**CSC 345 Final Exam Study Guide**

**May 1, 2014**

The final exam is scheduled for Monday, May 5th 10:30 a.m. – 1:20 p.m. in Forcina. Section 1 students are scheduled for Forcina 409 and Section 2 students are to use Forcina 407. I will be in one classroom for the first hour, then will change locations with Ms. Zsilavetz for the second hour, and then will switch again as needed. If you have a question that can not wait for my arrival, please ask Ms. Zsilavetz to permit you to leave the room to visit me in the next door classroom. I believe this arrangement will give you ample opportunity to seek any guidance that you may need in a timely manner.

As per department policy, you are to use the rest room prior to the exam, not during the exam.

**This guide supplements the review questions distributed earlier this semester.**

Define and explain the significance of each of the following terms:

aging

blocked state

cold start

critical section – something that can only be accessed one at a time

defragmentation

degree of multiprogramming - number of processes in a system, number of threads in a program

device driver

external fragmentation

I/O bound

indefinite postponement – system is too busy allocating for other requests the current request is never met

inode - describes a file system object

internal fragmentation

inter-process communication

kernel

monitor – has both mutex and ability to wait for a condition to be met

mutual exclusion

process control block

quantum – time slice

rotational optimization

round robin scheduling – gives all items equal time slices

safe/unsafe states – unsafe exceeds the capabilities of a system

seek time

throughput

turnaround time

working set of pages

Compare and contrast processes and threads. Consider their life-cycles, their use and coordination of system resources, and their difference of purpose. How do you determine when to create a process or to use threads?

* Process can contain threads, used as the core application
* A thread can do anything a process can do but there can be multiple threads inside a process
* Thread has shorter life cycle in general
* Threads used for smaller tasks
* Creating a user instance of a program

Summarize process scheduling as supported in the Linux operating system. Discuss how each of the Linux scheduling algorithms handles the various performance criteria (optimization criteria). How can knowledge of the scheduler benefit you as an application programmer (not as an OS programmer)?

* Short term scheduler
  + Also known as CPU scheulder
  + O(1)
    - Designed to make a constant time for processes
  + CFS
    - Uses red-black tree
      * Automatically balances tree
      * Stores records about planned tasks in tree
      * Stores time spent on process
      * Gives time to process that has gotten the leas time
      * Returns item to tree
      * Takes tasks from left side of the tree
    - O(log n) time
    - Uses “sleeper fairness”
      * Sleeping for waiting tasks
* Medium term scheduler
  + Remove process from main memory and places them on secondary memory
  + Also known as “swapping in” or “swapping out”
  + Decides to swap out a process which has not been active for some time or one with low priority
  + Also swaps out a faulting page
  + Swaps out program using large amount of memory to make space, swaps back in later
* Long term scheduler
  + Decides what goes into the ready queue
  + Determines the degree of concurrency
  + Controls degree of multiprogramming

Review the difference between CFQ and CFS (common error on exam 2).

* CFS – refer to previous
* CFQ
  + Places synchronous requests submitted by process into a number of per-process queues
  + Allocates time slices for each queue
  + Time slice length and number of requests in a queue is allowed to submit depends on the I/O priority of a given process
  + Asynchronous requests for all processes are batch together in a fewer queues
  + Does minor anticipatory I/O scheduling

What is the most important resource for an operating system to manage? Provide supporting arguments. How would you respond if someone were to counter suggesting that a different resource was the key component (consider this question for each of the resource categories).

* Debate between memory and CPU
* Theoretical for GPU

Describe three basic functions of a file system and highlight how these are supported in the Linux/Unix environment. Identify any significant differences in the Windows environment.

* Allow for access of files
* Manages files permissions
* Creates order
* Linux can delete open file
* Both use MBR
* Windows uses a file allocation table Linux uses superblock

Explain what leads to thrashing, how the OS might detect it, and how thrashing can be eliminated. What are the costs involved?

Name and define common measures of efficiency of resource utilization, system performance, and user service.

* Constantly swapping a page in and out of memory causes thrashing
* Locality is used, there is a lot of overhead involved with eliminating thrashing
  + Not often implemented due to this overhead
* Time for task to complete
* Time spent in memory
* Degree of multiprogramming

Draw a Gantt chart diagram depicting CPU use and I/O handling in a multiprogrammed system currently handling 3 processes. Discuss each key section of the diagram. State your assumptions about CPU scheduling (consider priority, preemptive and non-preemptive, etc., in different diagrams).

What process scheduling capabilities are expected of contemporary operating systems? What tradeoffs should be considered?

* Reasonable i/o performance
* Responsiveness to user
* Rapid processing
* Speed vs memory
* Energy efficiency
* Many can quickly become outdated DDR4 is on the rise PCI SSD

Why is the support of concurrency a vital topic in today’s computing environments? Why is there renewed interest today?

* Graphics Processing
* Rapid data processing
* Multi-core systems

Write pseudocode for classic synchronization problem classes, such as readers and writers, and bounded buffer (consider single producer/consumer, single producer with multiple consumers, and multiple producers and consumers).

How does a monitor structure provide synchronization? Is concurrency supported or does the use of a monitor serialize the code? Support your assertion.

* Monitor is a synchronization construct that allows threads to have both a mutual exclusion and the ability to wait for a condition to be met
* Monitor has a mutex object and condition variables
* Condition variable is a container of threads that are waiting on a certain condition
* Provides a mechanism for threads to temporarily give up access in order to wait for some condition to be met before regaining exclusive access and resuming their task

How is a page fault generated and identified? How is a page fault handled? Be specific about the structures that are used and discuss any strategies required. Consider the choices that must be made in determining algorithms that the memory management unit employs.

* If the requested page is not in memory it needs to be called for
* Identified by looking at the page table LRU vs FIFO vs ARC
  + ARC uses two lists in T1 and T2
    - T1 recent cache entries
    - T2 for frequent entries referenced at least twice
  + Ghost entries B1 and B2
    - B1 ghost entries recently evicted from the T1 cache but are still tracked
    - B2 similar entries but evicted from T2

In designing an operating system, we want to facilitate fast searching and access to files. Diagram a directory organization that provides fast searching and outline some techniques (hardware and/or software) used to facilitate fast access.

* Make a similar structure to a Linux system that treats everything as a file

Prepare a reflective essay on how this course has advanced your preparation for a career in computing engineering or computer science. What are the top three concepts you have learned that have strong pertinence to your computing foundation? Include discussion on two projects from this course to include in your portfolio.