



Final Examination

COIS 3320 – Fundamentals of Operating Systems

V2

Due By: March 16th – 6:00PM

Instructions:

- 1- Please submit your answers as a single PDF titled `firstname_lastname_final` on blackboard.
 - a. Make sure you include your name and student number at the very top of your submitted PDF.
 - b. The answers to the questions must be labelled and in order (e.g., answer Q1 first followed by Q2).
 - c. Your C program must be copied and pasted into your word processor. You do not need to submit the .C file.
 - d. Make sure you test your C program (input, expected output, actual output (screenshot), brief comment).
 - e. Your C program can be programmed in your favorite text editor or IDE, but it must run on Loki.
- 2- Your answers must be typed not handwritten and scanned. Handwritten finals will not be accepted.
 - a. Make sure you are showing your work. Even if the correct answer is written down without any explanation, I usually cannot award points for it.
- 3- Make sure you monitor the COIS 3320 MS Teams general chat.
 - a. All your questions must be asked on MS Teams, please do not email me for your own sake!
 - b. Any extensions will be announced on the MS Teams general chat.
- 4- This exam is open book. You can access all your notes and anything on the internet. However, you are not allowed to communicate with your classmates for help on the exam, and you are not allowed to communicate with any third party.
- 5- The exam is due by 6:00PM EST every minute you are late after that will constitute a 2-point penalty.
 - a. If you have an SAS accommodation letter, please ignore the 6:00PM deadline, your deadline is based on your extended time factor, which will be: 180 minutes multiplied by your extended time factor. Blackboard will say your exam is late, do not worry about that, I have a list of all of you!
- 6- Please do not post the final examination anywhere, it will be tracked!

1- Who is your favorite lecturer of all time? (2 points)

2- Page Reference Strings (4 points)

In a four-frame system where all frames are initially empty how many page faults would occur for the optimal, FIFO, and LRU algorithm; given the following page reference string:

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

Make sure you show all your work. A table would be optimal to show your work for this question!

3- Paging & Segmentation (8 points total, two points each)

Consider the following list of processes and their associated sizes.

Process	Size (Bytes)	# Of Segments	Segment Size(s)
P1	8111	5	2000, 3000, 2000, 1000, 111
P2	2100	2	1000, 1100
P3	2	2	2

a- Assuming that Paging is used, show the page table for each process under the following scenario: assume a page size of 2000 bytes and a main memory of size 32 frames with Frames 0, 2, 4, 5, 7, 10, 15, 16, 17, 21, 24 and 31 currently utilized. Also assume the free frames are ordered by number in the free-space list. Each Page Table entry should contain Page# and Frame#.

b- Assuming that Segmentation is used, show the Segment Table for each process using the **next fit** allocation algorithm under the following scenario: given a main memory size of 64,000 bytes with the following areas currently available. Assume a segment is added at the beginning of the free hole. Each segment table should contain the segment #, size, and the MM address.

Starting Location	Size (Bytes)
1000	5,000
7,000	12,000
20,000	4,000
30,000	3,500
35,000	29,000

- c- Repeat part b assuming worst fit allocation is used.
- d- Assuming that Paged-Segmentation is used, show the Segment and Pages Tables for each process using the same scenario for main memory as described in Part (a). The Segment Tables entries for this part should contain Segment#, Size, and Page Table id (start counting at 0).

4- Banker's Algorithm (6 points, 2 points each)

Given the following snapshot of a system:

- a- Is there an error in the max matrix? If yes, correct it.
- b- Is the system in a safe state (is there a sequence of processes which will not lead to a deadlock)? Show the safe sequence if yes, or show why it cannot run due to a deadlock if the answer is no.
- c- Can a request P3(5,3,2) be granted?

Allocation			
	A	B	C
P1	2	6	5
P2	8	4	3
P3	4	2	3

Max			
	A	B	C
P1	8	4	3
P2	8	6	4
P3	4	3	3

Available		
A	B	C
4	3	3

5- Page Fault Frequency Algorithm (4 points)

Physical memory is initially empty. The following reference string is processed:

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

Show which pages are resident under the page fault frequency algorithm with $d = 3$.
Indicate when page faults occur.

Using a table to show your work might be a good idea for this one!

Time t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
RS		0	1	4	0	2	0	0	1	0	3	0	4	0	3
Page 0															
Page 1															
Page 2															
Page 3															
Page 4															
Fault?															

6- Disk Scheduling Algorithms (8 points, 2 points each)

The r/w head of a disk is at track 143. The previous position was track 0. Requests to access the following tracks have arrived:

143, 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

In which order will the tracks be visited when using

a- FIFO

b- SSTF (Shortest Seek Time First)

c- Scan

d- C-Scan

- 7- Write a modular C program that will accept three command line arguments. The first argument is a symbol that indicates an operation, the second and third arguments are operands. Your program should be able to handle addition, subtraction, multiplication, and division (10 points). For example:

Input: ./final d 5 2

Output: 5 divided by 2 is 2.5

Input: ./final m 5 2

Output: 5 multiplied by 2 is 10

Input ./final a 5 2

Output: 5 + 2 is 7

Input ./final s 5 2

Output: 5 – 2 is 3

Tips & Hints:

Use a switch in your main function where the case is the first argument to decide which function to call.

You can assume the second and third arguments are valid, but you must check if a valid character was passed for the operation (e.g., d, a, s, or m).

I know I said you would have 2 C programming questions, but I had to cut one to make sure you had enough time.

Total Points: 42

Have a great Summer!! 😊