Spring 2019 COMP 3511 Homework Assignment #4
Handout Date: April 24, 2019 Due Date: May 8, 2019

Name: _Huang Daoji_ ID: _20623420_E-Mail: _dhuangal@connect.ust.hk_

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are total of **4** questions.
- When you write your answers, please try to be precise and concise.
- Fill in your name, student ID, email and Section number at the top of each page.
- Please fill in your answers in the space provided.

- **Homework Collection:** the homework is submitted to **assignment #4** on **CASS**

1. [20 points] Multiple choices

   1) Which of the following is true for compaction?
      A) It can be done at assembly, load, or execution time.
      B) It is used to solve the problem of internal fragmentation.
      C) It cannot shuffle memory contents.
      D) It is possible only if relocation is dynamic and done at execution time.
      Answer: _D_

   2) Assume a system has a TLB hit ratio of 90%. It requires 15 nanoseconds to access the TLB, and 85 nanoseconds to access main memory. What is the effective memory access time in nanoseconds for this system?
      A) 108.5
      B) 100
      C) 22
      D) 176.5
      Answer: _A_

   3) Consider a logical address with 18 bits used to represent an entry in a page table. How many entries are in the conventional page table?
      A) 262144
      B) 1024
      C) 1048576
      D) 18
      Answer: _A_

   4) Given the logical address 0xAEF9 (in hexadecimal) with a page size of 256 bytes, what is the page number?
      A) 0xAE
      B) 0xF9
      C) 0xA
      D) 0x00F9
      Answer: _A_

5) Which of the following is a benefit of allowing a program that is only partially in memory to execute?

A) Programs can be written to use more memory than is available in physical memory.

B) CPU utilization and throughput is increased.

C) Less I/O is needed to load or swap each user program into memory.

D) All of the above

Answer: _D_

6) Given the reference string of page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and a system with three frames, what is the final configuration of the three frames after the LRU algorithm is applied?

A) 1, 3, 4

B) 3, 1, 4

C) 4, 1, 2

D) 1, 2, 3

Answer: _B_

7) _____ allows the parent and child processes to initially share the same pages, but when either process modifies a page, a copy of the shared page is created.

A) copy-on-write

B) zero-fill-on-demand

C) memory-mapped

D) virtual memory fork

Answer: _A_

8) Consider a disk queue holding requests to the following cylinders in the listed order: 116, 22, 3, 11, 75, 185, 100, 87. Using the SSTF scheduling algorithm, what is the order that the requests are serviced, assuming the disk head is at cylinder 88 and moving upward through the cylinders?

A) 116 - 22 - 3 - 11 - 75 - 185 - 100 - 87

B) 100 - 116 - 185 - 87 - 75 - 22 - 11 - 3

C) 87 - 75 - 100 - 116 - 185 - 22 - 11 - 3

D) 100 - 116 - 185 - 3 - 11 - 22 - 75 - 87

Answer: _C_

9) Which of the following disk head scheduling algorithms does not take into account the current position of the disk head?

A) FCFS

B) SSTF

C) SCAN

D) LOOK

Answer: _A_

10) A volume control block ____.

A) can contain information needed by the system to boot an operating system from that partition

B) is a directory structure used to organize the files

C) contains many of the file's details, including file permissions, ownership, size, and location of the data blocks

D) contains information such as the number of blocks in a partition, size of the blocks, and free-block and FCB count and pointers

Answer: _D_

2. [20 points] Memory management

1) Consider a logical address space of 32 pages of 1KB each, mapped onto a physical memory of 128 frames. How many bits are there in the logical address and in the physical address, respectively? (4 points)

Answer:

- **The logical address has 5 + 10 = 15 bits**
- **The physical address has 7 + 10 = 17 bits.**

2) Suppose an OS has a 24-bit virtual address and a 20-bit physical address, and a 4KB page size. How many entries are there in the page table? What is the size of the physical memory? (4 points)

Answer:

- **There are 24 – 12 = 12 bits for page number, thus 2^12 = 4096 entries in the page table**
- **The size of physical memory is 2^20 = 1MB**

3) Page translation. In a 32-bit machine we subdivide the virtual address into 3 segments as follows:

page number		page offset
8-bit	12-bit	12-bit

We use a two-level page table (in memory) such that the first 8 bits of an address is an index into the first level page table and the next 12 bits are an index into a second level page table. Each page table entry is 32 bits in size. (12 points)

a) What is the page size in such a system? (2 points)

Answer:

- **2^12 = 4KB**

b) How many entries are in the 1st level page table? How many entries are in one 2nd level page table? (2 points)

Answer:

- **For the first level page table, there are 2^8 = 256 entries**.
- **For one second level page table, there are 2^12 = 4096 entries**.

c) How much space is occupied in memory by the page tables for a process that has 256MB of actual virtual address space allocated? Show your work with detailed explanation. (8 points)

Answer:

- **One second level page table is responsible for 4096 pages, which is 4k * 4k = 16M memory**
- **A process that has 256MB of actual virtual address needs one first level pagetable and 256 / 16 = 16 second level pagetables(at least).**
- **One first level pagetable is of 256 * 4 = 1k large, and one second level pagetable is of 4096 * 4 = 16k large**
- **So this process needs (at least) 1k + 16k * 16 = 257k memory for its pagetable**

3. [30 points] Virtual Memory

1) The following page table illustrates a system with 12-bit virtual and physical addresses and 256-byte pages. Free page frames are to be allocated in the order 9, F, D. A dash for a page frame indicates that the page is not in memory. (4 points)

Page	Page-Frame
0	0x4
1	0xB
2	0xA

3	-
4	-
5	0x2
6	-
7	0x0
8	0xC
9	0x1

Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. In the case of a page fault, you must use one of the free frames to update the page table and resolve the logical address to its corresponding physical address. (a) 0x2A1, (b) 0x4E6, (c) 0x94A, and (d) 0x316

Answer:

- **a**: 0xAA1
- **b**: 0x9E6
- **c**: 0x14A
- **d**: 0xF16

2) Suppose a replacement policy works as follows: it examines each page periodically and discards a page if it has not been used since the previous examination. How can you implement this algorithm? What is the advantage and disadvantage of such a replacement policy compared with LRU or second-chance replacement? (6 points) [Hint: use a reference bit]

Answer:

- **Use a reference bit associated to each page**, set the reference bit to 1 if it is accessed and periodically discards pages with reference bit being zero, and set all remained pages' reference bits to zero after examination.
- **Advantage: it saves more space as each page which has not been assessed in one time quantum is discarded**
- **Disadvantage: increased IO overhead because more pages are needed to write back or bring in; more page fault rate because more pages may be discarded before accessed again.**

3) Suppose demand-paging is used. Memory-access time is 100 nanoseconds. It takes 8 milliseconds to service a page fault if an empty frame is available and if the replaced page is not modified, or it takes 20 milliseconds if the replaced page is modified. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds? (5 points)

Answer:

- **From the definition of EAT, we have**: 100(1-p) + (8000000*0.3+20000000*0.7)p < 200
- **So p = 1/163999**

4) Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each. (1) LRU replacement (2) FIFO replacement (3) Optimal replacement. (15 points)

Answer:

Number of frames	LRU	FIFO	Optimal
1	20	20	20
2	18	18	15
3	15	16	11
4	10	14	8
5	8	10	7
6	7	10	7
7	7	7	7

4. [30 points] Secondary storage and file systems

1) Please name three advantages that NVM such as SSDs has over HDDs. (3 points)

Answer:
- **More reliable for no moving parts**
- **Less power consumption**

- **Much faster to access**

2) Can a RAID level 1 organization achieve better performance for read requests than a RAID level 0 organization (with nonredundant striping of data)? If so, how? (4 points)

Answer:
- **Yes, when serving a read request, a RAID level 1 organization can choose one disk which disk head is more closer to the requested data to perform reading, thus it could be faster than RAID 0.**

3) Disk scheduling. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is:

86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

Starting from the current head position (143), what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms? (12 points)

a) FIFO
b) SSTF
c) SCAN
d) LOOK
e) C-SCAN
f) C-LOOK

Answer:
- A: 7081
- B: 1745
- C: 9769
- D: 3319
- E: 9813
- F: 3363

4) The open-file table is used to maintain information about files that are currently open. The operating system usually maintains one table (system-wide file open table) that contains references to files that are currently being accessed by all users, instead of a separate table for each user. Please explain why. [Hint: consider two processes access the same file] (5 points)

Answer:
- **In this way, it is more easy for more than one processes to share one open file, just let each process's open**-file table point to system-wide fil open table.
- **It saves more space, for otherwise more spaces will be wasted for the same open**

file

- **It is more convenient for parent/child process to share the same file, and decrease the overhead in copying the open-file table for each process.**

5) Consider a file system that uses *inodes* to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system? (6 points)

Answer:

- 12 * 8 + 8 * 8 * 1024 / 4 + 8 * (8 * 1024 / 4)^2 + 8 * (8 * 1024 / 4)^3 = 68 753 047 648 KB = 64 TB