Fall Semester 2013 COMP 3511 Operating Systems Final Exam

Date: December 13, 2013 (Friday) **Time**: 12:30 pm - 3:00 pm

Name: ____________________ Student ID: __________________

Email: ____________________ Lecture Section: ______________

Note: Please write your *name*, *student ID*, and *section* **on this page**. Read the following instructions carefully.

1. This is a **CLOSED** book exam!
2. This examination paper consists of **6** questions and **11** pages (including this page).
3. You have **150** minutes to complete this exam.
4. All questions can be answered within the space provided on the examination paper. You may use back of the pages for your rough work. Be concise! This is NOT an essay contest.
5. Please read each question very carefully and answer the question clearly to the point.
6. Make sure that your answers are neatly written, legible, and readable.
7. Show all the steps used in deriving your answer, wherever appropriate.

Question	**Points**	**Score**
1	**10**	
2	**15**	
3	**20**	
4	**25**	
5	**10**	
6	**20**	
Total	**100**	

1. **[10 points] Multiple choices**

1) B 2) D 3) A 4) B 5)D 6)A 7)D 8)C 9)B 10)D

1) A cycle in a resource-allocation graph is ____.
   A) a necessary and sufficient condition for deadlock in the case that each resource has more than one instance
   B) a necessary and sufficient condition for a deadlock in the case that each resource has exactly one instance
   C) a sufficient condition for a deadlock in the case that each resource has more than one instance
   D) is neither necessary nor sufficient for indicating deadlock in the case that each resource has exactly one instance

   Answer: B

2) Which of the following is true of 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

3) 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

4) Given the logical address 0xAEF9 (in hexadecimal) with a page size of 256 bytes, what is the page offset? ____.
   A) 0xAE
   B) 0xF9
   C) 0xA
   D) 0xF900

   Answer: B

5) Belady's anomaly states that ____.
   A) giving more memory to a process will improve its performance
   B) as the number of allocated frames increases, the page-fault rate may decrease for all page replacement algorithms
   C) for some page replacement algorithms, the page-fault rate may decrease as the number of allocated frames increases

D) for some page replacement algorithms, the page-fault rate may increase as the number of allocated frames increases
Answer : D

6) _____ occurs when a process spends more time paging than executing.
A) Thrashing
B) Memory-mapping
C) Demand paging
D) Swapping
Answer: A

7) A mount point is _____.
A) a root of the file system
B) a location of a shared file system
C) only appropriate for shared file systems
D) the location within the file structure where the file system is to be attached.
Answer: D

8) The file-allocation table (FAT) used in MS-DOS is an example of _____.
A) contiguous allocation
B) indexed allocation
C) linked allocation
D) multilevel index
Answer: C

9) A RAID structure ____.
A) is primarily used for security reasons
B) is primarily used to ensure higher data reliability
C) is primarily used for performance reasons
D) is primarily used to decrease the dependence on disk drives
Answer: B

10) An interrupt priority scheme can be used to ____.
A) allow the most urgent work to be finished first
B) make it possible for high-priority interrupts to preempt the execution of a low priority interrupt
C) defer the handling of low-priority interrupt without masking off all interrupts
D) All of the above
Answer: D

2. [15 points] Deadlock

1) (3 points) Please briefly describe the three general ways that OS can handle a deadlock, each in one sentence.

Answer:

A deadlock can be prevented / avoided by using protocols to ensure that a deadlock will never occur/enter unsafe state.

A system may allow a deadlock to occur, detect it, and recover from it.

Lastly, an operating system may just ignore the problem and pretend that deadlocks can never occur.

2) (2 points) How to define that a system is in "safe state" in deadlock avoidance?

Answer:

A system is in safe state if there exists a sequence <P1, P2, …, Pn> of all the processes in the system such that for each Pi, the resources that Pi can still request can be satisfied by currently available resources + resources held by all the Pj, with j < i.

3) (10 points) Consider the following snapshot of a system:

	Allocation				Max			
	A	B	C	D	A	B	C	D
p_0	3	0	1	4	5	1	1	7
p_1	2	2	1	0	3	2	1	1
p_2	3	1	2	1	3	3	2	1
p_3	0	5	1	0	4	6	1	2
p_4	4	2	1	2	6	3	2	5

Using the Banker's algorithm, determine whether each of the following states is safe or not. If the system is safe, specify one execution sequence in that all the processes may complete. Otherwise, briefly justify why the state is unsafe.

a) (2 points) Available = (0, 3, 0, 1)

Answer:

Not safe. Only processes p_2, p_1 and p_3 can be finished. After that Available = (5, 11, 4, 2), no any remaining process can be finished.

b) (2 points) Available = (1, 0, 0, 3)

Answer：

Safe. A sequence of p_1, p_2, p_3, p_4, p_0 can be finished successfully.

c) (3 points) If Available = (1, 0, 0, 3), and if a request (1, 0, 0, 1) from process p_1 arrives, can this request be granted immediately?

Answer:

yes, request1 =(1, 0, 0 ,1) < available

	Allocation	Max	Need	Available
	A B C D	A B C D	A B C D	1 0 0 3
p_0	3 0 1 4	5 1 1 7	2 1 0 3	
p_1	2 2 1 0	3 2 1 1	1 0 0 1	
p_2	3 1 2 1	3 3 2 1	0 2 0 0	
p_3	0 5 1 0	4 6 1 2	4 1 0 2	
p_4	4 2 1 2	6 3 2 5	2 1 1 3	

As request1 = Need1, p1 can be finished upon this request, update Allocation, Need, Available:

	Allocation	Max	Need	Available
	A B C D	A B C D	A B C D	3 2 1 3
p_0	3 0 1 4	5 1 1 7	2 1 0 3	
~~p_1~~	~~3 2 1 1~~	~~3 2 1 1~~	~~0 0 0 0~~	
p_2	3 1 2 1	3 3 2 1	0 2 0 0	
p_3	0 5 1 0	4 6 1 2	4 1 0 2	
p_4	4 2 1 2	6 3 2 5	2 1 1 3	

The current state is safe, one successful execution sequence is p1, p2, p3, p4, p0.

d) (3 points) If Available = (1, 0, 0, 3), and if a request (1, 0, 0, 1) from process p_4 arrives, can this request be granted immediately?

Answer：

No, request4 =(1, 0, 0 ,1) < available

As request4< Need4, update Allocation, Need, Available:

	Allocation	Max	Need	Available
	A B C D	A B C D	A B C D	0 0 0 2
p_0	3 0 1 4	5 1 1 7	2 1 0 3	
p_1	2 2 1 0	3 2 1 1	1 0 0 1	
p_2	3 1 2 1	3 3 2 1	0 2 0 0	
p_3	0 5 1 0	4 6 1 2	4 1 0 2	
p_4	5 2 1 3	6 3 2 5	1 1 1 2	

The current state is unsafe, as no resource can satisfy the need of any process.

3. [20 points] Memory Management

1) (2 points) What do we refer as an address generated by a CPU? What is the hardware unit that maps a logical address to a physical address?
Answer:
Logical address (1mark), and memory-management-unit or MMU (**1mark**)

2) (5 points) Please briefly explain how external fragmentation occurs. Can you name two memory allocation methods that suffer from external fragmentation?
Answer:
As processes are loaded and removed from memory, the free memory space is broken into little pieces (1mark). External fragmentation occurs when there is enough total memory space to satisfy a new memory request, but the available spaces are not contiguous; storage is fragmented into a large number of small holes (2marks). Both contiguous allocation and segmentation strategies suffer from external fragmentation (2marks)
Best fit/ worst fit/ first fit are methods for finding memory holes to fit processes (ie. Solving a bin packing problem), it is a "component" of the more general contiguous memory allocation scheme. In the context of this question the correct answers are contiguous allocation and segmentation instead of them!

3) (3 points) Consider a logical address space of 16 pages with 1024 bytes per page; mapped onto a physical memory of 32 frames. How many bits are required in the logical address? How many bits are required in the physical address? Please justify your answer.
Answer:
16 pages requires 4 bits for pages (1mark), page and frame offset requires 10 bits (1mark), 32 frames requires 5 bits (1mark). So logical address has 14 bits, and physical address has 15 bits.
.

4) (10 points) Consider the following segment table:

Segment number	**Base**	**Length**
0	110	500
1	3500	18
2	7031	200
3	50	700
4	8402	100

What are the physical addresses for the following logical addresses?
a) 0, 300
b) 1, 34
c) 2, 201
d) 3, 170
e) 4,100

Answer:

a. 110+300=410 (2marks)

b. illegal reference, trap to operating system (2marks)

c. illegal reference, trap to operating system (2marks)

d. 50+170=220 (2marks)

e. 8402+100=8502 / or illegal reference (2marks)

4. [25] Virtual Memory

1) (3 points) What do we refer as an optimal page replacement or OPT algorithm? Why is this not commonly used? Why is it useful in practice?

   Answer:

   OPT selects a page to be replaced that will not be used in the longest time in the future (1 point). This is not commonly used because it requires future memory access knowledge, which is usually not available (1 point). This is used mostly for comparison with other page-replacement schemes or as a benchmark for comparison (1 point).

2) (2 points) What is a pre-paging scheme? What is the problem it resolves?

   Answer:

   Paging schemes, such as pure demand paging, result in large amounts of initial page faults as the process is started. Pre-paging is an attempt to prevent this high level of initial paging by bringing into memory, at one time, all of the pages that will be needed by the process at the start.

3) (3 points) What is the cause of thrashing? How does the OS detect thrashing? Once it detects thrashing, what may the OS do to eliminate this problem?

   Answer:

   Thrashing is caused by insufficient number of pages (or smaller than its minimum required pages) allocated to a process or processes (1 point). The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming, or this can be observed with high page faults (1 point). It can be eliminated by reducing the level of multiprogramming.

4) (9 points) Consider the following page reference string:

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

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms? Please illustrate each step.

a) LRU replacement
b) FIFO replacement
c) Optimal replacement

Answer:

LRU: 18
FIFO: 17
Optimal: 13

5) (4 points) Recall a clock algorithm can be used as an approximation for LRU algorithm in page replacement. Assume that you are monitoring the rate at which the pointer in the clock algorithm (which indicates the candidate page for replacement) moves. Can you explain the behavior when the pointer is moving fast, and the pointer is moving slow?

Answer:

If the pointer is moving fast, then the program is accessing a large number of pages simultaneously. It is most likely that during the period between the point at which the bit corresponding to a page is cleared and it is checked again, the page is accessed again and therefore cannot be replaced. This results in more scanning of the pages before a victim page is found. If the pointer is moving slow, then the virtual memory system is finding candidate pages for replacement extremely efficiently, indicating that many of the resident pages are not being accessed.

6) (4 points) Suppose that we use a demand-paged memory scheme, in which the page table is held in registers (TLB). Assume that it takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page has been modified since brought in. Memory access time is 100 nanoseconds. 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?

Answer:

$0.2\ \mu\text{sec} = (1 - P) \times 0.1\ \mu\text{sec} + (0.3P) \times 8\ \text{millisec} + (0.7P) \times 20\ \text{millisec}$

$0.1 = -0.1P + 2400\ P + 14000\ P$

$0.1 \simeq 16{,}400\ P$

$P \simeq 0.000006$

5. [10 point] File system

1) (2 points) The file system layers include I/O control, logical file system, basic file system, devices and file-organization module. What is the order from the highest level to the lowest level?
Answer:
logical file system, file-organization module, basic file system, I/O control and devices

2) (2 points) Recall OS uses two internal tables to keep track of open files, per-process and system-wide. Suppose there are two processes, where process A has two files open and process B has three files open. Two of the files are shared between the two processes. How many entries are in the per-process table of process A, in the per-process table of process B, and in the system-wide tables, respectively?
Answer: 2, 3, 3

3) (2 points) Why do all file systems suffer from internal fragmentation (disk)?
Answer:
Disk space is always allocated in fixed sized blocks. Whenever a file is written to disk, it usually does not fit exactly within an integer number of blocks so that a portion of a block is wasted when storing the file onto the device.

4) (4 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 one single, one double, and one triple indirect disk block respectively. 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)
= 96K+16M+32G+64T

6. [20 points] Disk and I/O

1) (3 points) Why is DMA used for devices that execute large transfers?
Answer:
Without DMA, programmed I/O must be used. This involves using the CPU to watch status bits and feed data into a controller register one byte at a time. Therefore, DMA was developed to lessen the burden on the CPU. DMA uses a special-purpose processor called a DMA controller and copies data in chunks.

2) (3 points) Please describe the technique in RAID that can enable multiple disks to be used to improve data transfer rate.
Answer:
One technique is bit-level striping. Bit-level striping consists of splitting the bits of each byte across multiple disks so that the data can be accessed from multiple disks in parallel.

Another method is block-level striping where blocks of a file are striped across multiple disks.

3) (9 points) 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 1805. 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, 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) LOOK. Please illustrate each step in order.

Answer：

a) FCFS: 2150, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681
Total seek distance =
(2150-1212)+(2800-1212)+(2800-544)+(1618-544)+(1618-356)+(4965-356)+(4965-3681)
= 13011
b) SSTF: 2150, 2069,2296,2800,3681, 4965, 1618, 1523, 1212, 544, 356
Total seek distance =(2150-2069）+(4965-2069)+(4965-356)=7586
c) SCAN: 2150, 2296, 2800, 3681, 4965, 4999, 2069, 1618, 1523, 1212, 544, 356
Total seek distance = (4999-2150)+(4999-356)=7492
d) LOOK: 2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356
Total seek distance = (4965-2150)+(4965-356)=7424

4) (5 points) Please use an example to illustrate why a nonblocking I/O system call is useful.

Answer:

If the user is viewing a web browser, then the application should allow keyboard and mouse input while it is displaying information to the screen. If nonblocking is not used, then the user would have to wait for the application to finish displaying the information on the screen before allowing any kind of user interaction.