University of California, Berkeley
College of Engineering
Computer Science Division – EECS

Fall 2016 Anthony D. Joseph

Midterm Exam #2

October 25, 2016
CS162 Operating Systems

Your Name:	
SID AND 162 Login:	
TA Name:	
Discussion Section Time:	

General Information:
This is a **closed book and one 2-sided handwritten note** examination. You have 80 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points for that question. You should read **all** of the questions before starting the exam, as some of the questions are substantially more time consuming.

Write all of your answers directly on this paper. ***Make your answers as concise as possible.*** If there is something in a question that you believe is open to interpretation, then please ask us about it!

Good Luck!!

QUESTION	POINTS ASSIGNED	POINTS OBTAINED
1	27	
2	30	
3	25	
4	18	
TOTAL	100	

NAME: _____

1. (27 points total) True/False and Why?
   a. (12 points) True/False and Why? **CIRCLE YOUR ANSWER.**
      i) It is possible for a FCFS scheduler to achieve a lower average turnaround time (the time a process takes to complete after it arrives) than a Round Robin scheduler, even if you assume there is no context switching overhead.

      TRUE FALSE

      Why?

      ii) 8MB 2-way set-associative cache with a Most Recently Used (MRU) replacement policy will always have better or equal cache performance when compared to a 4MB direct-mapped cache.

      TRUE FALSE

      Why?

      iii) Suppose we have a multithreaded process that accesses (reads and writes) to an external database, and it avoids race conditions by having each thread acquire a lock global variable before accessing the database. Using this approach, we can run two instances of this process accessing the same database at the same time.

      TRUE FALSE

      Why?

iv) If the Banker's algorithm will not approve a resource request, and the resource request is processed, then system necessarily will enter deadlock.

TRUE FALSE

Why?

b. (15 points) Short answer.

i) (4 points) Briefly, in two to three sentences, explain how having a combination of I/O-bound processes and CPU-bound processes maximizes system utilization.

ii) (4 points) Briefly, in two to three sentences, explain how given a set of jobs and their arrival and execution times, you could quickly (O(n) time) determine if FCFS would produce a schedule that is optimal in terms of average completion time.

iii) (4 points) Briefly, in two to three sentences, explain when a hard real-time scheduling system, such as Earliest Deadline First, will **not** be able to meet all deadlines.

iv) (3 points) Early Intel processors such as the 8086 did not support dual-mode operation. Briefly, in two to three sentences, explain whether you can or cannot implement virtual memory and address spaces on such systems. If so, explain how. If not, explain why not.

NAME: _____

2. (30 points total) Deadlock.

a. (8 points) What are the four requirements for deadlock? Provide a brief one sentence explanation of each requirement.

#1

#2

#3

#4

NAME: _____

b. (6 points) Consider the following resource allocation policy:
Requests for and releases of resources are allowed at any time. If a request for resources cannot be satisfied because the resources are not available, then we check any processes that are blocked, waiting for resources. If they have the desired resources, then these resources are taken away from them and are given to the requesting process. The vector of resources for which the waiting process is waiting is increased to include the resources that were taken away. If the resources needed by a blocked process become available, the process is put back on the ready queue.

For example, consider a system with three resource types and the vector *Available* initialized to (4,2,2).

- If process P0 asks for (2,2,1), it gets them.
- If P1 asks for (1,0,1), it gets them.
- Then, if P0 asks for (0,0,1), it is blocked (resource not available).
- If P2 now asks for (2,0,0), it gets the available one (1,0,0) and one that was allocated to P0 (since P0 is blocked). P0's *Allocation* vector goes down to (1,2,1), and its *Need* vector goes up to (1,0,1).

ii) With this resource allocation policy, can deadlock occur? If so, give an example. If not, which necessary condition cannot occur?

iii) Can starvation occur?

NAME: _____

c. (7 points total) Consider a monitor implemented as follows:

```
condition p, q;    // p and q are initialized for you
lock mutex = FREE;

void stop(void) {
     ......
     p.wait(&mutex);
     ......
     q.signal();
}

void go(void); {
     ......
     p.signal();
     q.wait(&mutex);
}
```

Suppose two processes P1 and P2 use the monitor in the following way:

```
P1: while (TRUE) {
     ......
     stop();
     ......
   }

P2: while (TRUE) {
     ......
     go();
     ......
   }
```

i) (5 points) Show that deadlock can occur by listing an execution sequence ending in deadlock with explanation for each step. *Only one process can run at a time at each step –* ***you do not have to fill in all the rows below****.*

P1	**P2**	**Description**

ii) (2 points) Can starvation occur?

d. (9 points) There are 5 lawyers sitting at a round dinner table. Lawyers repeat (forever) the following three things in order: (1) think, (2) eat, and (3) sleep. Each lawyer has a bowl of rice in front of them and a chopstick between each plate. **A lawyer must have two chopsticks to be able to eat!** A lawyer can only acquire the chopstick to the left of their plate and the chopstick to the right of their plate; they cannot reach across the table to use other chopsticks. When they are done eating, they put down (release) the two chopsticks for someone else to use. Below is code that attempts to solve this problem. Each lawyer will call this method with the parameter (0 to 4) indicating the lawyer's seat number. **Unfortunately, this code can deadlock.**
Your task is to fix this code so it cannot get stuck and so that more than one lawyer can eat at the same time – ***your solution cannot use any other synchronization primitives.***

The constructor is called once in the main function, and then 5 lawyer threads are created, each one executing the `Go()` method. `Think()`, `Eat()`, and `Sleep()` are already written and you don't know how long they take to run.

```
Lawyers() {
  for (int i=0; i<=4; i++) chopstick[i] = new Semaphore(1);
}
void Go(int p) {
   while (1) {
     Think();



     chopstick[p].P();
     chopstick[(p + 1) % 5].P();




     Eat();
     chopstick[p].V();
     chopstick[(p + 1) % 5].V();
     Sleep();
     }
  }
}
```

NAME: _____

3. (25 points total) Memory Management.

a. (12 pts) Consider the following string of memory references in terms of requested page number: 1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.

Fill in the table below with using the FIFO, MIN and LRU (Least Recently Used) page replacement algorithms for memory with five frames by filling in the table with page numbers. The top row is the string of memory references, and each row contains the page numbers that reside in each physical frame F1 – F5 after each memory reference. For readability purposes, please only fill in the table entries that have changed and leave the unchanged entries blank. If several pages meet the replacement criteria, replace the one with the smallest frame number.

i) FIFO

	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5	4	2
F1	1																					
F2		2																				
F3			3																			
F4				4																		
F5					5																	

What is the number of page faults, including compulsory faults?

NAME: _____

ii) MIN

	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5	4	2
F 1	1																					
F 2		2																				
F 3			3																			
F 4				4																		
F 5					5																	

What is the number of page faults, including compulsory faults?

iii) LRU

	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5	4	2
F 1	1																					
F 2		2																				
F 3			3																			
F 4				4																		
F 5					5																	

What is the number of page faults, including compulsory faults?

NAME: _____

b. (4 points) Some operating systems provide a virtual copy operation:
`VirtualCopy(proc, addr1, addr2, len)`,
where data of length `len` starting at location `addr1` in the caller's address space is "copied" to process `proc`'s address space starting at location `addr2`. Assume that `addr1`, `addr2`, and `len` are multiples of the page size. Assuming a page-table-based virtual memory system, explain how we can efficiently implement `VirtualCopy` so as to minimize the amount of copying required. *Note that after `Virtual Copy` completes, the two processes will* ***not*** *share writes to the data.*

NAME: _____

c. (9 points total) Inverted Page Tables. Assume we have a simple, demand paging environment, with no segmentation. Processes P 1 and P 2 both have logical memory addresses in the range 0 . . . 99, inclusive. The page size is 5. The hash table portion of the structure uses a hash function which simply calculates the index by adding the numerical portion of the process identifier and the logical page number. Note that this is a really bad hash function because it requires a large hash table, but for our exercise, it is sufficient. The hash table and the inverted page table shown are below.

Hashtable

Index	Process ID / logical page	Frame ID
…	…	…
3		
4	P2 2	1
5	P1 3	4
…	…	…
18	P1 17	2
19	P1 18	3
…	…	…
30		

Inverted Page Table

Index	Logical address info (minimized for simplicity)
0	
1	P2:RW
2	P1:RW
3	P1:R
4	P1:RW

Arrows in the hash table indicate chaining of entries when a hash conflict has occurred. Empty entries in the IPT indicate the associated physical frames are not allocated.

NAME: _____

i) (6 points) Calculate the physical addresses for the following logical addresses where the number after the colon is the logical address (**not the logical page**). Assume the first page of a process is page number 0, and that a mechanism exists to find a free frame in memory. .

Show your work for these calculations for partial credit – what calculations and table lookups were necessary to determine the actual address in memory from the logical address.

(1) P1:17

(2) P1:92

(3) P1:111

ii) (3 points) When process P2 attempts to read logical address 97, what happens? Specifically, describe the changes in the data structures, and what the process perceives of these changes.

NAME: _____

4. (18 points total) Scheduling. Five processes A, B, C, D and E arrive in this order at the same time with the following CPU bursts.

Process	CPU Burst
A	7
B	2
C	3
D	6
E	4

Fill in the entries of the following table with turnaround times and total turnaround time for each indicated scheduling policy and each process. Ignore context switching overhead. Turnaround time is defined as the time a process takes to complete after it arrives.

Scheduling Policy	Turnaround Time					**Total** Turnaround Time
	A	B	C	D	E	
First Come First Served						
Shortest Job First						
Round-Robin (quantum=2)						

Scratch space: