

Name (Please print legibly): _____

Net ID: _____

For Grader Use Only _____

Question	Q1 (5)	Q2 (5)	Q3(10)	Q4 (2)	Q5 (3)	Q6 (5)	Q7 (5)	Q8 (5)	Total/40
Mark									

Please work on this homework independently. Please answer in spaces provided.

Question 1 (5 x 1 point): Which of the following is correct use of pointers? (Mark correct or incorrect in the space provided)

a)
int *x, q=0;
*x = q; **Incorrect** (pointer x does not point to a valid memory location)

b)
int b[2];
*b = 20; **Correct**

c)
char greet[80];
strcpy ("Hello", greet); **Incorrect** ("Hello" is a constant and has no room for appends)

d)
int *p, q[2];
p = malloc (sizeof (int));
*p = 3;
q[2]=*p; **Incorrect** (array bound exceeded)

e)
int *x, y;
x = &y;
*x = 10; **Correct**

Question 2 (5 x 1 point): In (a) through (e) below, assume that all tasks **arrive together at time 0**. Consider the scheduling policies: (i) SJF, (ii) FIFO, (iii) Round-Robin with quantum set equal to 1/10 of the computation time of the shortest task.

a) Which scheduling policy minimizes average task waiting time? **SJF**

b) Which scheduling policy minimizes average task response time? **RR**

c) Which scheduling policy suffers the convoy effect? **FIFO**

d) Which scheduling policy has the largest context switch overhead? **RR**

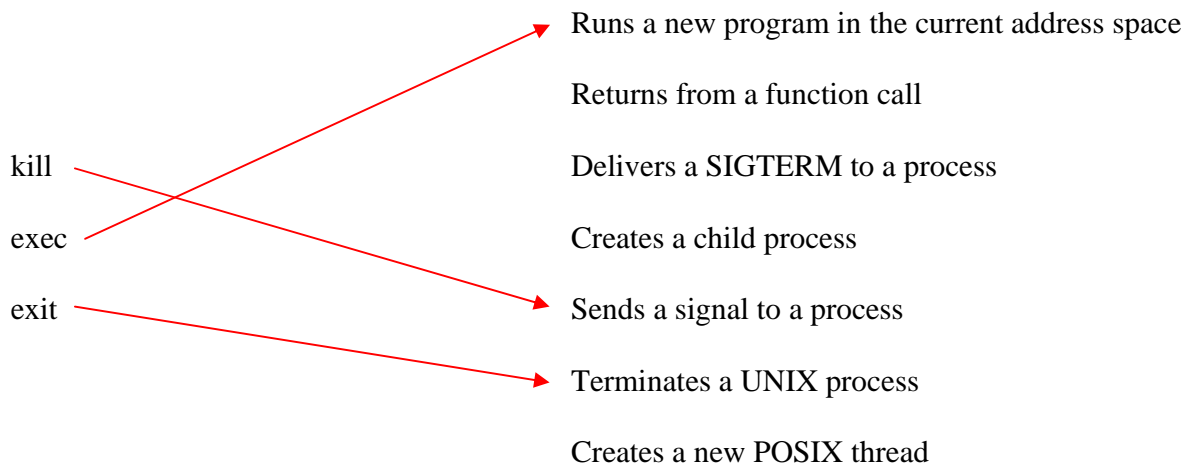
e) Which scheduling policy may suggest a starvation effect? **SJF**

Question 3 (2 x 1 point): Briefly, in TWO sentences or less, what is the difference between the following two cases?

- a) Function g() calls function f().
- b) Function g() creates a POSIX thread that executes function f().

In (a): control passes from function g() to function f() then returns to g() when f() ends.
In (b): when the new thread is created, f() executes concurrently with g().

Question 4(3 x 1 point): Connect each of three command names or calls on the left with the most appropriate definition on the right. If more than one definition applies, choose the best and most accurate one.



Question 5 (10 x 1point): The following cases show pseudo-code executed by two different threads, thread 1 and thread 2 around a critical section. (If code of thread 2 is not mentioned, it is the same as thread 1.) Fill-in the table below specifying, for each case, (i) whether mutual exclusion is guaranteed and (ii) whether progress is guaranteed. In all cases, assume that all variables are initialized to zero in the beginning. In answering the question you should consider all possible scenarios including scenarios where a thread executes more than once (sequentially) and scenarios where only one of the two threads remains, while the other has terminated.

	Case (a)	Case (b)	Case (c)	Case (d)	Case (e)
Mutual exclusion (Yes/No)	No	Yes	Yes	Yes	No
Progress (Yes/No)	Yes	No	Yes	No	No

	Thread 1	Thread 2
a)	<pre>while (x > 0) {}; x ++; /* You may assume this line is atomic */ execute critical section; x --; /* You may assume this line is atomic */</pre>	
b)	<pre>x2 = 1; while (x1 != 0) {}; execute critical section; x2 = 0;</pre>	<pre>x1 = 1; while (x2 != 0) {}; execute critical section; x1 = 0;</pre>
c)	<pre>while (testandset (x)) {}; /* testandset returns current value then sets it to 1 */ execute critical section; x = 0;</pre>	
d)	<pre>if (x is odd) { execute critical section; x=2; }</pre>	<pre>if (x is even) { execute critical section; x=1; }</pre>
e)	<pre>while (y == 1) {}; y = 1; execute critical section;</pre>	<pre>while (y == 0) {}; y = 0; execute critical section;</pre>

Question 6 (5 x 1 point): Mark True or False.

- a) You cannot join a detached thread TRUE
- b) In non-preemptive scheduling a process cannot go from ready to blocked state TRUE
- c) Unlocking a mutex can cause some process to go from blocked to ready state TRUE
- d) When a thread calls **pthreadexit**, the entire process terminates FALSE
- f) alarm(X) always terminates the process in X seconds FALSE

Question 7 (5 points): You want to change the default behavior of **SIGUSR1** to print a message on the screen. You also want to block **SIGINT** while **SIGUSR1** is handled. Your process should

continue to execute in the absence of signals. Which of the following calls you are likely to use? (Circle all that apply.) Each wrong choice knocks off one point.

sleep

alarm

sigwait

sigprocmask

sigaction

sigsuspend

Question 8 (5 x 1 point): Please complete the definition of each of the performance criteria below by filling-in the spaces. Only ONE word per space, please.

- a) The turnaround time minus the execution time is equal to the waiting time.
 - b) The turnaround time is the time from the process creation time to its finish time.
 - b) The total time a process spends in queues is called waiting time
 - d) Response time is the interval from arrival time to start time
-

Thank you and good luck!