Midterm Exam
CS 111, Principles of Operating Systems
Fall 2017

Name: Cody Hubbard

Student ID Number: 004 843 389

This is a closed book, closed note test. Answer all questions.

Each question should be answered in 2-5 sentences. DO NOT simply write everything you remember about the topic of the question. Answer the question that was asked. Extraneous information not related to the answer to the question will not improve your grade and may make it difficult to determine if the pertinent part of your answer is correct. Confine your answers to the space directly below each question. Only text in this space will be graded. No question requires a longer answer than the space provided.

1. In a system using modern virtual memory techniques, what is the relationship between a page and a page frame?

A page is the virtualization of physical memory that is allocated/distributed to a process as its address space. A page frame is the actual location in physical memory of a page.

2. Why are operating system ABIs of importance for convenient application software distribution?

ABI's allow software to be distributed to compliant systems/hardware without needing recompilation. That is, ABI's are user friendly as users do not need to compile any given software that is run on their system.

3. Why is information hiding a good property in an operating system interface?

Information hiding is a good property because it helps enforce the principals of encapsulation and protection between processes.

4. When an operating system performs a context switch between processes, what information must the OS save?

Most usually the contents of the processes' registers as well as its state. Also some parts of its address space may be saved, its page table and/or TLB depending on the memory virtualization method being used.

5. What is the purpose of a trap table?

The purpose of the trap table is to hold the different instructions or required behaviors that correspond to exceptions raised by a process.

That is, anytime an exception/trap is raised the OS determines what type and then the hardware consults the trap table to determine what the necessary action.

6. What is a race condition?

A race condition is the problem that occurs when concurrent operations cause non-deterministic outcomes. That is, the order of operations by competing processes may cause incorrectness or unwanted behavior. These problems are usually caused by unfortunate interrupts during critical sections of code or multiple processes/threads entering critical sections at the same time.

7. Why is blocking a problem for user-mode threads? Why isn't it a problem for kernel-mode threads?

Blocking is a problem for user mode threads because when a single user-mode thread is blocked all of the corresponding threads in that process are also blocked. The scheduler does not know other user-mode threads even exist so it cannot run them. However Kernal mode threads do not have this problem as the scheduler is well aware of their existence and can switch to them. I should also mention that user-mode threads cannot take advantage of multi-processor / multi-core / multi-thread systems so when a user-mode thread blocks it blocks the entire core running that process, not the case with kernal-mode threads as they can be run on different cores / processors.

8. Why does Shortest Time-To-Completion First (STCF) scheduling require preemption?

Because STCF scheduling requires the ability to interrupt a running process so that a new process, which may have just arrived, with an estimated shorter completion time can be run instead. By "shorter completion time" I mean that the new process has a shorter total running time than a current process's remaining running time.

9. When a Unix-system follows a fork with an exec, what resources of the forked process are replaced?

The exec will replace most if not all of its resources. The CODE segment, the heap, the stack, all of its registers & PS/PC.

10. What form of fragmentation do we still suffer if we use a paging memory management system? For a segmented paging system, how much fragmentation per segment do we see?

~~F~~FIXED SIZE SEGMENTS

The paging memory management system still suffers from internal fragmentation, usually at about ½ page per process.

For a FIXED SIZE SEGMENTED memory management system we usually see an internal fragmentation rate of about ½ of each fixed size segment. It also still suffers from external fragmentation.