by the operating system of a running program.
        - Each process will have its own memory space and resources allocated by the OS upon its creation and released after its execution.
        - The operating system needs to schedule the processes to maximize CPU utilization and resources.
        - Processes can communicate with each other through inter-process communication with mechanism like piping.
        - Processes use mechanism like lock to synchronize their activities without having any conflicts and ensure data consistency.
    - Context Switching:
        - Context switching is a mechanism that save the current state of a process and restore the state of another process.

    - The state of an executing process includes register values, program counter, and stack pointer.
    - The mechanism allows multiple processes to run concurrently in a single CPU.
    - The process of doing a context switch includes saving the current state, selecting the next process, restoring the process's state, and updating the process control block.

- Syscalls (traps):
    - A system call is a mechanism in operating system that enable user-level process to perform kernel-level operations like calling an I/O or handling interrupts. The set of system call is map to specific kernel services.
    - Traps is when the a user-level process execute a system call, the processor will push the program counter, flags, and a few other registers onto a per-process kernel stack. The CPU then generates a trap, which switch the operating mode to kernel mode. When in kernel mode, the kernel validates the system call parameters and performs the requested action. It then returns back to user-mode, pop the value off the stack, and resume execution.
    - Syscall (traps) provides security in the operating system the prevent any processes executing I/O or any privilege operations. Moreover, it prevents user-level process from accessing kernel memory.

2. Reflection Prompt 2:

- If I have to implement the context switching and system call mechanism, I might not be able to do it because there are a lot of knowledge in both software and hardware before building such mechanism.
- Require research needed to be done:
    - Processor Architecture: Every processor has its own architecture and instructions set, so understanding the instruction set, processor modes, and the register can be beneficial to work on mechanism like context switching.
    - Interrupt Handling: research interrupt handling is necessary because in order to apply system call, we need to understand how interrupts are generated and how the OS should respond each time there is a system call. It would include knowledge of hardware interrupts, software interrupts, and exceptions.
    - Memory Management: Since memory is one of the key components of system call and context switching, more research needs to be done in order to implement these mechanisms efficiently into the OS.

3. Reflection Prompt 3:
    - Process Scheduling is the mechanism used by the OS to manage the allocation of CPU resources to running processes. It creates an illusion that multiple processes are running concurrently and brings a better performance to the computer system.
    - There are different several scheduling algorithms to schedule processes like round-robin, priority-based, and lottery scheduling. Each has its own advantages and disadvantages, and the choice of scheduling largely depends on the specific needs of the system

- Preemption is a key component of process scheduling as it is the ability to forcibly interrupt a running process and allocate the CPU to another process. It ensures fairness and preventing process starvation in the execution process.

4. Reflection Prompt 4:
   - If I were given enough time and effort to implement scheduling algorithms, I would be able to do it. Beside the knowledge of scheduling algorithms, another knowledge that I need to implement:
   - Processor Architecture: The processor architecture always plays a key role in implementing this design because we are using its instruction set, processor modes, and register.
   - Data Structure: Different Scheduling algorithms require different data structures to manage the state of the running processes, so we need to understand of these data structure to implement the algorithms efficiently.
   - Time Management: Some algorithms like the Round Robin requires the use of a timer interrupt to enforce time slices so if the time slice is too short, the CPU overhead of context switching might be significant.
5. Reflection Prompt 5:
   - Virtual memory is a mechanism in operating systems that provide each process with its own virtualized memory space. This prevents processes interfering with each other and accessing each other's memory.
   - It create an abstraction between the physical memory and the memory set by the processes. It allows the OS to manage physical memory efficiently.

- There are different ways for the OS to create a virtual memory for the process. Two of them discusses in the class are segmentation and paging, with paging is the more popular one among modern operating systems.
- Memory swapping allows the OS to swap pages of memory between the physical memory and disk storage. It helps OS handle more processes than the available physical memory.
- Memory Sharing allows multiple processes to share the same memory pages, and memory protection is the OS trying protect the memory pages from invalid access.

6. Reflection Prompt 6:
   - Virtual memory concepts like page tables and page fault handler are difficult concepts that I might not be able to do if given time and effort. These are the knowledge that I need to more research on to implement them:
     - Kernel Architecture, process management, and interrupt handling are the key concepts to grasp while managing virtual memory. These concepts have close coordination with each other so building a page table and page fault handler while dealing with these mechanisms might bring up the complexity.
7. Reflection Prompt 7:
   - The key concepts that I have not been able to fully understand are the concepts in Page Fault Handling as I only understand the reason why page fault handling is important. One of my most confusing part about page fault handling is how it can determine the missing page. Moreover, what if thrashing happen as there are more active processes than the physical memory can handle, and what does the operating system do during the process.

8. Reflection Prompt 8:
    - One of the most thing that I found helpful during the class to help me learn is the class readings and the discussion afterwards because the reading helps me understand the core concepts in the class while the discussion helps me brainstorm the idea that I have been reading.
    - The API reading given in class is quite useful, but those are the readings that I have been skipping. I should focus more on those readings because it helps me broaden my programming skills while strengthening my knowledge of the concepts studied in class.
    - I have also not watched enough of the optional videos because I read the readings, but some of the confusing parts in the reading are explained in the optional videos.
9. Reflection Prompt 9:
    - Reflection Prompt 1 contributes the most to help me learn and revise the concepts that I have been learning in the past weeks. It is better than other prompts because it focuses more on details of concept of scheduling like syscall, process, and context switching. Other prompts are useful too, but it would be better if we have more particular question regards of that topic.
    - Even though all the reflection prompts are useful, I feel like reflection prompt 3 should be more specific like asking on how each scheduling algorithm works or what would be a performance metric to decide a scheduling algorithm. These questions are particular and provide a good insight into how to pick the right scheduling algorithm.