Midterm Test

(85 points)

This exam has a total of 8 questions, spread over 7 pages, including this cover page.

Student Name: ....

Student ID:..

You are allocated a maximum amount of space to answer each question. (We have provided sufficient lines.) Adhere to those limitations when you formulate your answers. Do not use the backside of the pages; no additional pages are allowed.
Please make an effort to write in a readable fashion. We will skip over (and therefore not grade) non-readable portions.

"I have adhered to the Aggie Code of Honor."

1(11)
2(8)
3(5)
4(6)
5(9)
6(9)
7(23)
8(14)
T(85)

Signature:

……………………………………………………..

1. [11 pts: 1pts each] Match each term in the left column to the definition/description in the right column that fits best. Do this by filling in the void entries on the left:

( )	Privacy	(A)	System Call
( )	Throughput	(B)	Could result in starvation
( )	Exception	(C)	Contains threads waiting to execute on the CPU
( )	Context switching	(D)	Used to create a new process
( )	Round Robin Scheduling	(E)	Routine in the Kernel to process an Interrupt
( )	Wait queue	(F)	Data is available only to authorized users
( )	Ready Queue	(G)	Number of operations completed per unit of time
( )	Handler	(H)	Transfer of control to OS in response to an event
( )	Shortest Job First	(I)	Action performed by OS to remove a process from processor and replace it with another
( )	Trap	(J)	Contains processes that are held up due to an IO operation
( )	Fork	(K)	CPU allocation with time slices

2. [8 pts: 2 pts each] Circle the following statements TRUE or FALSE

T F The exec() system call creates a new process

T F First Come First Served (FCFS) scheduling algorithm performance can never equal the performance of Shortest Job First (SJF) for average response time

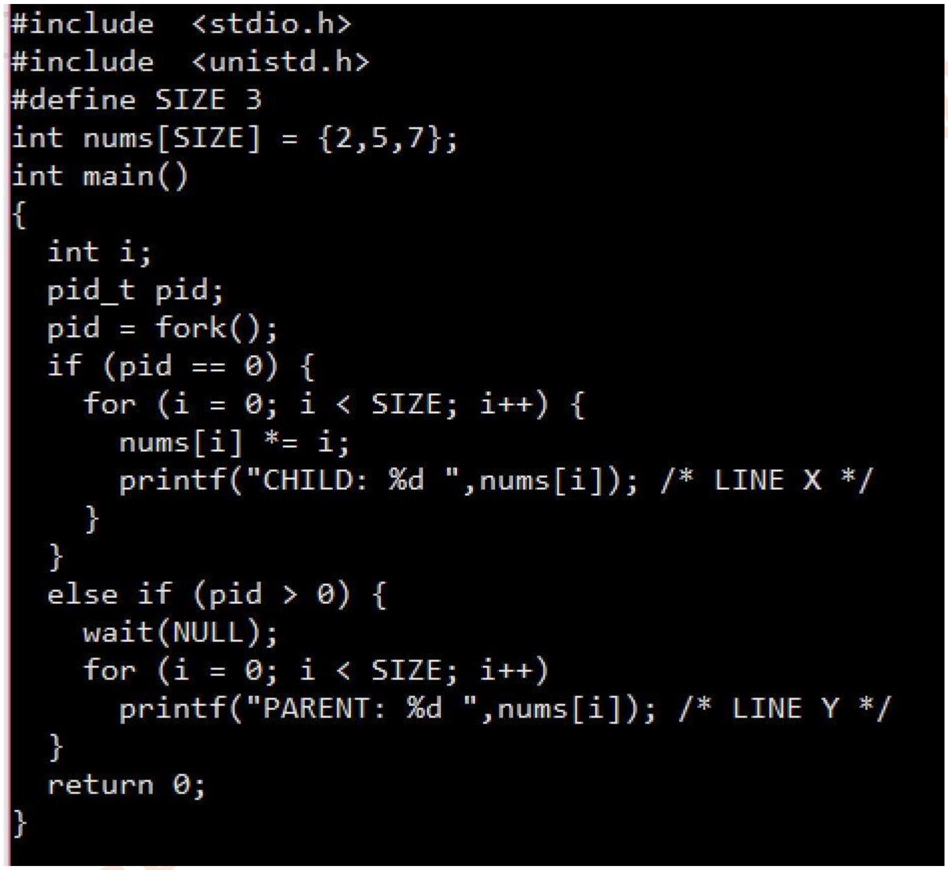
T F A race condition results when several threads try to access and modify the same data concurrently

T F Faults may allow return of control to the process that caused the exception

3. [5 pts: 1 pt each] Which of the following actions should be allowed only in kernel privileged mode? (circle correct answer)

(a) Masking of an Interrupt	YES NO
(b) Exception Handling	YES NO
(c) String search in an open file	YES NO
(d) Creation of a file	YES NO
(e) Add instruction operation	YES NO

4. [6 pts] In the following code, what will be the output at Lines X and Y? Explain!



```
#include  <stdio.h>
#include  <unistd.h>
#define SIZE 3
int nums[SIZE] = {2,5,7};
int main()
{
  int i;
  pid_t pid;
  pid = fork();
  if (pid == 0) {
    for (i = 0; i < SIZE; i++) {
      nums[i] *= i;
      printf("CHILD: %d ",nums[i]); /* LINE X */
    }
  }
  else if (pid > 0) {
    wait(NULL);
    for (i = 0; i < SIZE; i++)
      printf("PARENT: %d ",nums[i]); /* LINE Y */
  }
  return 0;
}
```

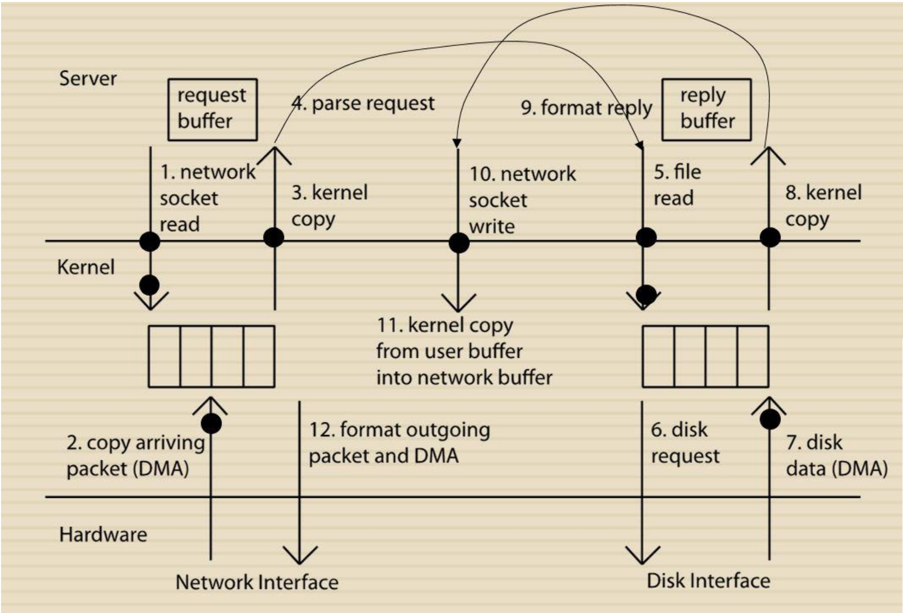
4a. [4 points] The output at Line (X) is ________________________________

Note: Space provided for explanation

4b. [2 points] The output at Line (Y) is __________________________________

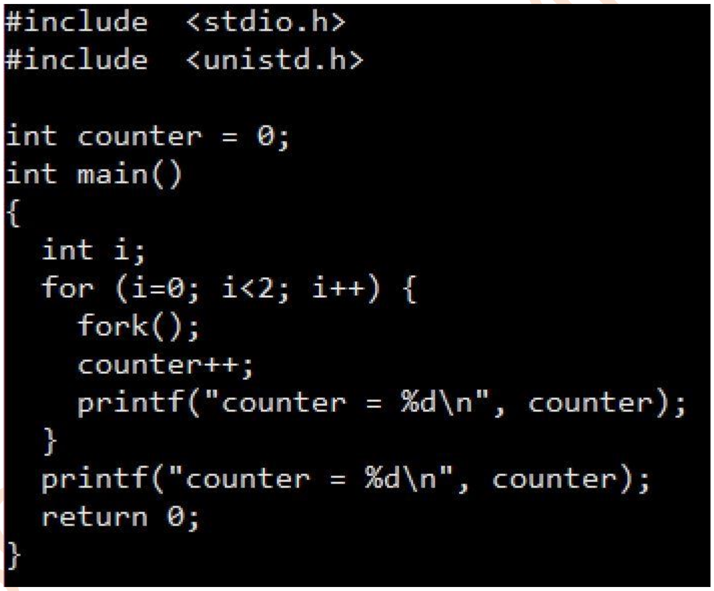
Note: Space provided for explanation

5. [9 pts, 1 pt for each correct label] Label the solid circles in the below webserver transaction. The 4 labels are listed on the right side of the pic. No explanation is needed, simply write the label next to the solid circles (you can use Arrow pointers to labels.



SYSCALL
WAIT
INTERRUPT
COPY

6. [9 pts: 3 pts for each part] Consider the program below:



```
#include  <stdio.h>
#include  <unistd.h>

int counter = 0;
int main()
{
  int i;
  for (i=0; i<2; i++) {
    fork();
    counter++;
    printf("counter = %d\n", counter);
  }
  printf("counter = %d\n", counter);
  return 0;
}
```

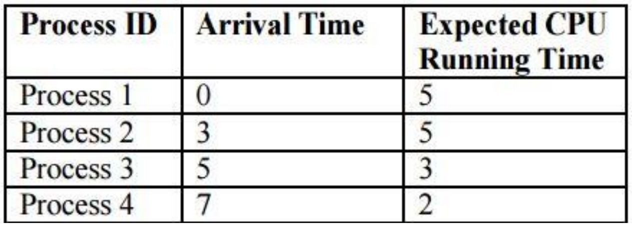
Please answer the questions below. No explanation is needed.

6a. How many times would the value of counter be printed: ______________________________

6b. What is the value of counter printed in the first line? ________________________________

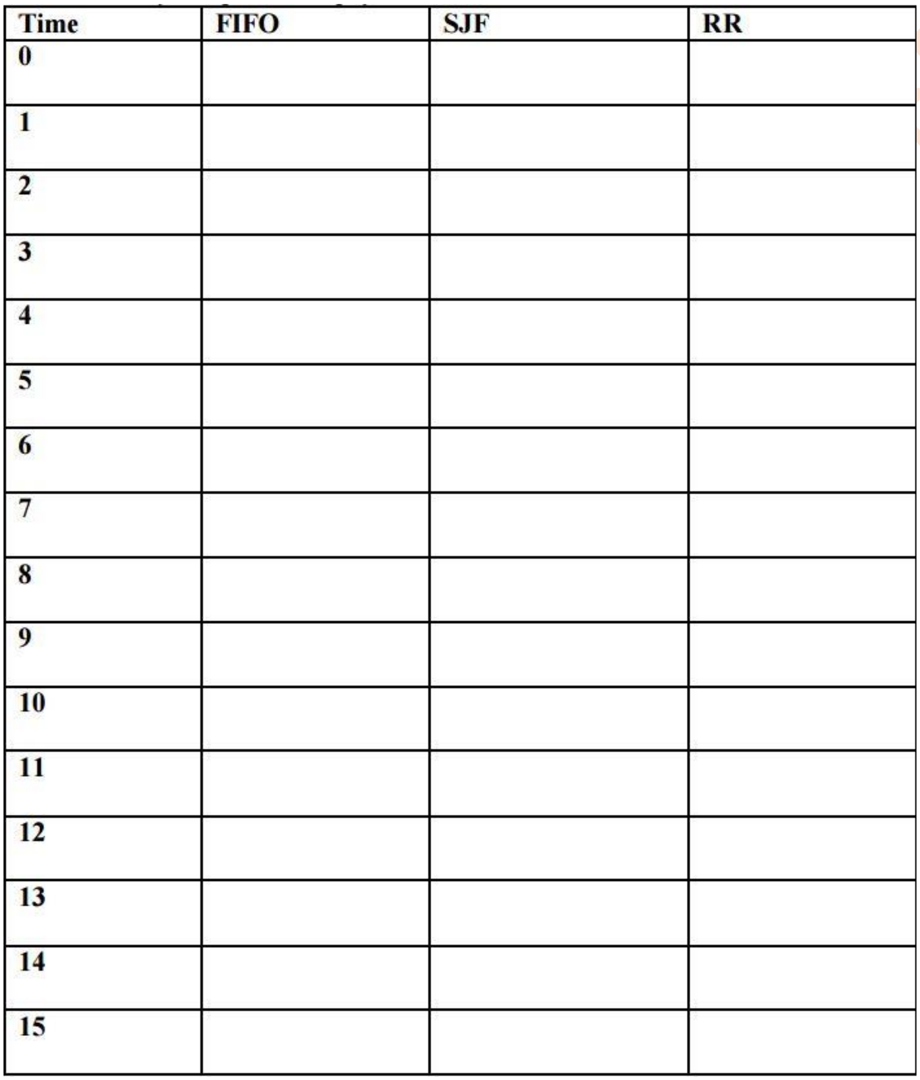
6c. What is the value of counter printed in the last line? _________________________________

7. [23 pts] Here is a table of processes and their associated arrival and running times



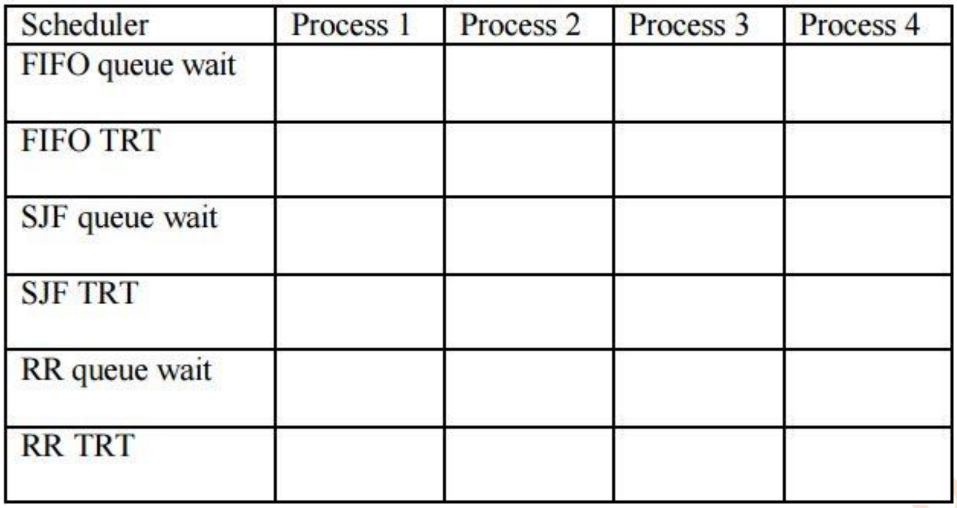
Process ID	Arrival Time	Expected CPU Running Time
Process 1	0	5
Process 2	3	5
Process 3	5	3
Process 4	7	2

7a. [9 pts] Show the scheduling order for these processes under First-In-First-Out (FIFO), Shortest-Job First (SJF), and Round-Robin (RR) with a quantum = 1 time unit. *Assume that the context switch overhead is 0 and new processes are added to the head of the queue except for FIFO.* You can use P1, P2, etc. notations for Process1 Process2 etc.



Time	FIFO	SJF	RR
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

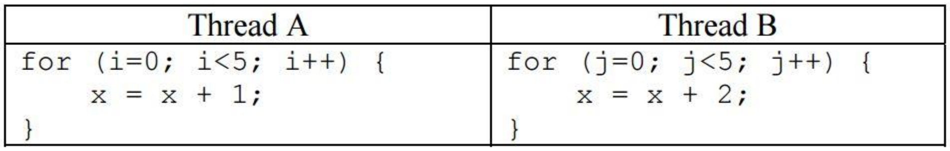
7b. [12 pts] For each process in each schedule above, indicate the queue wait time and turnaround time (TRT). *The queue wait time is the total time a thread spends in the wait queue. The turnaround time is defined as the time a process takes to complete after it arrives.*



Scheduler	Process 1	Process 2	Process 3	Process 4
FIFO queue wait				
FIFO TRT				
SJF queue wait				
SJF TRT				
RR queue wait				
RR TRT				

7c. [2 pts] Explain how to fool the multi-level feedback scheduler's heuristics into giving a long-running task more CPU cycles.

8. [14 pts]: Consider the following two threads of a process, to be run concurrently in a shared memory (all variables are shared between the two threads):



Thread A	Thread B

`for (i=0; i<5; i++) {`
`    x = x + 1;`
`}`
	`for (j=0; j<5; j++) {`
	`    x = x + 2;`
	`}`

Assume the following:

1. a single-core system
2. load and store are **atomic** (i.e. they start and finish without interruption)
3. x is initialized to 0 before either thread starts, and
4. x must be loaded into a register before being incremented (and stored back to memory afterwards).

The following questions consider the <u>final</u> value of x <u>after</u> both threads have completed.

8a. [4 pts]: State the upper and lower bounds of value of x when both threads have completed.

8b. [4 pts]: Give a concise proof why x≠1 when both threads have completed.

8c. [6 pts]: Suppose we replace 'x = x+2' in Thread B with an **atomic double increment** operation atomicIncr2(x) that **cannot be preempted** while being executed. What are **ALL possible final values** of x? Explain.