CS 354 (Park) **Final** **May 6 (Tue.), 2014**

Remarks: Keep the answers compact, yet precise and to-the-point. Long-winded answers that do not address the key points are of limited value. Binary answers that give little indication of understanding are no good either. Time is not meant to be plentiful. Make sure not to get bogged down on a single problem.

PROBLEM 1 (40 pts)

(a) What is the key difference between XINU and UNIX file systems when it comes to locating the data blocks of a file, and what role does the "mice and elephants" property discussed in class play?

(b) Give an example of external fragmentation in memory management. What is the conceptual solution adopted for this problem and why is hardware support essential to make the solution viable?

(c) What are the main components of context-switch cost under process management (material covered before the midterm)? What additional costs are introduced under memory management (material covered after the midterm)? Which cost may be higher and why?

(d) What are the definitions of tickful and tickless kernel designs? How does the design difference show up in clock interrupt handler code (in XINU or any other kernel) when it comes to how a sleep queue is managed? What are the pros/cons of a tickless versus a tickful kernel? Which design may be more suited to mobile environments?

PROBLEM 2 (36 pts)

(a) What is a page fault? How are page faults handled in today's operating systems? With flash memories making significant inroads as persistent memory and the speed-gap with RAM shrinking, what changes to page fault handling may be feasible in future kernels and why?

(b) In lab3, you implemented a version of Solaris that uses a multi-level feedback queue to achieve TS scheduling. What are the enqueuing and dequeuing costs of a process in XINU, and why is the cost considered constant (i.e., $O(1)$) in the Solaris implementation?

(c) Suppose your laptop or PC is connected to a USB 2.0 web cam, the hardware includes DMA support, and the operating system is a modern kernel (e.g., Linux/Windows). What happens when small packetized video pieces from the web cam arrive at the USB interface every 125 μsec? What does the DMA do? What is the role of the top (or fast) half of the lower half? What does the bottom (or slower) half do? What is the responsibility of the upper half of the kernel? How does a kernel contribute to degradation of video quality?

PROBLEM 3 (24 pts)

(a) Suppose a kernel (e.g., XINU) uses a delta list to manage sleep events requested by processes using sleep system calls. Suppose at time 0 the delta list empty. At time 1 sec, process A calls sleep with argument 12 sec. At time 4 sec, process B calls sleep with argument 8 sec. At time 6 sec, process C calls sleep with argument 8 sec. Assuming the kernel follows a tickful design and its tick value is 1 sec, show the contents of the delta list at time instances 1 sec, 2 sec, 4 sec, and 6 sec.

(b) What is a TLB miss? In computing system where TLB misses are handled by an operating system, what are the steps undertaken by a kernel to resolve a TLB miss? Assuming a disk is used for persistent storage, when is disk I/O needed to handle a TLB miss? What is a common method used by kernels to avoid such disk I/O? Why are 2-level page tables relevant for avoiding disk I/O for TLB misses? What happens after a TLB miss has been resolved?

BONUS PROBLEM (10 pts)

Our discussions showed that locality of reference is a crucial property of real-world programs that allows computing systems to be economically designed and implemented. Give two examples where kernels actively exploit this property to improve the performance of computing systems.