Fall 2017 COMP 3511 Homework Assignment #4 Solution

Handout Date : Thursday (16 Nov.) Due Date: Tuesday (30 Nov.)

Name: ______________ ID: ___________________E-Mail: _____________

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are a total of **3** 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, or you can type your answers in the MS Word file.

- **Homework Collection: the hardcopy** is required and the homework is collected in collection box #20. The collection boxes locate outside Room 4210, near lift 21 (there are labels attached on the boxes).

1. (20 points) Multiple choices

   1) If the size of logical address space is 2 to the power of m, and a page size is 2 to the power of n addressing units, then the high order _____ bits of a logical address designate the page number, and the ____ low order bits designate the page offset.
   a) m, n
   b) n, m
   c) m – n, m
   d) m – n, n

   Answer: d

   2) For larger page tables, they are kept in main memory and a __________ points to the page table.
   a) page table base register
   b) page table base pointer
   c) page table register pointer
   d) page table base

   Answer: a

   3) Time taken in memory access through PTBR is :
   a) accelerated by a factor of 3
   b) accelerated by a factor of 2
   c) slowed by a factor of 3
   d) slowed by a factor of 2

   Answer: d

   4) Effective access time is directly proportional to

a) page-fault rate
b) hit ratio
c) memory access time
d) none of the mentioned

Answer: a

5) In the working set model, for:
2 6 1 5 7 7 7 7 5 1 6 2 3 4 1 2 3 4 4 4 3 4 3 4 4 4 1 3 2 3
if *DELTA* = 10, then the working set at time t1 (….3 4 4 4) is ___________.

a) {1, 2, 4, 5, 6}
b) {2, 1, 6, 7, 3}
c) {1, 6, 5, 7, 2}
d) {3, 4}

Answer: c

6) A process is thrashing if
a) it is spending more time paging than executing
b) it is spending less time paging than executing
c) page fault occurs
d) swapping cannot take place

Answer: a

7) What is the mounting of file system?
a) crating of a filesystem
b) deleting a filesystem
c) attaching portion of the file system into a directory structure
d) removing portion of the file system into a directory structure

Answer: c

8) Mapping of file is managed by
a) file metadata
b) page table
c) virtual memory
d) file system
Answer: a

9) The data structure used for file directory is called
a) mount table
b) hash table

c) file table
d) process table

Answer: b

10) When a process closes the file
a) per-process table entry is not removed
b) system wide entry's open count is decremented
c) all of the mentioned
d) none of the mentioned

Answer: b

2. (30 points) Q&A.

1) (5 points) Consider a paging system with the page table stored in memory.

a. If a memory reference takes 300 nanoseconds, how long does a

paged memory reference take?

b. If we add associative registers, and 70 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

Answer:

a. 600 nanoseconds; 300 nanoseconds to access the page table and 300 nanoseconds to access the word in memory.

b. Effective access time = 0.7 × (300 nanoseconds) + 0.3 × (600 nanoseconds) = 390 nanoseconds.

2) (6 points) On a system with paging, a process cannot access memory that it does not own; why? How could the operating system allow access to other memory? Why should it or should it not?

Answer: An address on a paging system is a logical page number and an offset. The physical page is found by searching a table based on the logical page number to produce a physical page number. Because the operating system controls the contents of this table, it can limit a process to accessing only those physical pages allocated to the process. There is no way for a process to refer to a page it does not own because the page will not be in the page table. To allow such access, an operating system simply needs to allow entries for non-process memory to be added to the process's page table. This is useful when two or more processes need to exchange data—they just read and write to the same physical addresses (which may be at varying logical addresses). This makes for very efficient interprocess communication.

3) (4 points) Discuss the hardware support required to support demand paging.

Answer: For every memory access operation, the page table needs to be consulted to check whether the corresponding page is resident or not and whether the program has read or write privileges for accessing the page. These checks would have to be performed in hardware. A TLB could serve as a cache and improve the performance of the lookup operation.

4) (5 points) A certain computer provides its users with a virtual-memory space of 2^32 bytes. The computer has 2^18 bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

Answer: The virtual address in binary form is

0001 0001 0001 0010 0011 0100 0101 0110

Since the page size is 2^12, the page table size is 2^20. Therefore the low- order 12 bits "0100 0101 0110" are used as the displacement into the page, while the remaining 20 bits "0001 0001 0001 0010 0011" are used as the displacement in the page table.

5) (5 points) Some file systems allow disk storage to be allocated at different levels of granularity. For instance, a file system could allocate 4 KB of disk space as a single 4-KB block or as eight 512-byte blocks. How could we take advantage of this flexibility to improve performance? What modifications would have to be made to the free- space management scheme in order to support this feature?

Answer:

Such a scheme would decrease internal fragmentation. If a file is 5 KB, then it could be allocated a 4 KB block and two contiguous 512-byte blocks. In addition to maintaining a bitmap of free blocks, one would also have to maintain extra state regarding which of the subblocks are currently being used inside a block. The allocator would then have to examine this extra state to allocate subblocks and coalesce the subblocks to obtain the larger block when all of the subblocks become free.

6) (5 points) 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, plus single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

Answer:

(12 * 8 /KB/) + (2048 * 8 /KB) + (2048 * 2048 * 8 /KB/) + (2048 * 2048 * 2048 * 8 /KB) = 64 terabytes

3. (30 points) Consider the following page reference string:

   7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 3

   Assuming demand paging with 3. Please illustrate each step that the following replacement algorithms work for this reference string and compute the page faults in each algorithm.

   1) FIFO replacement

   2) LRU replacement

   3) Optimal replacement

Answer:

1) FIFO replacement

7	2	3	1	2	5	3	4	6	7	7	1	0	5	4	6	2	3	0	3
7	7	7	**1**		1		1	**6**	6		6	**0**	0	0	**6**	6	6	**0**	
	2	2	2		**5**		5	5	**7**		7	7	**5**	5	5	**2**	2	2	
		3	3		3		**4**	4	4		**1**	1	1	**4**	4	4	**3**	3	

Page faults: 16

2) LRU replacement

7	2	3	1	2	5	3	4	6	7	7	1	0	5	4	6	2	3	0	3
7	7	7	**1**		1	**3**	3	3	**7**		7	7	**5**	5	5	**2**	2	2	
	2	2	2		2	2	**4**	4	4		**1**	1	1	**4**	4	4	**3**	3	
		3	3		**5**	5	5	**6**	6		6	**0**	0	**0**	**6**	6	6	**0**	

Page faults: 17

3) optimal replacement

7	2	3	1	2	5	3	4	6	7	7	1	0	5	4	6	2	3	0	3
7	7	7	**1**		1		1	1	1			1		1	1	1	1		
	2	2	2		**5**		5	5	5			5		**4**	**6**	**2**	**3**		
		3	3		3		**4**	**6**	**7**			**0**		0	0	0	0		

Page faults: 13

4. (20 points) Disk scheduling problem

   Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1280. The queue of pending requests, in FIFO order, is:

   2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Starting from the current head position (cylinder 2150), 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?

a) FCFS

b) SSTF

c) SCAN

d) C-SCAN

e) LOOK

f) C-LOOK

Answer:

The FCFS schedule is 2150, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681. The total seek distance is 13011.

The SSTF schedule is 2150, 2069, 2296, 2800, 3681, 4965, 1618, 1523, 1212, 544, 356. The total seek distance is 7586.

The SCAN schedule is 2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356. The total seek distance is 7492.

The C-SCAN schedule is 2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069. The total seek distance is 9917.

The LOOK schedule is 2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356. The total seek distance is 7424.

The C-LOOK schedule is 2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069. The total seek distance is 9137.