This is a practice final exam. The final is during a three-hour block during the final exam period and will be 18 questions (+/- 10 minutes per question). The exam is largely conceptual, so if you would like feedback on your answers, you can submit your exam to Gradescope by November 20th and I will grade it as if it were a final so that you can know what to expect.

Name: _____

Booting a Device and System Calls

[Q1] On most modern devices, there are several steps that need to occur before the user is prompted to log in. Suppose we wanted a platform in which the sole task of the device was to execute a single program (e.g., a training server). What would need to happen for the device to go from boot straight to the program? Describe both the mechanism and some of the implications of the mechanism (positive or negative) in your solution. (~2 paragraphs)

[Q2] Compare and contrast booting a device with MacOS (Apple hardware and Apple software) versus Ubuntu or Linux (heterogeneous hardware and software platforms). (~3-5 sentences)

[Q3] Suppose a program is written to make a function call to `open`. Describe the interface that the program will take to implement this procedure. A complete answer describes what the operating system expects the user to pass to the kernel at the time the system call is made, what the user expects the operating system to return when the system call completes, and the mechanism that the program uses to invoke the operating system. (~3-5 sentences)

[Q4] We frequently described the operating system as a referee, illusionist, and glue in the context of a holistic system. In what ways is the system call interface an application of these principles? (~1-3 sentences)

Processes

[Q5] Think about the operating system that we implemented throughout the course. What is the relationship between processes, threads, and contexts? Why might an operating system like Linux use a different set of structures? (~1-3 sentences)

[Q6] Suppose, in an effort to implement fair scheduling, the timer interrupt was set to be the highest level priority interrupt. How would this help towards implementing fairness? What could go wrong? (~3-5 sentences)

File Systems

[Q7] Consider an operating system implementation for a system whose disk hardware is known to be faulty. In an effort to overcome potential failures, block allocation on this disk is implemented as allocating three-blocks as opposed to one. Whenever a disk block is read, all three blocks are read to confirm that they have the same state. If there is a fault in one of their state, then data read to the processor is the "mode" of their view of the state (the most common version of the block). If you were to design a file system for this scheme, what performance considerations might you make in terms of how the disk is laid out? (~3-5 sentences)

[Q8] Suppose you are running a virtual machine that has its own system. While running the virtual machine, you access a file that is private to that virtual machine. Describe the disk operations that would be performed relative to the file system on the host (i.e., the system hosting the VM) and the virtual machine itself. (~3-5 sentences)

[Q9] In an effort to save space in the super block, an operating system designer eliminates the free block list. In its place, they choose to allocate a single bit at the beginning of each disk block to indicate whether or not the block is free. Describe an instance in which such a scheme could achieve *better* performance than using a free list of data blocks. You may want to consider the storage device technology in your answer. (~3-5 sentences)

[Q10] In order to improve the performance of a directory lookup, an operating system may choose to organize the data block of a directory using a variety of potential data structures. Suppose you were to implement the directory as a hash map: keys in the hash map describe file names and values are the inode associated with the file name. Describe the pros and cons of using such a scheme relative to a traditional array. (~3-5 sentences)

Memory Management

[Q11] Recall that the operating system's goal in acting as a referee in a system is to allow all processes to fairly share the resources of the system. Consider the implications of this role on memory management. What would be a behavior that a process could attempt to do that would require refereeing and how would the operating system intervene? (~3-5 sentences)

[Q12] What is an advantage of performing slab allocation over a more complicated system like buddy allocation? Why might an operating systme want to use buddy allocation over slab allocation? (~3-5 sentences)

[Q13] Describe what is meant by a "huge page" in terms of address translation. What are some of the benefits achieved by using a huge page? (~3-5 sentences)

[Q14] Consider the following claim: "Every process has the OS kernel in its address space, so every process has an ability to refer to global structures of the OS, like the file system." If someone makes this (incorrect) claim to you, what might you say in response to correct them? (~3-5 sentences)

General Topics

[Q15] Consider a system that uses a non-volatile memory (NVM) instead of a disk. When the machine is off, all data is in memory but persists throughout the duration of the power-off time. Your goal is to re-imagine the boot procedure such that the OS kernel is able to execute once power is restored to the device. What would be some of the steps in the procedure that would change or stay the same? Speculate as to when this would be a desirable system. (~2 paragraphs)

[Q16] Consider an implementation of the OS kernel ABI in which function arguments are passed via the stack as opposed to through registers. The procedure that is called finds its arguments as variables on the stack (e.g., the first variable is immediately above the saved instruction pointer, the second variable is immediately above that, etc...). How would such an implementation of the ABI influence the context switching procedure? You may want to think about the fields of a context and the trade-offs relative to the current `x86_64` ABI in your solution. (~3-5 sentences)

[Q17] Consider the deployment of an operating system on an HPC server in a data center as compared to an operating system on a laptop. What are some of the workload characteristics in each of these environments that influence the operating system design space? Provide 3 examples and, for one of them, go into depth as to how the deployment case might influence a design decision that you would make. (~3-5 sentences)

[Q18] In order to implement performance engineering, modern operating systems support hardware event-tracking through a system called eBPF (the extended Berkeley Packet Filter). Whenever the event occurs in hardware, an interrupt is raised so that the operating system can execute a small kernel hook (developed by an end-user) that gets loaded *into the operating system*.

More generally, describe some of the challenges that could arise with running user code in the operating system. You may want to consider the larger roles of the OS (referee, illusionist, glue) in your response. (~3-5 sentences)