CS 354 (Park) **Final** **Dec. 12 (Tue.), 2023**

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) Explain how indexing in traditional file systems achieves constant overhead for small files and logarithmic overhead for large files. What technique is employed in some file systems to help achieve constant indexing overhead even for large files? FAT and XINU file systems incur linear indexing overhead. Why are FAT file systems still used in real-world environments? Suppose a FAT file system is implemented on SSD. What is the role of the flash translation layer, and why is it essential?

(b) What are the pros/cons of using interrupt disabling to achieve mutual exclusion? Counting semaphores implemented through XINU's wait() and signal() system calls are considered software solutions for supporting process coordination and synchronization. Why is "software solution" an imprecise characterization? A producer/consumer buffer may be protected using a single binary semaphore or a pair of counting semaphores to prevent data corruption. Describe how the pair of counting semaphores are used by writer and reader to share a FIFO buffer. What is a scenario where using a pair of counting semaphores is superior to using a single binary semaphore? Explain your reasoning.

PROBLEM 2 (40 pts)

(a) Describe what internal and external fragmentation are. In a 32-bit CPU with a single-level page table and 4 KB page size, how many page table entries are there? What information is contained in each page table entry to handle external fragmentation? What is demand paging? What information is contained in each page table entry to support demand paging? What is a major page fault? Every process possesses its own page table. Explain how per-process page tables are related isolation/protection. What are the main drawbacks of per-process page tables?

(b) Suppose a FIFO buffer is shared between upper and lower halves of a kernel to support device I/O. In the case of a read operation (e.g., read() system call to read from a device in Linux), describe who the writer of the shared kernel buffer is, and who the reader is, under the assumption of context-borrowing. Be precise when specifying "who" the writer and reader are. What happens if the buffer is full when the lower half attempts to write? What would be a common reason for the kernel buffer to become full?

PROBLEM 3 (20 pts)

Suppose a page fault occurs and a kernel needs to evict a page to make room because all frames are occupied. If the kernel could see into the future, what would be the optimal method for selecting a page to be evicted? In what sense is it optimal? Discuss why LRU is able to approximate the optimal method. Given the costs associated with implementing LRU in hardware support, describe how global clock works. In what way can global clock be considered an approximation of LRU? What part of a kernel carries out global clock?

BONUS PROBLEM (10 pts)

What is memory thrashing? What are its main symptoms? Would increasing CPU speed help mitigate the problem? Explain your reasoning.