Fall 2015 COMP 3511 Operating Systems
Final Examination Solutions

Date: Dec 11, 2015 (Friday) Time: 8:30 - 10:30 a.m.

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 7 questions and 12 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 used in deriving your answer, wherever appropriate.

Question	**Points**	**Score**
1	10	
2	12	
3	22	
4	23	
5	14	
6	13	
7	6	
Total	100	

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

1.1. 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 once instance

D) neither necessary nor sufficient for indicating deadlock in the case that each resource has exactly one instance

Answer: B

1.2. Suppose that there are ten 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
P0	10	4
P1	3	1
P2	6	4

A) It is safe.

B) It is not safe.

C) The state cannot be determined.

D) It is an impossible state.

Answer: B

1.3. _____ is the method of binding instructions and data to memory performed by most general-purpose operating systems.

A) Interrupt binding

B) Compile time binding

C) Execution time binding

D) Load-time binding

Answer: C

1.4. 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

1.5. 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

1.6. 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

1.7. Suppose that the operating system uses two internal tables to keep track of open files. Process A has two files open and process B has three files open. Two files are shared between the two processes. How many entries are in the per-process table of process A, the per-process table of process B, and the system-wide tables, respectively?

A) 5, 5, 5

B) 2, 3, 3

C) 2, 3, 5

D) 2, 3, 1

Answer: B

1.8. 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

1.9. A RAID structure ____.

A) is primarily used for security reasons

B) is primarily used to ensure higher data reliability

C) stands for redundant arrays of inexpensive disks

D) is primarily used to decrease the dependence on disk drives

Answer: B

1.10. Host-attached storage is _____.

A) a special purpose storage system that is accessed remotely over a data network

B) not suitable for hard disks

C) accessed via local I/O ports

D) not suitable for use in raid arrays

Answer: C

2. [**12 points**] Deadlock

2.1. Please briefly explain the four necessary conditions for deadlock. (4 points)

Answer:

Mutual exclusion (0.5pt) – Only one thread at a time can hold a given resource (0.5pt)

Hold and Wait (0.5pt) – Thread holding at least one resource is waiting for another one (0.5pt)

No Preemption (0.5pt) – Resources are released only voluntarily by the thread holding the resource, after thread is finished with it(0.5pt)

Circular Wait (0.5pt)– There exists a set {T_1, …, T_n} of waiting threads, with T_1 waiting for a resource held by T_2, T_2 waiting for a resource held by T_3, …, T_n waiting for a resource held by T_1(0.5pt)

2.2. Consider the following snapshot of a system:

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	1	1	2	0	1	2	3	1	0	1	0
P_1	1	0	0	2	2	0	0	2				
P_2	0	1	0	1	0	2	0	1				
P_3	1	1	0	0	2	2	0	0				
P_4	0	1	2	0	0	2	2	0				

Please answer the following questions using the Banker's algorithm (8 points)
(Note: a simple Yes or No without justification receives 0 mark).

a) What is the content of the Need Matrix denoting the number of resources needed by each process? (3 points)

Answer:

	Need			
	A	B	C	D
P0	0	0	1	1
P1	1	0	0	0
P2	0	1	0	0
P3	1	1	0	0
P4	0	1	0	0

b) Is the system in a safe state? If the answer is yes, please give a safe sequence and resources available after each process finished. If the answer is no, please specify the processes that might involve in a deadlock (unsafe) (5 points)

Answer: Yes. The safe sequence: < P1, P0, P2, P3, P4>

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

3. [**22 points**] Memory Management

3.1. Consider a computer that utilizes a 64-bit logical address space, with page size of 4KB and each page table entry occupies 4 bytes.

a) With a single-level paging scheme, how many entries are there in the page table and what is the size of the page table? (3 points)

Answer: Number of entries = 2^{64} / 2^{12} = 2^{52} (1 pt)

Space needed = 2^{52} * 4 bytes = = 16PB (2 pt)

b) Consider the following paging scheme instead:

Unused	2^{nd} outer page	Outer page	Inner page	Offset
u	p_1	p_2	p_3	d
16 bits	16 bits	10 bits	10 bits	12 bits

How much space is occupied in the memory by the page tables for a process that has 8192MB of actual virtual address space allocated? Show detailed explanation. (6 points)

Answer:

Process uses 8192MB = 2^{33} bytes, page size = 2^{12} bytes, number of pages = $2^{33}/2^{12}$ = 2^{21} pages (1pt)

To hold 2^{21} pages, number of inner page tables needed = $2^{21}/2^{10}$ = 2^{11} (1pt)

To hold 2^{11} inner page table, number of outer page tables needed = $2^{11}/2^{10}$ = 2 (1pt)

Number of 2^{nd} outer page tables needed = 1 (1pt)

Total space occupied in memory = $(1*2^{16} + 2*2^{10} + 2^{11}*2^{10})$ * 4 bytes = 8658944 bytes = 8456KB = 8.258MB (2pt)

3.2. Consider the following segment table:

	Limit	**Base**
0	0x1000	0x1400
1	0x0400	0x6300
2	0x0400	0x4300
3	0x1100	0x3200
4	0x1000	0x4700

What are the physical addresses (in decimal) for the following logical addresses (in decimal)? (3 points)

a) 1, 1048
b) 3, 1048

What are the logical addresses (in decimal) for the following physical addresses (in decimal)? (3 points)

c) 16384
d) 6144

Answer: (1.5pt each)

a) Illegal reference
b) 13848
c) 3, 3584
d) 0, 1024

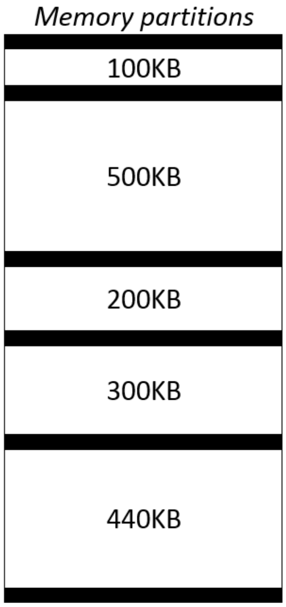
3.3. Compare the types of fragmentation in memory management, and describe the reasons why pure segmentation and pure paging suffer from different types of fragmentation. (4 points)

Answer: (1pt each)

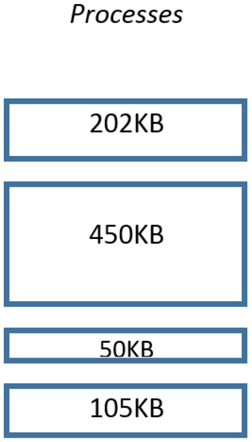
External fragmentation: total memory space exists to satisfy a request, but it is not contiguous.
Internal fragmentation: allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used.

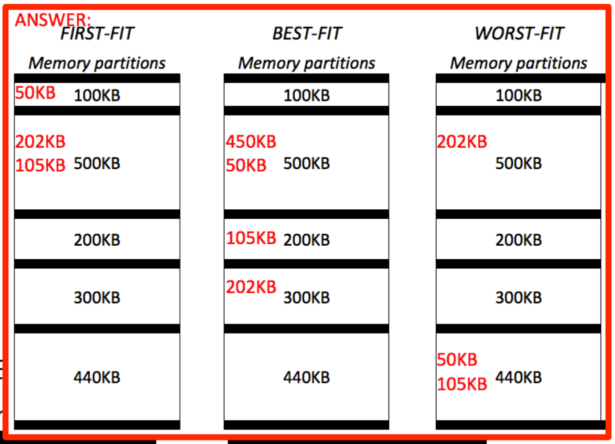
Pure segmentation: A segment of a process is laid out contiguously in physical memory and external fragmentation would occur as segments of dead processes are replaced by segments of new processes.
Pure paging: Processes are allocated in page granularity and if a page is not completely utilized, it results in internal fragmentation and a corresponding wastage of space.

3.4. Given five memory partitions of 100KB, 500KB, 200KB, 300KB, and 440KB (in order). How would each of the first-fit, best-fit, and worst-fit algorithms place processes of 202KB, 450KB, 50KB, and 105KB (in order)? Please mark on the bottom figures.
Note that not all of the processes may be allocated immediately. (3 points)

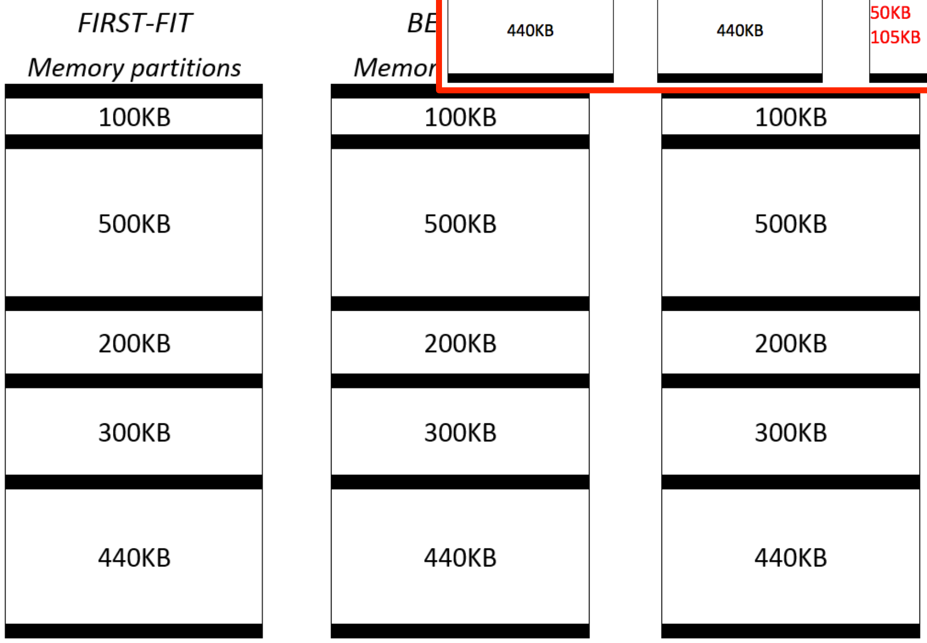


(Not drawn in scale)





Mark your answers below:



(Not drawn in scale) (Not drawn in scale) (Not drawn in scale)

4. [**23 points**] Virtual Memory

4.1. Would increasing the number of frames always decrease the number of page faults for a particular reference string for FIFO? Why or why not? (2 points)

Answer: No. (1pt) Belady's anomaly states that it is possible to have more resources, and worse access behavior. (1pt)

4.2. Is it possible for a process to have two working sets, one representing data and another representing code? Explain the benefit or the drawback. (2 points)

Answer: Yes. (1pt) As an example, the code being accessed by a process may retain the same working set for a long period of time. However, the data the code accesses. (1pt)

4.3. Please compare and contrast the second-chance algorithm for page replacement and the FIFO page replacement algorithm. (3 points)

Answer: The second-chance algorithm is based on the FIFO replacement algorithm (1 pt) and even degenerates to FIFO in its worst-case scenario. In the second-chance algorithm, a FIFO replacement is implemented along with a reference bit. If the reference bit is set, then it is cleared, the program moves along until a page with a cleared reference bit is found and subsequently replaced. (2 pt)

4.4. What is thrashing? What is local replacement algorithm? Can local replacement algorithm alleviate the problem of thrashing? (3 points)

Answer: Thrashing occurs when processes are busy swapping pages in and out due to the lack of enough pages. (1pt) With local replacement, each process can only select from its own set of allocated frames. (1pt) It can alleviate the problem of thrashing, if one process starts thrashing, under local allocation it cannot steal frames from another process and will not cause the latter to thrash. (1pt)

4.5. What is TLB reach? Please list two strategies for increasing TLB reach. State the drawback if any. (4 points)

Answer:

The amount of memory that can be accessed from TLB (2 points)

(1) Increasing the number of entries in the TLB

Increasing the number of entries in the TLB is a costly strategy as the TLB consists of associative memory, which is both costly and power hungry.

(2) Increasing the page size. Increasing the page size (or providing multiple page sizes) allows system designers to significantly increase the TLB reach without changing the size of the TLB. But there might be internal fragmentation.

4.6. Consider the following page reference string:

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

Assuming demand paging with four frames, how many page faults would occur for the following replacement algorithms? Please illustrate each step (9 points)

a) FIFO replacement

b) LRU replacement

c) Optimal replacement

Answer:

a) FIFO

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

Page faults: 16

b) LRU

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

Page faults: 15.

c) Optimal

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

Page faults: 11

5. [**14 points**] File System and Implementation

5.1. Why do all file systems suffer from internal fragmentation (disk)? (2 points)

Answer:

In general, disk space is allocated in fixed sized blocks. Therefore, when a file is written to the disk, it may not fit exactly within an integer number of blocks so that a portion of a block will be wasted.

5.2. Please briefly describe the three memory allocation methods. (3 points)

Answer: (1pt each)

Contiguous allocation. Physical memory is usually divided into two partitions, which are high memory for users and low memory for resident OS. Physical address space of a process is a block of continuous Physical memory.

Paging allocation. Physical memory is divided into fix-sized frames and logical memory into fix-sized pages. Physical address space of a process is a collection of frames and can be non-continuous. The OS keeps the trace of all free frames and manages the page table in main memory.

Segmentation allocation. Physical memory is divided into variable-sized blocks and local memory identified as <segment no., offset> into variable-sized segments. Physical allocation to a process is a collection of segments. The OS keeps trace segment table in main memory

(Contiguous allocation, index allocation and linked allocation are also given full marks.)

5.3. 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 16 direct disk blocks, plus one single, one double, and one triple indirect disk block respectively.

a) What is the maximum size of a file that can be stored in this file system? (4 points)

b) Suppose one third of all files are exactly 6KB and the rest files are exactly 8KB, what fraction of disk space would be wasted? (Consider only blocks used to store data) (3 points)

c) If we change the block size to 4KB, will the fraction of wasted disk space in (b) be reduced? Please justify your answer. (2 points)

Answer:

a) (16 * 8 /KB/) + (2048 * 8 /KB) + (2048 * 2048 * 8 /KB/) + (2048 * 2048 * 2048 * 8 /KB) = 128K+16M+32G+64T

b) Both 6KB and 8KB files will use 8KB space. For each 8KB file, no space will be wasted. For each 6KB file, 2KB space will be wasted. Therefore, the fraction of waste will be 2/8*1/3+0/8*2/3=1/12=8.33%.

c) No. For each 6KB file, still 2KB space will be wasted. For each 8KB file, still no space will be wasted. Therefore, the fraction of waste will still be 8.33%.

6. [**13 points**] Secondary Storage

6.1. Suppose a hard disk drive has 600 cylinders numbered 0 to 599. The disk is currently serving a request at the cylinder 60 and the previous request was at the cylinder 40. The pending requests in the queue (in FIFO order) are at the cylinders:

70, 20, 30, 530, 330, 150, 450, 400, 570, 290

For each of the following disk scheduling algorithms:

a. FCFS

b. SSTF

c. SCAN

d. C-LOOK

a) Write out the sequence of disk head positions (in terms of cylinder numbers) in order to satisfy all the pending requests starting from the current head position. Calculate the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests. (8 points)

Answer:

FCFS: (60) -> 70 -> 20 -> 30 -> 530 -> 330 -> 150 -> 450 -> 400 -> 570 -> 290

Distance: 10 + 50 + 10 + 500 + 200 + 180 + 300 + 50 + 170 + 280 = 1760

SSTF: (60)-> 70 -> 30 -> 20 -> 150 -> 290 -> 330 -> 400 -> 450 -> 530 -> 570
Distance: 10 + (70 – 20) + (570 – 20) = 610

SCAN: (60)-> 70-> 150 -> 290 -> 330 -> 400 -> 450 -> 530 -> 570 -> (599) -> 30 -> 20
Distance: (599 – 60) + (599 – 20) = 1118

C-LOOK: (60)-> 70 -> 150 -> 290 -> 330 -> 400 -> 450 -> 530 -> 570 -> 20 -> 30
Distance: (570 –60) + (570 – 20) + (30-20) = 1070

b) For systems that place a heavy load on the disk, why SCAN usually perform better than SSTF? (1 points)

Answer: SCAN can perform better because it's will not encounter starvation problem.

6.2. List the two main goals of parallelism in a disk system, via data stripping. (Hint: consider small access and large access.) (2 points)
Answer: Increase the throughput of multiple small accesses by load balancing.
Reduce the response time of large access.

6.3. List and describe the two techniques used in RAID for data stripping. (2 points)
Answer: Bit-level striping: splitting the bits of each byte across multiple disks so that the data can be accessed from multiple disks in parallel.
Block-level striping: blocks of a file are striped across multiple disks.

7. [**6 points**] Monitor and Its Implementation

7.1. How does the signal() operation inside a monitor differ from the corresponding operation signal() defined for semaphores? (3 points)
Answer: The signal() operation associated with monitors is not persistent in the following sense: if a signal is performed and if there are no waiting threads, then the signal is simply ignored and the system does not remember that the signal took place (1 pt). If a subsequent wait operation is performed, then the corresponding thread simply blocks. (1 point) 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 (1 point)

7.2. Consider the implementation of x.wait() on the conditional variable x inside a monitor using semaphores below. Please explain how the process release the monitor and which process may obtain the monitor. (3 points)

```
x_count++;
if (next_count > 0)
   signal(next);
else
   signal(mutex);
wait(x_sem);
x_count--;
```

Answer: The process first checks to see if there are processes waiting to enter the monitor (next_count>0); if there is, the process wakes up one process (signal(next)), release the monitor and let it enter the monitor (2 points). If there is no process waiting, the process invoking x.wait() simply releases the monitor (signal(mutex)). (1 point)