EECE.4810/EECE.5730: Operating Systems

Spring 2018

Extra Credit Homework

Due **11:59 PM, Friday, 5/4/18**

Notes:

- All submissions must be e-mailed to Dr. Geiger at Michael_Geiger@uml.edu.
- Any electronic submission must be in a single file. Archive files will not be accepted.
- This assignment is worth up to **7.5 extra points on your final average—half of what each exam is worth.**

1. (24 points) ***Page table organization***

Given a system using 8 KB pages, a 48-bit virtual address, and 8 GB of physical memory, determine the amount of space required to store each of the following types of page tables. Assume each page table entry (PTE) requires 4 bytes unless the problem states otherwise.

a. A basic, one-level page table that covers the entire virtual address space.

b. A two-level page table, assuming the currently running process is using the entire first half of its virtual address space.

- ***NOTE:*** Typically, in multi-level paging, the upper address bits are split evenly to index into each level of the page table. For example, given a 32-bit address with a 10-bit page offset, in a two-level page table, the remaining 22 address bits are split so 11 bits are used for the first-level table and 11 bits are used for the second-level tables.

  However, if the number of remaining bits are odd, assume the extra bit is used to index the first-level table. So, in the example above, if the page offset were 11 bits, not 10, the remaining 21 address bits would be split such that 11 bits were used for the first-level table and 10 bits were used for the second-level tables.

c. A hash table in which:

i. Each entry in the hash table contains a single 32-bit pointer to a chain of PTEs

ii. Each PTE consists of 8 bytes—4 bytes for the typical PTE content plus a 4 byte pointer to the next PTE in the chain.

iii. The number of entries in the hash table (not necessarily the number of entries in each chain) is 1/16 of the number of entries in the one-level page table

d. An inverted page table, with one PTE for each physical frame.

2. (12 points) ***Page replacement***

Say the currently running process has 16 active pages, P0-P15. P0 and P8 both have their reference bits set to 0, while all other pages have their reference bits set to 1. If the operating system uses the clock algorithm for page replacement, the pages are ordered numerically around the "clock" (P0 is first, P1 is second, etc.), and the "clock hand" currently points to P4, what are the first four pages to be replaced, assuming none of the currently active pages are referenced before four replacements are required? **Explain your answer for full credit.**

3. (15 points) ***Virtual memory***

A portion of the currently running process's page table is shown below:

Virtual page #	Valid bit	Reference bit	Dirty bit	Frame #
7	1	1	1	3
8	0	0	0	--
9	1	0	1	4
10	0	0	0	--
11	1	1	0	0
12	1	1	0	1

Assume the system uses 20-bit addresses and 16 KB pages. The process accesses four addresses: 0x25FEE, 0x2B149, 0x30ABC, and 0x3170F.

Determine (i) which address would cause a page to be evicted if there were no free physical frames, (ii) which one would mark a previously clean page as modified, if the access were a write, and (iii) which one accesses a page that has not been referenced for many cycles. **For full credit, show all work.**

4. (24 points) ***File systems: organization***

This problem asks you to assess file access time for each of the three file systems (FAT, FFS, NTFS) discussed in class. Assume, in all cases:

- Each disk block is 8 KB
- The time required to access a new disk block is 20 ms
- Once a disk block is accessed for the first time, it remains cached in main memory, and every access to the cached block after the first one will take 2 ms.
- ***NOTE (ADDED 5/1):*** The address size is 32 bits.

Determine the worst case access time for a single read operation to an undetermined word of data in a 16 MB file if the file system used is (a) FAT, (b) FFS, (c) NTFS.

- For NTFS, assume all records in the MFT fit in a single disk block, and the file in question requires 4 MFT records to store all necessary metadata.

5. (10 points) ***File systems: reliability***

a. (5 points) Explain why methods for ensuring reliability in file systems center on operations that write a single sector.

b. (5 points) A Linux variant called TxOS, developed at the University of Texas, supports transactions with shadowing by decomposing inodes into two parts: a header that contains infrequently modified data about each file, and a data component holding fields that are commonly modified by system calls. The header contains a pointer to the related data component, and the data component contains a pointer to the header.

Explain how this inode organization makes it relatively easy to implement shadowing for changes to a file's metadata.

6. (9 points) ***Protection***

Given the access matrix below, answer questions (a)-(c).

Domains	**Objects**						
	F_1	F_2	F_3	D_1	D_2	D_3	D_4
D_1	read* write*		write		control		
D_2		owner					
D_3	read owner		read*				
D_4		write	write			control	

a. Can a process in domain D_1 grant read privileges for object F_3 to a process in domain D_2? Why or why not?

b. Can a process in domain D_3 revoke the write privileges of a process in domain D_1 for object F_3? Why or why not?

c. Can a process in domain D_2 give read, write, and execute privileges for object F_2 to a process in domain D_3? Why or why not?

7. (6 points) ***Security***

Two professors are discussing the security of a distributed system in which the key used to encrypt messages is known. Professor A claims that knowing the encryption key will allow him to decrypt any message sent between nodes in the system once he determines the type of cryptography used, thus making the system insecure. Professor B claims it is possible for the system to be secure, depending on what method is used to encrypt messages. Which professor is right, and why?