CISC 3320 Midterm Review Sheet.

The format of the midterm is open notes, i.e., you are allowed to bring one sheet of notes (letter-sized, double-sided is ok but it must be in one piece and no flaps). The test will include True or False, multiple-choice, and short questions and answers. The scope will cover Ch1~6.

Please note that this list is not meant to be exhaustive due to the nature of T/F and multiple-choice questions, but it should cover the vast majority of topics to be tested.

Ch1

what do operating systems do, their goals, components of a computer system, what are attached to the system bus, bootstrap program, volatile vs nonvolatile storage and their roles in reading data from I/O, functions of interrupts and how they are handled, storage-device hierarchy, multiprogramming vs multitasking, user-kernel mode, program counter, characteristics of real-time system, what computing environments are there, services used in cloud computing.

Ch2

operating systems services (user interface, program execution, … etc.), what are system calls, OS structures and their characteristics comparison, how modern OS's adopt them.

Ch3

program vs process, parts in a process address space, states of process and their transition, PCB and context switch, how processes are launched in Unix/Linux (be specific on it steps), waiting on child process, android process importance hierarchy, synchronous vs asynchronous communication, rendezvous.

Ch4

benefits of threading, what are shared among threads, threading model choices and what do modern OS's use, Amdahl's law (its essence and basic calculation), Pthreads API specification vs implementation, common Pthreads functions, how threads are canceled, local variables vs thread-local storage.

Ch5

events for preemption, basics of CPU scheduling algorithms, which of them are preemptive, criteria of them, being able to draw Gantt charts for common schedulers and calculate their wait times (study the examples shown in the slides), Linux's choice for scheduler and its features.

Ch6

does Peterson's solution always work? Hardware supported solutions, Mutex and Semaphore, busy waiting vs sleep waiting.