Exam 2

601.418/618 Operating Systems

April 8, 2024

Complete all questions.

Time: 75 minutes.

I affirm that I have completed this exam without unauthorized assistance from any person, materials, or device.

Signed: ____

Print name: ____

Date: ____

Question 1. [5 points] Briefly explain the purpose of the TLB (Translation Lookaside Buffer.) Be sure to indicate what information the TLB stores and how that information is accessed.

Question 2. [5 points] On some architectures, TLBs support "address space identifiers" (ASIDs), where TLB entries are marked with an identifier indicating which address space the entry belongs to.

(a) What is the advantage of supporting ASIDs? Explain briefly.

(b) Are there any drawbacks or difficulties related to ASIDs? (For example, suggest some reasons why an architecture might not choose to implement them.)

Question 3. [15 points] Assume that a system has 4 pages of physical memory available for user processes. Assume that a single user process is running, and that initially none of its virtual pages are mapped to a physical page.

The process accesses its virtual pages according to the following reference string (each number represents a virtual page number):

8 3 5 7 3 5 8 6 7 3 7 8 1 10 8

(a) How many page faults occur if the page replacement policy is LRU? Draw a diagram that indicates, for each physical page, which virtual page is mapped to that physical page at each step in the reference string. (This will help us evaluate your answer for partial credit.)

(b) How many page faults occur if the page replacement policy is FIFO? (As with (a), draw a diagram showing how virtual pages are assigned to physical pages at each step.)

Question 4. [10 points] Recall that the MMU typically maintains some information about references to virtual pages in their page table entries. For example, 32-bit x86 CPUs store a 1 in the "A" ("accessed") bit of a virtual page's page table entry the first time it is is accessed. An accessed bit can be used by the OS kernel to *approximate* LRU page replacement (e.g., using the clock algorithm.)

(a) What information would the MMU need to record in order to implement "true" LRU eviction of pages?

(b) Is true LRU eviction practical? Discuss briefly.

Question 5. [10 points] Briefly describe how the clock algorithm is used to perform page replacement. Be sure to indicate when (under what circumstances) the clock algorithm executes, and what it does when it executes.

Question 6. [5 points] Briefly explain *demand paging.* State some advantages and disadvantages of demand paging.

Question 7. [10 points] In Assignment 3a you are implementing support for disk paging in Pintos. One of the key data structures is the *frame table*. Each entry in the frame table represents one page of physical memory, and stores information about which address space it is being used by (if any.)

(a) Describe a design for the frame table data structure that would allow for *shared memory*, i.e., pages of physical memory which are mapped by multiple user address spaces. Drawing a diagram could be helpful.

(b) Describe the steps executed by the OS kernel when it steals a page, i.e., unmaps a physical page from an address space in order to use the page elsewhere. Be explicit about what information the OS kernel updates and how it accesses that information.

Question 8. [20 points] Assume that a hard disk

- has a single platter with a single surface,
- has 8 tracks on that surface (numbered 0–7),
- has 10 sectors in each track (numbered 0–9),
- spins at 5400 RPM (rotations per minute)

Assume that 0 is the outermost track and 7 is the innermost track.

Assume that a sector's "logical address" is $10t + s$, where t is the track number and s is the sector number within its track. For example, logical addresses 0–9 are the sectors of track 0, logical addresses 10–19 are the sectors of track 1, etc.

Assume that 1 ms is required to move the disk head to an adjacent track. So, moving from track 2 to track 5 would require $(5 - 2) \times 1 = 3$ ms.

For parts (b) through (f), assume that the a burst of I/O requests for the sectors with logical addresses

7 21 5 59 61

arrive, in that order. Assume the head is initially positioned at track 4.

(a) What is the average rotational delay of this hard disk? (Note that 5400 RPM is $11.\bar{1}$ milliseconds per rotation.)

(b) What is the total seek time if the I/O requests are scheduled in First-Come First-Served (FCFS) order?

I/O requests: 7 21 5 59 61

Disk head is initially positioned at track 4.

(c) In what order are the requests handled if they are scheduled in Shortest Seek Time First (SSTF) order?

(d) What is the total seek time if the I/O request are processed in Shortest Seek Time First (SSTF) order?

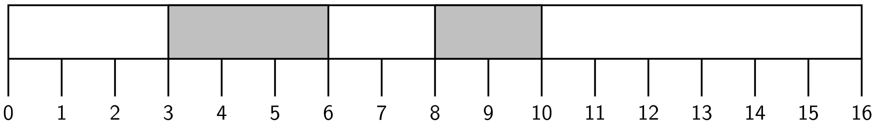
I/O requests: 7 21 5 59 61

Disk head is initially positioned at track 4.

(e) In what order are the requests handled if they are scheduled using the C-SCAN algorithm? (Recall that C-SCAN is like SCAN (the "elevator" algorithm), but sweeps occur in only one direction. Assume the sweep moves the head from lower-numbered tracks towards higher-numbered tracks.)

(f) What is the total seek time if the I/O request are processed using the C-SCAN algorithm?

Question 9. [10 points] Assume that a memory allocator has 16 units of memory. Assume the current state of the heap looks like this (shaded blocks are allocated, white blocks are available space):



(a) Show the state of the heap after allocations of sizes $\boxed{2, 4, 3}$ (in that order) using First-Fit allocation. (Assume scan for available space starts at offset 0.) If any allocations fail, mention which ones.

(b) Show the state of the heap after allocations of sizes $\boxed{2, 4, 3}$ (in that order) using Best-Fit allocation. If any allocations fail, mention which ones.

(c) Briefly discuss some advantages and disadvantages of using Worst-Fit allocation.

Question 10. [10 points] Assume a heap with 64 units of memory is managed using a buddy allocator. Assume that initially, the heap is completely empty.

(a) Given the starting conditions described above, show the state of the heap after allocations of sizes 8 and 4. Show the blocks that exist, the size of each block. Make sure it's clear which blocks are allocated and which blocks are available. (Reminder: in a buddy allocator, allocated and available blocks always have a size that is a power of 2.)

(b) In general, what is the worst-case running time of a single memory allocation using a buddy allocator? Assume that the largest allocation size supported is 2^n, and the smallest allocation size supported is 2^m, such that n and m are integers, and $n > m$. Explain briefly.

Bonus question. [3 points] Intel's documentation for x86 CPUs recommends that if an operating system modifies the A ("Accessed") bit in a page table entry, it should also use the `invlpg` ("invalidate page") instruction to invalidate the corresponding TLB entry.

Explain why invalidating the TLB entry is necessary.

Bonus question. [3 points] Let's say that an architecture does not support any hardware mechanism to record that a virtual page has been accessed. I.e., there is no hardware-maintained information in a page table entry similar to the x86 "accessed" bit.

Could an OS kernel still implement the clock algorithm to approximate LRU page replacement? Explain.

[Extra page for answers and/or scratch work.]

[Extra page for answers and/or scratch work.]