Final of Introduction to Operating Systems, 2023 Fall Semester, NYCU

Total: 104 pts.

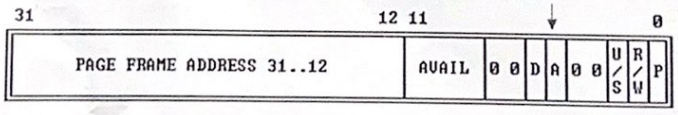
***Provide sufficient explanation in your answer to get full credits.*

***You are solely responsible for your answer. TA's explanation is for your reference only.*

1. [Ch7, 6pts] Consider a set of processes that have been deadlocked. Which one(s) of the following can possibly resolve the deadlock? Why?
   a. Increasing the CPU clock rate
   b. Adding more free memory
   c. Killing a process in the deadlock
2. [Ch7, 4pts] In Banker's Algorithm, why is a safe sequence tested with the largest resource requests of processes?
3. [Ch8, 8pts] Best-Fit, First-Fit, and the buddy system are algorithms for dynamic memory allocation. Answer the following questions:
   a. There are two types of memory fragmentation, internal fragmentation and external fragmentation. Which one(s) that the buddy system is subject to? Use an example to explain your answer.
   b. Show an initial memory layout and a sequence of memory allocations/deallocations for which First-Fit fails but Best-Fit succeeds (Hint: it's about how free holes are created)
4. [Ch8, 8pts] Consider a 32-bit microprocessor which employs two-level page tables. The first-level page table has 1024 entries, each of which referring to a second-level page table. A second-level page table has 1024 entries, each of which referring to a 4KB page. Answer the following questions:
   a. An entry of the first-level page table contains an address to a second-level page table. Is the address a physical address or a virtual one? Why?
   b. Extend this two-level page table design to support both 4KB pages and 4MB pages at the same time. Use an example to explain your design, and also explain how address bits are used during address translation.
5. [Ch8, 6pts] In typical cases, code segments cannot be modified, and stack segments cannot be executed. Explain why.
6. [Ch9, 5pts] Consider the following sub-operations of demand paging. Which one(s) are conducted by hardware (MMU), and which one(s) are carried out by software (OS)? Why?
   a. Looking up the TLB
   b. Looking up the page table
   c. Updating a page table entry
   d. Allocating the memory space for a page table
   e. Finding a victim page for page replacement
7. [Ch9, 5pts] The following figure shows the format of a page table entry (PTE) of Intel 386+ processors. Explain the purpose of the following bits:

Figure 5-10. Format of a Page Table Entry

A: referenced

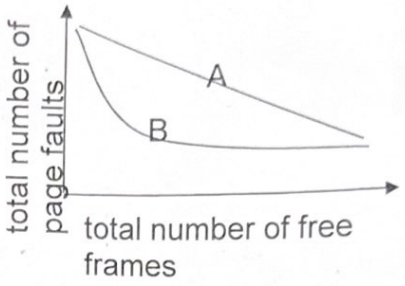


P - PRESENT
R/W - READ/WRITE
U/S - USER/SUPERVISOR
D - DIRTY
AVAIL - AVAILABLE FOR SYSTEMS PROGRAMMER USE

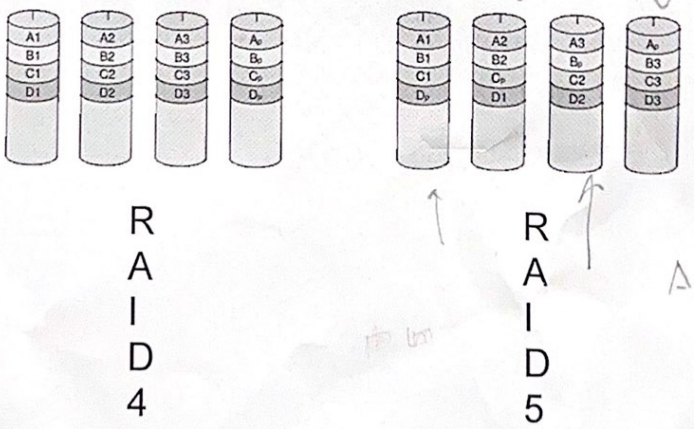
NOTE: 0 INDICATES INTEL RESERVED. DO NOT DEFINE.

   a. D: Dirty
   b. A: Accessed, as known as referenced
   c. U/S: User/Supervisor
   d. R/W: Read/Write
   e. P: Present, as known as valid

8. [Ch9, 6pts] Least-Recently Used (LRU) and Least-Frequently Used (LFU) are two typical page replacement algorithms. Which one better suits the following workloads? Why?
   a. Intensive reference to a small page set interleaved by sequential scan of a large page set
   b. Migrating among a few sets of popular pages.
9. [Ch9, 6pts] See the right-hand side figure, which depicts the total number of page faults versus the total number of free frames. Which curves of A and B best describes the performance trend of a typical program? Why?



10. [Ch10, 5pts] Suppose that you are creating a file link across different file systems. Between soft link and hard link, which one can do this? Why?
11. [Ch10, 5pts] What is the dangling pointer problem of file links? How do UNIX-based file systems resolve this problem?
12. [Ch 11, 5pts] Consider the UNIX inode scheme. Let an inode contain 10 direct pointers, 1 indirect pointer, 1 double indirect pointer, and 1 triple indirect pointer. Let the block size (for both index blocks and data blocks) be 4KB, and let a pointer be of 4 bytes. Calculate the largest file size. (it's okay to give a formula)
13. [Ch 11, 6pts] Explain why the following designs improve file system performance:
   a. Embedding a few direct pointers to inodes
   b. Embedding tiny files to directories
   c. Allocating file space using extents
14. [Ch 11, 6pts] Answer the following questions regarding journaling file systems:
   a. What is a transaction?
   b. How does write-ahead logging (WAL) work?
   c. How does WAL guarantee the atomicity of transactions?
15. [Ch 11, 5pts] We have talked a lot about conventional UNIX file systems, e.g., the Ext4 file system, and log-structured file systems. Now consider an application scenario that involves intensive random updates to small files. In this scenario, the file system space utilization (fullness) is not higher than 50%. Between the two types of file system, which one better suits this application scenario? Why?
16. [Ch12, 6pts] SSTF (Shortest Seek Time First) is a good disk scheduling algorithm in terms of seek time reduction. However, it is rarely used in real systems, at least not in its original form. What is the major problem of SSTF? Propose a workaround for this problem.
17. [Ch12, 6pts] The figure shows the organizations of RAID4 and RAID5. Their difference is that parity blocks (Ap, Bp, Cp, and Dp) of RAID5 are distributed over disks. Explain the rationale of this design.



18. [Ch12, 6pts] Although (flash-based) SSDs do not involve mechanical parts and thus are free from the latencies of seek and rotation, writes necessitate SSD internal management activities, which may degrade the I/O performance. Answer the following:
   a. What is garbage collection? How does it work?
   b. What is wear leveling? Why is it essential to extend SSD lifetime?