CMPT 300 Midterm Review

Coverage: Lecture 1 – Lecture 18

Introductory topics

- Basic computer system organization
- Interrupts
- I/O structure
- Basic storage structure
- System calls
- Hardware protection
- Operating System Design (monolithic vs microkernel, virtual machines)

*Won't be asked a direct question about anything in this section but it's good to know it to understand the later topics

Processes

- Process states
- Process model of an operating system
- Precedence and concurrency relationships…
- Critical sections
    i. Software solutions to the CS problem
    ii. Hardware solutions to the CS problem (*IMPORTANT understand test and set)
- Semaphores
    i. Producer-consumer problem
    ii. Semaphore implementation using P and V

- Monitors
    i. Monitor implementation using semaphores

*Important to understand how different implementations relate to each other even though they basically achieve the same thing!

*For software solutions, consider the case where there are more than 2 processes (unlike the example on the notes

*Question that might be asked: Why would you use a busy wait (software solution) over a hardware solution?

Answer: If we know the busy wait won't be very long, it's a good idea to use it.

Critical sections cont'd

- IPC
  i. IPC semantics
  ii. Message capacity of an IPC system
  iii. IPC message types (fixed-size, variable-sized and typed)
  iv. Lost messages
  v. Remote procedure call
  vi. Administrator-worker model of IPC (*important to understand)

*It won't be useful to memorize monitor implementation using semaphores. It will be more useful to actually understand what's happening in depth

- Multi-programming
- Process scheduling
  i. First come first serve (non-pre-emptive)
  ii. Shortest job first (non-pre-emptive)
  iii. Shortest remaining time first (Pre-emptive)
  iv. Round robin (pre-emptive)

*Could get a question where we get a ready queue and get asked which order they execute as well as what is the average turnaround time

*If the arrival time of the processes is not explicitly stated, then we can assume that all the processes arrive at the same time (important for calculating the turnaround time)

-Multi-level queues
-Multi-level feedback queues

Memory management

- General memory management
  i. Fence for memory protection

-Compiling, Linking and Loading
  i. Static vs dynamic relocation (should spend a good amount of time carefully understanding this!)

-Swapping (goes both ways)

-Contiguous memory allocation
i. Upper/lower or base/limit registers

ii. multi-programming with a fixed number of tasks (MFT)

- Fragmentation

*There is overhead with dynamic relocation

For question 7 on the sample midterm: We have to carefully realize that this looks quite like the critical section solution examples from the notes with one key difference. The main difference is that the processes are setting the turn variable equal to their turn (e.g turn = i) during their execution in the sample midterm example.

In the lecture notes example, process j would set the turn variable equal to i (turn = i) when its finished its turn.

For question 8 on the sample midterm: we must assume that P and V are not atomic because we are NOT using the system's library functions (we must implement P and V). This means that there is actually a critical section between P and V themselves we must solve on top of what we're being asked to do.

The "lock" Boolean variable is for solving the critical section between P and V whereas the "blocked" variable is for the busy waits we must implement.