

Please write in your own words and be sure to show all your work and write down any assumptions you make.

1. Protection and security:

- (a) [5 points] What are the main differences between access lists and capability lists? Where does the Unix protection mechanism fit in the above schemes?
- (b) [5 points] Discuss ways by which managers of systems connected to the Internet could have limited or eliminated the damage done by the Morris Worm. What are the drawbacks of making such changes to the way in which the system operates?

2. Distributed file systems:

- (a) [5 points] NFS (Network File System) is said to provide a stateless file service while AFS (the Andrew File System) provides a stateful file service. What do these terms mean/what are some of the tradeoffs?
- (b) [5 points] We have discussed the Google, Andrew, and General Parallel File Systems in class. What are the environments for which each is tailored? Describe at least two file system design decisions made by each in order to support its target environment.

3. Virtual machines: A virtual machine monitor provides execution environments with an interface identical (or nearly identical) to the underlying bare hardware. These execution environments are called virtual machines (VMs). An operating system and its user programs run in each virtual machine. Conventionally, a process page table maps program *logical* addresses to *physical* addresses. However, the inside-VM physical address may not be equal to the real *machine* address in a VM-based system. For proper execution on a processor with hardware-loaded TLBs (such as x86), a page table that maps program logical addresses to real machine addresses must be presented to the hardware. In practice, the virtual machine monitor maintains one such *shadow* page table (logical to real machine address mapping) for each conventional inside-VM process page table (logical to VM physical address mapping).

- (a) [5 points] To maintain the shadow page table, does the virtual machine monitor need to trap updates to conventional process page tables (made by the inside-VM operating system)? Does the virtual machine monitor need to trap reads to conventional process page tables?
- (b) [5 points] Through operating system instrumentation (on Xen) or binary rewriting (on VMware), the inside-VM operating system can make explicit calls to notify the virtual machine monitor of page table updates (as opposed to using traps). What are the benefits of this approach?