

Final Exam Notes

MPCS 51082 – Introduction to Unix Systems

November 27, 2023

Final Exam Structure

The exam will consist of the following types of questions:

- Short Answer Questions (which may have multiple parts)
- Be a Computer Questions (Given some code what is the output of the code.)
- Short Programming Questions (i.e., given a problem, provide the C code that solves this problem)

You will be given 120 minutes to complete the exam. The final exam will be closed-everything (no books, notes, cheat-sheets, etc). The exam is a fully written exam that I provide (no computers, etc.). I will provide any scratch paper if requested. Make sure to bring a pencil or pen.

Final Exam Topics

The final exam will all material in **Modules 3-5**. Module 3 included two separate slides (part1 and part2) so make sure to review both sets. You do not need review modules 1 and 2; those topics, will no be included on the exam. No bash scripting will be on the exam; however, you do need to understand any commands we discussed in the specified modules. Although all material in modules 3-5 are fair game, please pay close attention to these specific topics:

- Understand C pointers and the difference between static v.s. dynamic memory allocation
- Exceptions and Syscalls
 - What is an exception and how are syscalls related to exceptions?
 - Asynchronous Exceptions
 - Synchronous Exceptions
 - Understand the process of how a syscall is executed (slide 102 of m3_part.pdf)
- Process Control and Management
 - Multiprocessing
 - Concurrent vs Sequential processes
 - Process creation
 - Process suspension
 - Process terminal
 - Processing waiting (wait) and execution (execve)
- Understand how to create a process graph given a concurrent C program.
- Understand the Process states and how a process can transition from one state to another.

- Memory management in a UNIX kernel.
- Interprocess communication via using signals
 - Understand what are signals and why they are necessary
 - Process group and session
 - Signal handling
 - Waiting for signals
- How does UNIX kernel represents open files
- Access Control
 - User/Groups
 - File permissions
- File systems
 - Understanding the basics
 - Structure of a hard-disk: Partitions, Volumes, Inodes, etc.
 - Hard vs Symbolic links
 - The Virtual File System (VFS)
- For module 5, do not worry about understanding slides from 65 and on. Those topics will not be covered on the exam.

How to Study for the Exam?

Unfortunately, this one of the first iterations of the course so I do not have practice problems. However, I think the best way to study for the exam is going through the lecture slides. You can reference the book for more in-depth information about a certain topic if you're still confused. If you're comfortable with the material presented in the lecture slides along with any companion material than you should be a good shape for doing well on the exam. Here's some additional tips

- I looked at this website <https://www.w3resource.com/c-programming-exercises/> and it has C problems to help you along with implementing C programs. Take a look at the following exercises with these sections (Note: you won't be able to do all exercises in those sections and that's okay):
 - Basic Algorithm
 - String
 - Array
 - Pointer
 - Conditional Statement
 - Input, Output
- Make sure to review the lecture videos. For each video, I took the time to provide time segment so you can skip to the material you need more clarification on instead of watching the entire video over again.
- Make sure you can generate the process graphs from the code presented in M4. Try to just look at the code without looking at the process graph next to it.
- Make sure you review your C syntax. For the coding problems, you can still get the majority of the problem correct even with bad C syntax. I care more about the algorithm than fully correct C syntax.