Spring 2015 COMP 3511 Operating Systems
Final Examination Solution

Date: May 20, 2015 (Wednesday) Time: 12:30 - 14:30

Name: ____________________ Student ID: __________________

Note: Please write your name and student ID on this page. Please 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 120 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 necessary in deriving your answer(s), wherever appropriate.

Question	Points	Score
1	10	
2	15	
3	25	
4	25	
5	13	
6	12	
Total	100	

1. [10 points] Multiple Choices

1.1. Which of the following page replacement algorithms suffers from Belady's Anomaly?
   A) Optimal replacement
   B) LRU
   C) FIFO
   D) Both optimal replacement and FIFO
   Answer: C

1.2. _____ is the concept in which a process is copied into main memory from the secondary memory according to the requirement.
   A) Paging
   B) Swapping
   C) Segmentation
   D) Demand paging
   Answer: D

1.3. A locality is ____________.
   A) a set of pages that are actively used together
   B) a space in memory
   C) an area near a set of processes
   D) None of these
   Answer: A

1.4. 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

1.5. What is the mounting of a file system?
   A) Crating of a file system
   B) Deleting a file system
   C) Attaching portion of the file system into a directory structure
   D) Removing portion of the file system into a directory structure
   Answer: C

1.6. The first fit and best fit algorithms suffer from _________.
   A) internal fragmentation
   B) external fragmentation
   C) starvation
   D) All of these.
   Answer: B

1.7. Which of the following is not one of the three major methods of allocating disk space?
   A) Contiguous allocation
   B) Linked allocation
   C) Indexed allocation

D) Hashed allocation

Answer: D

1.8. The time taken to move the disk arm to the desired cylinder is called the _________.

A) positioning time

B) random access time

C) seek time

D) rotational latency

Answer: C

1.9. Which bit map represents a disk with free blocks 1, 2, 5, 9, 15?

A) 0011101110111110

B) 0110010001000001

C) 1100010001000001

D) 1100100010000010

Answer: B

1.10. The disk bandwidth is __________.

A) the total number of bytes transferred

B) total time between the first request for service and the completion on the last transfer

C) the total number of bytes transferred divided by the total time between the first request for service and the completion on the last transfer

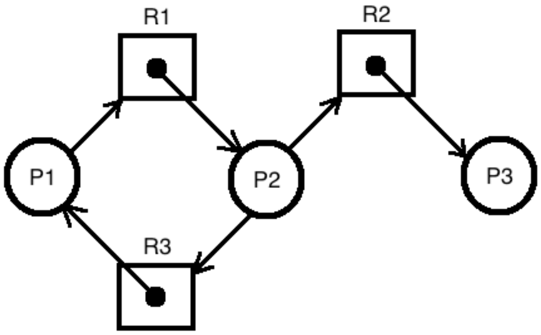
D) None of these

Answer: C

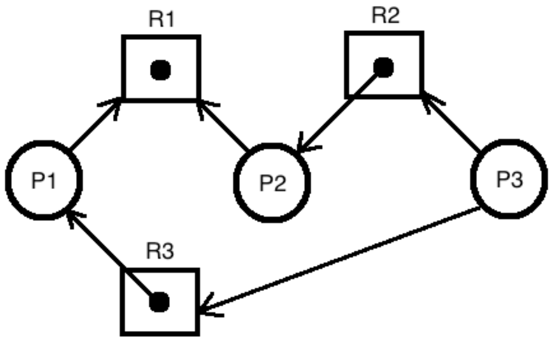
2. [15 points] Deadlock

2.1. (4 points) Consider the following two resource allocation graphs

a) (2 points) Is there a deadlock in this resource allocation graph? Please briefly explain your answer.



b) (2 points) Is there a deadlock in this resource allocation graph? Please briefly explain your answer.



Answer:

a) Yes. There is a cycle in the graph. In a resource allocation graph, if each resource type has exactly one instance, then a cycle implies that a deadlock has occurred.

b) No. There is no cycle in the graph.

2.2. (3 points) What are the possible actions that operating system can take to prevent a deadlock to occur? Which one seems to be more feasible in term of resource utilization?

Answer: In order to prevent deadlock to occur, we need to ensure at least one of the four necessary conditions does not hold (1 point). Since mutual exclusion must be enforced for non-shared resource, hold and wait or no preemption requires either resources to be allocated at once or to be released and re-allocated, which will result in poor resource utilization (1 point), so the only feasible solution would be to impose a ordered list of resource requests, so to avoid circular wait (1 point)

2.3. (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) (2 points) Use 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.

b) (2 points) If a request from process P3 arrives for (1, 0, 0, 1), can this be granted immediately? If yes, show an execution order.

c) (2 points) If a request from process P3 arrives for (0, 1, 0, 0), can this be granted immediately? If yes, show an execution order.

d) (2 points) If a request from process P0 arrives for (1, 0, 0, 0), can this be granted immediately? If yes, show an execution order.

Answer:

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

b) No. Request = (1, 0, 0, 1) > Avaliable.

c) Yes. Request = (0, 1, 0, 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	1 0 0 4	2 0 1 5	1 0 1 1	1 0 0 0
P1	0 2 1 2	2 2 1 3	2 0 0 1	
P2	2 1 0 3	2 1 3 4	0 0 3 1	
P3	0 2 2 1	1 2 2 3	1 0 0 2	
P4	0 2 0 2	1 2 0 2	1 0 0 0	

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

d) No. Request = (1, 0, 0, 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 0 0 4	2 0 1 5	0 0 1 1	0 1 0 0
P1	0 2 1 2	2 2 1 3	2 0 0 1	
P2	2 1 0 3	2 1 3 4	0 0 3 1	
P3	0 1 2 1	1 2 2 3	1 1 0 2	
P4	0 2 0 2	1 2 0 2	1 0 0 0	

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

3. [25 points] Memory Management

3.1. (6 points) Two-level paging scheme

In a 32-bit machine we subdivide the virtual address as follows.

page	number	page offset
10-bit	10-bit	12-bit

We use a two-level page table (in memory) such that the first 10 bits of an address is an index into the first level page table and the next 10 bits are an index into a second level page table. Suppose that each page table entry size is 4 bytes.

a) (1 point) What's the page size in such a system?

Answer: Because 12 bits are used to offset into a page, the page size is 2^{12} = 4096 bytes = 4KB.

b) (2 points) How much memory does the first-level page table occupy?

Answer:

Since an index in the first level or second level page table is 10 bits, there must be 2^{10} =1024 =1K entries per page table (in both 1st and 2rd level page table). (1 point)

Since each page entry is 4 bytes, a first-level page table occupies 4 bytes * 1K entries = 4 KB. (1 point)

c) (3 points) How much space is occupied in memory by the second-level page tables for a process that has a size of 16MB?

Answer:

Since the process uses 16 MB = 2^{24} bytes and each page is 4KB = 2^{12} bytes, there must be $2^{24}/2^{12} = 2^{12}$ pages. (1 point)

Thus, we need a sufficient number of second level page tables to hold 2^{12} entries. Since each second level page table has 2^{10} entries, we need $2^{12} / 2^{10} = 2^2 = 4$ second level page tables. (1 point)

So there are 4 second-level page entries in total. And since each entry is 4 Byte, the space occupied by the second-level page tables are 4*4 Byte = 16 KB. (1 point)

3.2. (4 points) Consider a system implemented paging scheme with TLB.

Let's make following assumptions:

- Page fault rate is 10%
- Memory access time is 2 microseconds
- Hit radio of TLB is 80%, and the access time of TLB can be ignored.
- It takes 10 microseconds to serve the page fault on average.

Now please calculate the effective access time (EAT) of the memory system. (4 points)

Answer:

No page fault 90% (1 point)

Hit in TLB 80%, 2ms

Miss in TLB, search in the page table 20%, 4ms (1 point)

Page fault 10%

Average handling time, 10ms (1 point)

EAT = 0.9 * (0.8 * 2 + 0.2 * 4) + 0.1 * 10 = 3.16 ms. (1 point)

3.3. (5 points) Consider the following segment table

Segment	Base	Length
0	132	300
1	1672	150
2	1960	90
3	445	78
4	598	696

What are the physical addresses for the following logical addresses?

a) 0, 230

b) 3, 85
c) 4, 100

What are the logical addresses for the following physical addresses?
d) 1800
e) 2030

Answer:
a) 132+230=362
b) Illegal reference because 85>78
c) 598+100=698
d) 1, 128
e) 2, 70

3.4. (3 points) Please briefly describe the advantages and disadvantages between a paging scheme and a segmentation scheme.

Answer:
Paging scheme enables a better memory utilization, while a segmentation scheme resembles the logical structure of a program (1 point). The internal fragmentation problem in a paging scheme is usually less severe than the external fragmentation problem in a segmentation scheme (1 point). A page table is often much larger than a segmentation table, which occupies more memory (1 point).

3.5. (3 points) In the classical dynamic storage-allocation problem, what are the three common methods to satisfy a memory request of a size n (variable)? Please briefly explain each.

Answer:
First-fit, worst-fit and best-fit.
First-fit: Allocate the first hole that is big enough (1 point).
Best-fit: Allocate the smallest hole that is big enough. We must search entire list, unless they are ordered by size. It produces the smallest leftover hole (1 point).
Worst-fit: Allocate the largest hole. We must also search entire list. It produces the largest leftover hole (1 point).

3.6. (4 points) The program along with its data have to be bind to actual memory address before it can be executed. Please briefly explain the three stages of addressing binding. Under which address binding scheme, the logical address and physical address are different?

Answer: (1) compile time, in which absolute code will be generated. If the location changes later, the code has to be recompiled (1 point); (2) load time, in which compiler must generate relocatable code. If starting address changes, we need to reload the code (1 point); (3) execution time, in which the process can moved during its execution from one memory address to another (1 point).
The execution time address binding results different logical and physical addresses (1 point).

4. [25 points] Virtual Memory

4.1. (3 points) What are the differences between a global replacement and a local replacement in page replacement? What kind of replacement policy are we using when we discuss FIFO or LRU replacement algorithm?

Answer: As for global replacement, each process selects a replacement frame from the set of all frames in the memory (1 point). However, as for local replacement, each process selects from only its own set of allocated frames (1 point).
LRU replacement algorithm uses local replacement (1 point).

4.2. (4 points) What is a working-set model trying to estimate? How to determine whether the system is thrashing with a working-set model?

Answer: Suppose we have selected an adequate working-set window size, the working-set or WSS_i (working-set for a process i) would roughly capture the **current locality**, i.e., the minimum number of pages a process needs at the moment (1 point). This allows the OS to dynamically adjust the number of frames allocated to each process based on the current working-set (1 point).
Since WSS_i is a good approximation of the number of pages needed for a process i at a given time, the total memory needed at the time by all processes is D = \sum WSS_i. If D > m (the actual physical memory size), thrashing will occur, in which at least one process is constantly in short of memory, thus its pages are swapped in and out (2 point).

4.3. (9 points) Reference string, to compute the page fault for FIFO, optimal and LRU. Consider the following sequence of page references in term of page numbers:

1, 2, 1, 4, 3, 1, 5, 3, 2, 1, 4, 1, 3, 2, 1, 4

Suppose there are **3** frames allocated for this process, please illustrate the contents of the frames under 3 different page replacement algorithms and compute the **number of page faults** for each algorithm.

a) (3 points) FIFO
b) (3 points) LRU
c) (3 points) Optimal

Answer:

a) FIFO

Frame\Page	1	2	1	4	3	1	5	3	2	1	4	1	3	2	1	4
F1	1	1	1	1	3	3	3	3	2	2	2	2	3	3	3	3
F2		2	2	2	2	1	1	1	1	1	4	4	4	2	2	2
F3				4	4	4	5	5	5	5	5	1	1	1	1	4

Number of page faults: 12

b) LRU

Frame\Page	1	2	1	4	3	1	5	3	2	1	4	1	3	2	1	4
F1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	4
F2		2	2	2	3	3	3	3	3	3	4	4	4	2	2	2
F3				4	4	4	5	5	5	1	1	1	1	1	1	1

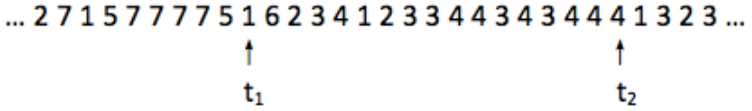
Number of page faults: 11

c) Optimal

Frame\Page	1	2	1	4	3	1	5	3	2	1	4	1	3	2	1	4
F1	1	1	1	1	1	1	5	5	5	1	1	1	1	1	1	1
F2		2	2	2	2	2	2	2	2	2	4	4	4	4	4	4
F3				4	3	3	3	3	3	3	3	3	3	2	2	2

Number of page faults: 8

4.4. (4 points) In a working set model, for the page reference table shown below:



… 2 7 1 5 7 7 7 7 5 1 6 2 3 4 1 2 3 3 4 4 3 4 3 4 4 4 1 3 2 3 …

If the parameter△ of the working set window is 10,

a) (2 points) What is the working set at time t_1 (….7 7 5 1)?

b) (2 points) What is the working set at time t_2 (….3 4 4 4)?

Answer:

a) WS(t1) = {1, 7, 5, 2}

b) WS(t2) = {3, 4}

4.5. (5 points) Please discuss three factors that affect the selection of page size in a paging scheme.

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.

5. [13 points] File System

5.1. (4 points) What are the differences between the linked list and hash table for implementing directory?

Answer:

Linked list: A linear list of file names with pointers to the data blocks. This is simple to program but time-consuming to execute (searching a file requires linear search, caching the most recently used directory information can be useful as directory information is frequently used)

Hash Table: A linear list stored the directory entries, but a hash is used which takes a value computed from the file name and return a pointer to the file name in the linear list. This greatly reduces the directory search time, also makes it easier for insertion and deletion. Collisions are possible, in which two or more file names are hashed to the same location. We can further use a linked list (with the same hashed values for different file names) instead of a single value to resolve the collision.

5.2. (4 points) For disk block allocation, please compare link allocation and file allocation table (FAT), what is the similarity and difference between these two methods?

Answer:

Linked allocation: stores file by block with pointers to the first and last blocks. This solves size-declaration in contiguous allocation. It only allows sequential access to the file and pointers in each block needs extra space.

A variation of the linked allocation is the use of File Allocation Table (FAT) (MS-DOS), in which the number of entries in FAT corresponds to the number of blocks in a file. The directory entry contains the block number of the first block of the file. The FAT (or the table) entry indexed by that block number contains the block number (disk address) of the next block allocatedto the file.

5.3. (5 points) Consider a file system that uses inodes to represent files. Disk blocks are 8KB in size. The disk address is 32 bits. This file system has 10 direct disk blocks, as well as one indirect, one double indirect and one triple indirect disk block.
What is the maximum size of a file that can be stored in this file system?

Answer:

32bits = 4 bytes = 4B

(10 * 8KB) + (8KB/4B * 8KB) + (8KB/4B *8KB/4B * 8KB) + (8KB/4B * 8KB/4B * 8KB/4B * 8KB)

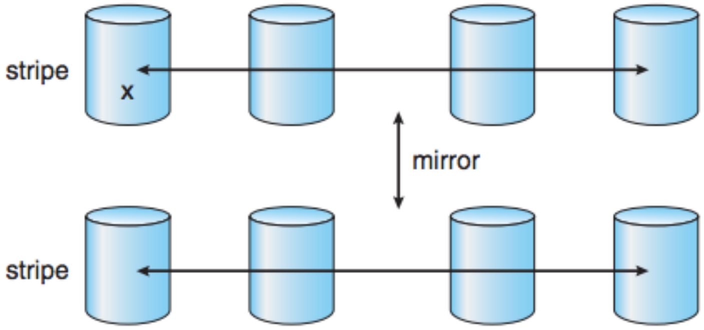
= 80*1024B + 16*1024*1024B + 32*1024*1024*1024B + 64*1024*1024*1024*1024B

= 7.04e13 bytes or 70.04TB

6. [12 points] The secondary storage system

6.1. (4 points) Please briefly explain the difference between:

a) RAID 0+1



a) RAID 0 + 1 with a single disk failure.

b) RAID 0 (non-redundant stripping)
c) RAID 1 (mirrored disks)

Answer:
RAID 1 offers redundancy through mirroring, i.e., data is written identically to two drives. RAID 0 offers no redundancy and instead uses striping, i.e., data is split across all the drives. This means RAID 0 offers no fault tolerance; if any of the constituent drives fails, the RAID unit fails.
In RAID 0+1, the RAID0 is on the lower layer and the mirroring RAID1 is on the upper layer. Should one disk fail on a RAID 0+1, you would be forced to replace both RAID 0 disks on that side of the parent RAID 1. The RAID 1 would then rebuild the two new disks. RAID0 is like JBOD in this sense, where if you lose one disk, the entire virtual sub-array has failed. RAID 0+1 would, therefore, only be practically able to lose one disk. Should a second disk fail on the other side of the array for some reason, even if it held the other half of the RAID 0 data, the controller will fail the RAID 0 portion on that side as well and you will have no disks left.

6.2. (8 points)Disk scheduling problem
Suppose that a disk drive has 6000 cylinders, numbered 0 to 5999. The drive is currently serving a request at cylinder 2500, and the previous request was at cylinder 1400. The queue of pending requests, in FIFO order, is:

500, 4000, 1500, 2000, 5500, 3500, 5000, 4500

Starting from the current head position (2500), 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 2500, 500, 4000, 1500, 2000, 5500, 3500, 5000, 4500. The total seek distance is 2000+3500+2500+4000+2000+1500+500 = 16000.

The SSTF schedule is 2500, 2000, 1500, 500, 3500, 4000, 4500, 5000, 5500. The total seek distance is 2000+5000 = 7000.

The SCAN schedule is 2500, 3500, 4000, 4500, 5000, 5500, (5999), 2000, 1500, 500. The total seek distance is 3499+5499= 8998.

The C-SCAN schedule is 2500, 3500, 4000, 4500, 5000, 5500, (5999), (0), 500, 1500, 2000. The total seek distance is 3499+5999+2000 = 11498.

The LOOK schedule is 2500, 3500, 4000, 4500, 5000, 5500, 2000, 1500, 500. The total seek distance is 3000+5000 = 8000.

The C-LOOK schedule is 2500, 3500, 4000, 4500, 5000, 5500, 500, 1500, 2000.
The total seek distance is 3000+5000+1500=9500.