Fall 2014 COMP 3511 Operating Systems

Final Examination Solutions

Date: December 19, 2014 (Friday) Time: 8:30 am - 11:00 am

Name: ____________________	Student ID: __________________
Email: ____________________	Lecture Section: ______________

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

1. This is a **CLOSED** book exam!
2. This examination paper consists of 5 questions and X pages (including this page).
3. You have **150** minutes to complete this exam.
4. Answer all questions within the space provided on the examination paper. You may use back of the pages for your rough work. Please 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 and legible.
7. Show all the steps used in deriving your answer, wherever appropriate.

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

1. [10 points] Multiple choices

1) Suppose that there are eleven resources available to three processes. At time 0, the following data is collected. The table indicates the process, the maximum number of resources needed by the process, and the number of resources currently owned by each process. Which of the following correctly characterizes this state?

Process	Maximum Needs	Currently Owned
P_0	10	4
P_1	3	1
P_2	6	4

A) It is safe.
B) It is not safe.
C) The state cannot be determined.
D) It is an impossible state.
Answer: A

2) 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 page 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

3) The ______ is an approximation of a program's locality.
A) locality model
B) working set
C) page fault frequency
D) page replacement algorithm
Answer: B

4) The _____ is the number of entries in the TLB multiplied by the page size.
A) TLB cache
B) page resolution
C) TLB reach
D) hit ratio
Answer: C

5) Systems in which memory access times vary significantly are known as ___________.
A) memory-mapped I/O
B) demand-paged memory
C) non-uniform memory access
D) copy-on-write memory
Answer: C

6) Which of the following is true of the direct-access method?
   A) It is the most common mode of access.
   B) It allows programs to read and write records in no particular order.
   C) Files are made up of variable-length records.
   D) It is not a good method for accessing large amounts of data quickly.
   Answer: B

7) Order the following file system layers in order of lowest level to highest level.
   [1] I/O control
   [2] logical file system
   [3] basic file system
   [4] file-organization module
   [5] devices
   A) 1, 3, 5, 4, 2
   B) 5, 1, 3, 2, 4
   C) 1, 5, 3, 4, 2
   D) 5, 1, 3, 4, 2
   Answer: D

8) In an environment where several processes may open the same file at the same time, _____.
   A) the operating system typically uses only one internal table to keep track of open files
   B) the operating system typically uses two internal tables called the system-wide and per-disk tables to keep track of open files
   C) the operating system typically uses two internal tables called the system-wide and per-process tables to keep track of open files
   D) the operating system typically uses three internal tables called the system-wide, per-disk, and per-partition tables to keep track of open files
   Answer: C

9) A disk with free blocks 1, 2, 5, 9, 15 would be represented with what bit map?
   A) 0011101110111110
   B) 1100010001000001
   C) 0110010001000001
   D) 1100100010000010
   Answer: C

10) Low-level formatting _____.
   A) does not usually provide an error-correcting code
   B) is usually performed by the purchaser of the disk device
   C) is different from physical formatting
   D) divides a disk into sections that the disk controller can read and write
   Answer: D

2. [20 points] Monitor and Deadlock

2.1. (2 points) What are the four necessary conditions for a deadlock to occur?

Answer: mutual exclusion, hold and wait, no preemption, and circular wait (0.5 points each)

2.2. (3 points) If a deadlock occurs, please elaborate the two methods that OS recover from the deadlock.

Answer: 1) Process termination: abort all processes or abort one process; the selection of the process to be aborted can depend on the process priority, CPU time elapse, resource used and etc. 2) Resource preemption: to select a victim and make sure the system can roll back to a safe state. (1.5 points each)

2.3. (7 points) The following codes implement condition variables using semaphores.

```
//The implementation of x.wait()
x_count++;
if (next_count > 0)
   signal(next);
else
   signal(mutex);
wait(x_sem);
x_count--;
```

```
//The implementation of x.signal()
if (x_count > 0) {
   next_count++;
   signal(x_sem);
   wait(next);
   next_count--;
}
```

a) Explain why we need x_count, x_sem, next, next_count, mutex respectively, please be specific (5points)

b) Please tell the difference between x.signal() and the signal() operation associated with semaphores (2 points)

Answer:

a) **x_count**: The variable is used for storing the number of threads that are executing x.wait(). In this way, the operation of x.signal() will be ignored if there is no process waiting in x.wait().
x_sem: It's a semaphore used to block the process of x.wait() until any other process calls the x.signal().
next: In order to block the latest process that executes x.signal()

next_count: The variable is used for storing the number of threads that are calling x.signal()
mutex: It's a semaphore used for the notification that a process is waiting for a x.signal()

b) The **x.signal()** operation associated with a monitor is not persistent in the following sense: if a signal is performed and if there are no waiting threads, the signal is simply ignored and the system does not remember that the signal ever took place. If a subsequent wait operation is performed, then the corresponding thread simply blocks. In semaphores, on the other hand, every signal results in a corresponding increment of the semaphore value even if there are no waiting threads. A future wait operation would immediately succeed because of the earlier increment.

2.4. (8 points) Consider the following snapshot of a system:

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P0	2 1 1 3	2 3 1 6	3 3 2 1
P1	1 0 0 1	4 2 1 2	
P2	3 1 1 1	5 2 5 2	
P3	0 3 0 1	4 4 1 3	
P4	1 4 3 2	3 6 6 5	

a) Using the banker's algorithm to determine whether the current system is safe or not, if yes, please demonstrate an order in which all the processes may complete. (2 point)

b) If a request from process P4 arrives for (0, 1, 1, 0), can this be granted immediately? If yes, show an execution order. (3 point)

c) If a request from process P0 arrives for (0, 1, 0, 1), can this be granted immediately? If yes, show an execution order. (3 point)

Answer:

a) The current system is safe. A sequence of p1, p3, p0, p4, p2 can be finished successfully.

b) Yes. Request = (0, 1, 1, 0) < Available, Thus Available can be updated.

	Allocation	Max	Need	Available
	A B C D	A B C D	A B C D	A B C D
P0	2 1 1 3	2 3 1 6	0 2 0 3	3 2 1 1
P1	1 0 0 1	4 2 1 2	3 2 1 1	
P2	3 1 1 1	5 2 5 2	2 1 4 1	
P3	0 3 0 1	4 4 1 3	4 1 1 2	
P4	1 5 4 2	3 6 6 5	2 1 2 3	

A feasible execution order is p1, p3, p0, p4, p2.

c) No. Request = (0, 1, 0, 1) < Available, Thus Available can be updated.

	Allocation	Max	Need	Available
	A B C D	A B C D	A B C D	A B C D
P0	2 2 1 4	2 3 1 6	0 1 0 2	3 2 2 0
P1	1 0 0 1	4 2 1 2	3 2 1 1	
P2	3 1 1 1	5 2 5 2	2 1 4 1	
P3	0 3 0 1	4 4 1 3	4 1 1 2	
P4	1 4 3 2	3 6 6 5	2 2 3 3	

The current state is NOT safe, since available resource D is less than the need of every process.

3. [25 points] Memory

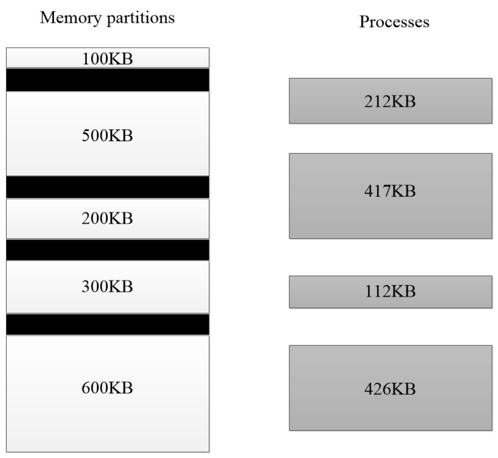
3.1. (3 points) In a paging scheme, there are many factors that influence the design of page size. Can you name any three factors, and briefly describe how they affect the page size.

Answer: 1) Page table size: larger page size leads to smaller page table; 2) Internal fragmentation: on average smaller page size results in smaller internal fragmentation; 3) I/O: larger page size usually reduces I/O time and the number of page faults, while smaller page size often results in less I/O and less total allocated memory.

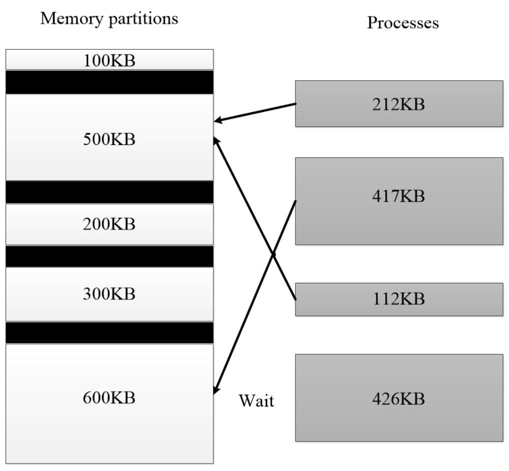
3.2. (3 points) Explain the differences between internal and external fragmentation, and name two memory allocation methods that suffer from external fragmentation.

Answer: External fragmentation occurs when there is sufficient total free memory to satisfy a memory request, yet the memory is not contiguous, so it cannot be assigned. (1 point) Internal fragmentation occurs when a process is assigned more memory than it has requested and the wasted memory fragment is internal to a process. (1 point) Both contiguous allocation and segmentation suffer from external fragmentation. (1 point)

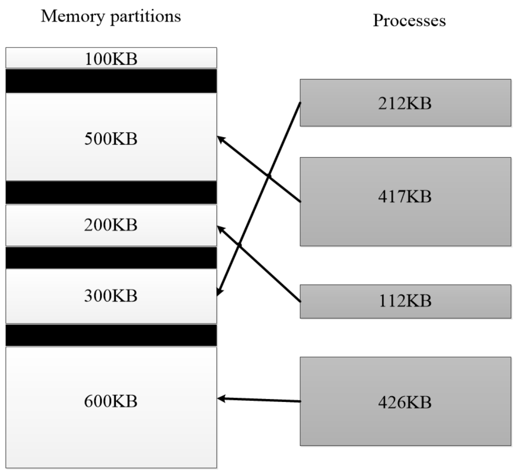
3.3. (4 points) Given 5 memory partitions of 100KB, 500KB, 200KB, 300KB and 600KB (in order) as following figure shows, how would first fit, best fit and worst fit algorithms place processes of 212KB, 417KB, 112KB, and 426KB (in order)? Please draw the figures to show the placement results of different allocation algorithms.



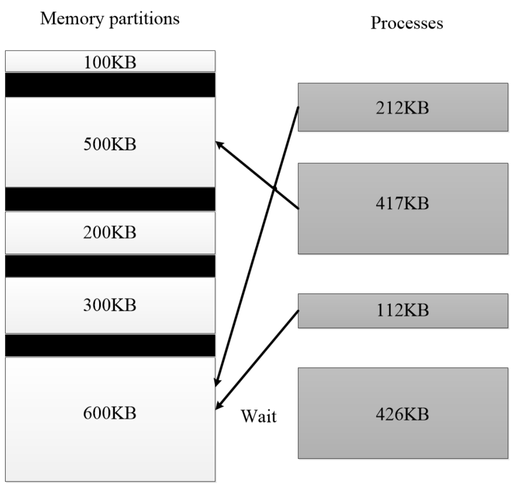
Answer: First fit



Best fit



Worst fit



3.4. (4 points) Consider the following segment table

Segment	**Base**	**Length**
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?
a) 0, 430
b) 4, 112

What are the logical addresses for the following physical addresses?
c) 2310
d) 1727

Answer:
a) 219+430=649
b) Illegal reference because 112>96
c) 1, 10
d) 3, 400

3.5. (4 points) Suppose that the TLB access time is 25 ns, and the memory access is 200 ns. What is the minimum TLB hit ratio in order to obtain the effective access time to be 260 ns?

Answer:
Assume that TLB hit ratio is P, we should make (200+25) P+ (25+200+200) (1-P) <260
Therefore, we get P>0.825. The minimum TLB hit ratio is 0.825

3.6. (7 points) In a 32-bit machine we subdivide the virtual address into 3 parts as follows

Page number		**Page offset**
11	11	10

We use a two-level page table (in memory) such that the first 11 bits of an address is an index into the first level page table and the next 11 bits are an index into a second level page table. Each page table entry is 64 bits in size.

a) What's the benefit of two-level paging scheme? (2 points)

b) How much space is occupied in memory by the page tables for a process that has 128MB of actual virtual address space allocated? Show your work with detailed explanation. (5 points)

Answer:

a) Two-level paging scheme can divide the large page table into many smaller pieces. Therefore, there is no need to allocate the whole page table contiguously in main memory.

b) The page size is 2^10=1024 B=1KB
For each first-level page table, it has 2^11=2048=2K entries. Since each entry is 8B (64 bits) in size, the size of the first-level page table is 16KB.
For each second-level page table, its size is also 16KB.
For a process that has 128MB, it has 128MB/1KB=128K pages. It needs 1 first-level page table and 128K/2K=64 second-level page table.
Therefore, the answer is (1+64)*16KB=1040KB

4. [25 points] Virtual Memory

4.1. (2 points) What is the utility of a dirty (or modify) bit?

Answer: A modify bit is associated with each page frame. If a frame is modified (i.e. written), the modify bit is then set. The modify bit is useful when a page is selected for replacement. If the bit is not set (the page was not modified), the page does not need to be written to disk. If the modify bit is set, the page needs to be written to disk when selected for replacement.

4.2. (2 points) Explain pre-paging scheme, the problem it tries to resolve, and in what cases pre-paging may offer advantage.

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. In s pre-paged pages, α is actually used, then the cost of the s* α saved page faults is greater than the cost of pre-paging s*(1- α).

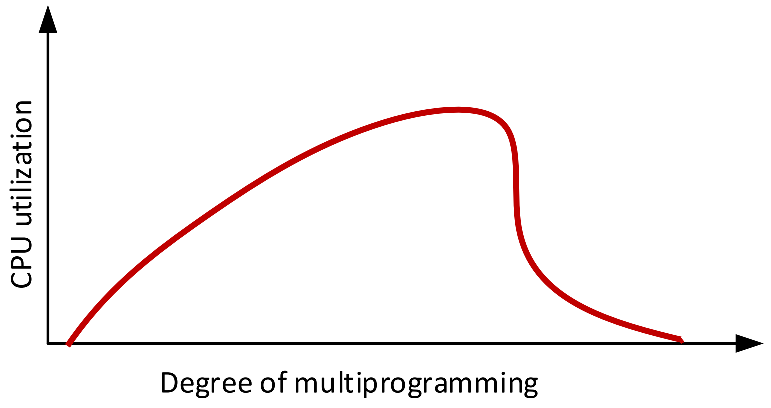
4.3. (3 points) Why is LRU replacement algorithm not practically useful? Please describe how the Second Chance algorithm can approximate a LRU algorithm.

Answer: The implementation of LRU requires either hardware assistance or updates for every memory reference, which is too frequent to be practically useful (1 points).
The Second Chance algorithm works as the FIFO algorithm, but instead of immediately paging out that page, it checks to see if its referenced bit is set. If it is not set, the page is swapped out. Otherwise, the referenced bit is cleared. Since the referenced bits of recently used pages are being set recently, it is unlikely to swap them out at first. That's how the Second Chance algorithm can approximate a LRU algorithm (2 points).

4.4. (4 points) Assume we have a demand-paged memory. The page table is held in registers. It takes 6 milliseconds to service a page fault if an empty page is available or the replaced page is not modified and 14 milliseconds if the page is modified. Memory access time is 50 nanoseconds. Assume that the page replaced is modified 50 percent of time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

Answer:
$(1\text{-}p) \times 50 + p \times (0.5 \times 6 \times 10^6 + 0.5 \times 14 \times 10^6) \leqslant 200$
$p \leqslant 1.5 \times 10^{-5}$

4.5. (4 points) Please explain the phenomenon of the entire curve, why the CPU utilization drops sharply at the end and how to solve this problem.



Answer:
Root cause: initially CPU utilization increases as the degree of multiprogramming, later on thrashing sets in, processes are busy swapping pages in and out due to the lack of enough pages, and thus CPU utilization drops (2 points).
We must decrease the degree of multiprogramming, we can also limit effects by using local or priority page replacement (2 points).

4.6. (10 points) Consider the following page reference string:

0 2 1 3 0 1 4 0 1 2 3 4

a) Assume demand paging with three frames, how many page faults would occur for FIFO, Optimal, and LRU replacement algorithms respectively?

Answer: 10, 7, 9 (1, 1.5, 1.5 points);

b) Assume demand paging with four frames, how many page faults would occur for FIFO, Optimal, and LRU replacement algorithms respectively?

Answer: 7, 6, 8 (1, 1.5, 1.5 points);

c) Compare (a) and (b), what do you observe?

Answer: We can reduce the number of page faults by using more frames (1 points);.

5. [20 points] File Systems and Secondary Storage

5.1. (2 points) What is the advantage and disadvantage of contiguous allocation and link allocation?

Contiguous allocation is simple, and supports both sequential and random access. It suffers several problems similar to contiguous memory allocation: difficult to find a contiguous space for a new file, dynamic storage allocation problem (first fit or best fit), and external fragmentation, difficult to grow the file size.
Linked allocation stores file by block with pointers to the first and last blocks, which solves size-declaration in contiguous allocation. But it only allows sequential access to the file, and pointers in each block needs extra space.

5.2. (2 points) In RAID, why is striping useful? What is the purpose of mirror disks?

In disk striping, all the data is spread out in chunks across all the disks in the RAID set. It provides great performance because you spread the load of storing data onto more

physical drives, and the system has more spindles to read from or write to than when data is accessed from a single drive.
For disk mirroring, all the data is written to at least two separate physical disks. The disks are essentially mirror images of each other. If one of the disks fails, the other can be used to retrieve data, so there is no loss of data in case of single disk failure or fault.

5.3. (8 points) Consider a file system that uses inodes to represent files. Disk blocks are 6KB 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.

a) What is the maximum size of a file that can be stored in this file system?

(12 * 6KB) + (6KB/4B * 6KB) + (6KB/4B *6KB/4B * 6KB) + (6KB/4B * 6KB/4B * 6KB/4B * 6KB) = 2.228e13 bytes or 22.28 TB

b) Suppose one third of all files are exactly 4-KB and the rest files are exactly 6KB, what fraction of disk space would be wasted? (Consider only blocks used to store data)
Both 4KB and 6KB files will use 6KB space. For each 6KB file, 0 space is wasted, for each 4KB file, 2KB space is wasted, thus the fraction wasted is 0/6 * 2/3 + 2/6 * 1/3 = 1/9 = 11.1%

c) Does it help to reduce the fraction of wasted disk space in (b) if we change the block size to 3KB? Justify your answer.
No, does not help at all. Both 4KB and 6KB files will still use 6KB space. For each 6KB file, 0 space is wasted, for each 4KB file, 2KB space is wasted, thus the fraction wasted is 11.1%.

5.4. (8 points) Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2500, and the previous request was at cylinder 2650. 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?

FCFS, SSTF, SCAN, C-SCAN, LOOK and C-LOOK

Starting from the current head position:

FCFS: The schedule is 2500, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681. The total distance is 13361.

SSTF: The schedule is 2500, 2296, 2069, 1618, 1523, 1212, 544, 356, 2800, 3681, 4965. The total distance is 2500 - 356 + 4965 - 356 = 6753.

SCAN: The schedule is 2500, 2296, 2069, 1618, 1523, 1212, 544, 356, 0, 2800, 3681, 4965. The total distance is 2500 + 4965 = 7465.

LOOK: The schedule is 2500, 2296, 2069, 1618, 1523, 1212, 544, 356, 2800, 3681, 4965. The total distance is 6753.

C-SCAN: The schedule is 2500, 2296, 2069, 1618, 1523, 1212, 544, 356, 0, 4999, 4965, 3681, 2800. The total distance is 2500 + 4999 + 4999 – 2800 = 9698.

C-LOOK The schedule is 2500, 2296, 2069, 1618, 1523, 1212, 544, 356, 4965, 3681, 2800. The total distance is 2500 – 356 + 4965 – 356 + 4965 – 2800 = 8918.