Exam 2

601.418/618 Operating Systems

April 9, 2025

Complete all questions.

Time: 75 minutes.

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

Signed: _____

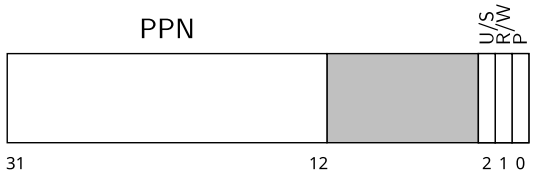
Print name: _____

Date: _____

Question 1. [10 points] Recall that virtual memory on the 32-bit x86 architecture has the following characteristics:

- Pages are 2^{12} bytes in size
- Two levels of page tables
- Each page table has 1,024 (2^{10}) page table entries
- Each PTE is 32 bits (4 bytes)
- P=1 means page is present (valid)
- R/W=0 means page is writable
- U/S=0 means page is a user page

Page table entry format (not all bits are labeled):



Show code to implement the `map_vp_to_pp` function below. It should map the virtual page whose virtual address is given as the `va` parameter to the physical page whose physical address is given as the `pa` parameter. The mapping should be writable and user-accessible (not kernel-only.) The `pagedir` parameter is the kernel virtual address of the page directory (root page table.) Assume that the function call `palloc_get_page(PAL_ZERO)` allocates a zero-filled memory page and return its kernel virtual address, or returns `NULL` if the allocation fails. Assume that the `vtop` and `ptov` functions convert between kernel virtual addresses and physical addresses. (I.e., page allocation and kernel virtual memory work the same way as in PintOS.) The function should return `true` if successful or `false` if not.

```
bool map_vp_to_pp( uint32_t *pagedir, uint32_t va, uint32_t pa ) {
```

[Continue on next page if necessary.]

[Continue your answer to Question 1 on this page if necessary.]

Question 2. [10 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):

5 4 2 3 1 4 1 7 2 4 3 6 4 10 9

(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 3. [10 points] 32-bit x86 CPUs starting with the Intel Pentium support the *Page Size Extension* feature. When bit 7 of a page directory entry ("PDE", a PTE in the root page table) is set to 1, then rather than referring to the physical address of a level 2 page table, the entry directly refers to a "page" of physical memory of size 4 MB (2^{22} bytes), with the 10 most significant bits of the PDE indicating the physical address of the 4MB "page" being mapped.

(a) Explain why in some cases mapping a 4MB page rather than 1,024 4KB pages would be useful. Justify your answer.

(b) State two difficulties that the OS kernel would face in utilizing a 4MB page as part of a user address space.

Question 4. [5 points] Explain how the "Accessed" bit in the PTE is used by the clock algorithm.

Question 5. [5 points] What is the advantage of the two-handed clock algorithm over the basic (single-handed) clock algorithm? Describe a scenario where the two-handed version would be more effective.

Question 6. [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 7200 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

9 74 16 30 65

arrive, in that order. Assume the head is initially positioned at track 6.

(a) What is the average rotational delay of this hard disk? (Note that 7200 RPM is $8\frac{1}{3}$ milliseconds per rotation.)

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

I/O requests: 9 74 16 30 65

Disk head is initially positioned at track 6.

(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? (You should ignore rotational delay.)

I/O requests: 9 74 16 30 65

Disk head is initially positioned at track 6.

(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? (You should ignore rotational delay.)

Question 7. [10 points] Recall that in Unix, metadata about a file or directory such as its owner, permissions, and access time is stored in the inode rather than the directory entry.

(a) How would access to files and directories work differently if file ownership and permissions were stored in the directory entry? Your answer should address how this change would affect users and applications.

(b) State one possible advantage of the change mentioned in (a). Justify your answer briefly.

(c) State one possible disadvantage of the change mentioned in (a). Justify your answer briefly.

Question 8. [10 points] Assume that

- Disk storage blocks are 2K (2^{11} bytes) in size, and consist of 4 consecutive 512-byte sectors
- A disk sector address is 64 bits (8 bytes) in size
- An inode has 4 direct blocks, 2 singly-indirect blocks, and 1 doubly-indirect block

(a) How many disk sector addresses fit in one disk storage block?

(b) What is the maximum amount of storage (in bytes) that can be allocated for a file or directory using this scheme? A precise formula showing how to calculate the maximum storage is a sufficient answer. Show your work.

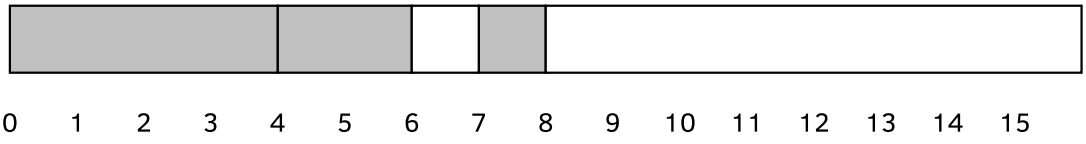
Question 9. [10 points] Consider the following heap, where shaded blocks are allocated and white blocks are available (each block is labeled with its size):

In the parts (a) and (b), assume that first-fit considers available blocks from left to right.

(a) Show a sequence of allocation requests which, if executed in order, will all succeed using a best-fit allocator but for which at least one will fail using a first-fit allocator. Explain briefly.

(b) Show a sequence of allocation requests which, if executed in order, will all succeed using a first-fit allocator but for which at least one will fail using a best-fit allocator. Explain briefly.

Question 10. [5 points] Consider the following heap with 16 units of storage, managed using a buddy allocator, where shaded blocks are allocated and white blocks are available:



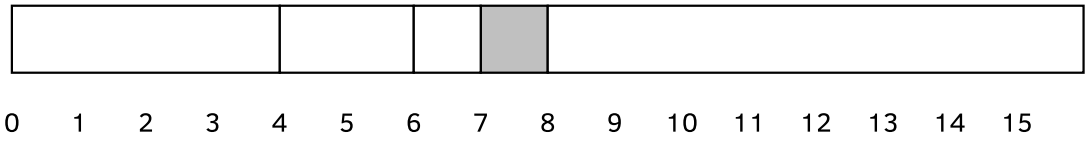
(a) Show the freelists for each possible block size. Use the heap addresses (0–15) to represent the identities of free blocks.

(b) Starting with the initial heap shown above, describe the sequence of steps to allocate a buffer of size 2, including

- the address of the allocated block, and
- which block(s) are split (if any)

Also, show the freelists and sketch the state of the heap as they would be after the allocation.

Question 11. [5 points] Consider the following heap with 16 units of storage, managed using a buddy allocator, where shaded blocks are allocated and white blocks are available:



(a) Describe the sequence of steps to would occur when the allocated block at address 7 is freed.

(b) Show the freelists and sketch the state of the heap after the block at address 7 is freed.

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

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