Operating Systems cs5460/6460 **Name (Print):** _____
Spring 2014
Final
04/28/14
Time Limit: 8:00am - 10:00am

- **Don't forget to write your name on this exam.**
- **This is an open book, closed Internet exam.**
- **Ask me if questions are confusing.**
- **Organize your work**, in a reasonably neat and coherent way, in the space provided. Work scattered all over the page without a clear ordering will receive very little credit.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by explanation will receive no credit; an incorrect answer supported by substantially correct explanations might still receive partial credit.
- If you need more space, use the back of the pages; clearly indicate when you have done this.

Problem	Points	Score
1	25	
2	10	
3	15	
4	20	
5	20	
Total:	90	

1. File systems.

   (a) (10 points) You work on implementing a storage layer designed to serve photos for a social network web site like Facebook. You have the following setup. A web server like Apache runs as an xv6 process. Each time it receives a network request to serve a photo image, it opens a file, reads it, and sends it over the network. Performance of the storage layer—both file system and block layer—is critical. How many block reads are involved in each request (you don't have to be precise, but you need to justify your answer well).

   (b) (15 points) Suggest a design of a storage layer optimized for keeping a large number of small files, e.g. photos. Your main goal is to minimize the number of disk accesses involved in performing each request. Additionally make sure you reduce the number of disk seeks.

2. Linking and loading.

   (a) (5 points) Explain symbol relocation, and how it works. Specifically, 1) what gets relocated, 2) how the relocation information is encoded, 3) explain the actual relocation algorithm. Use diagrams.

   (b) (5 points) Explain position independent code. What purpose does it serve? How does it work? Use diagrams.

3. Memory management.

   (a) (5 points) Explain organization of the slab allocator.

   (b) (10 points) Why do you think xv6 does not have buddy or slab allocators? Under what conditions you would have to add these allocators to the xv6 kernel?

4. Virtual memory management.

   You plan to extend xv6 with support for demand paging, shared libraries, and copy-on-write fork. To implement these features, you need to design a virtual memory layer for xv6.

   (a) (10 points) What data structures are required to support shared libraries, e.g. sharing the code section of a library across multiple address spaces? Draw a diagram.

   (b) (10 points) Describe (draw a diagram) data structures required to support demand paging. You want to be able to detect idle (not frequently accessed) memory pages, swap them out to disk, and swap them in on a pagefault.

5. Virtualization. You decided to paravirtualize the xv6 kernel. The paravirtualized guest xv6 kernel runs as an xv6 process. Similar to a normal process, the guest kernel is not allowed to touch parts of the physical hardware that can subvert its isolation, e.g., the guest kernel is not allowed to update page, interrupt, and global descriptor tables in the CPU. Instead, every time the guest kernel needs to perform an operation on its hardware it invokes an explicit system call to the xv6 host kernel. This system call checks that update is safe, e.g., the guest page tables do not try to map parts of the host kernel, and then updates the state of the real hardware.

   (a) (10 points) How do you ensure isolation and protection across xv6 host kernel, xv6 guest kernel, and the processes inside xv6 guest kernel? Hint: use combination of paging and segmentation mechanisms.

   (b) (10 points) What is your approach to virtualizing page tables, e.g., what exactly is happening when the guest xv6 kernel tries to update its page tables?