

## CSSE 332 - OPERATING SYSTEMS

## Introduction to Multi-threading

Name: \_\_\_\_\_

**Question 1.** (5 points) What are the main differences between a *thread* and a *process*? In your answer, list out the pieces of memory and the processor that are shared between threads and those that are not.

**Question 2.** (5 points) Consider the following function that we'd like to run in its own separate thread:

```
1 void *thread_function(void *arg) {  
2     // argument contains an id given to the thread  
3     int my_id = *(int*)arg;  
4  
5     printf("Hello from thread %d\n", my_id);  
6     return NULL;  
7 }
```

In the box below, write a piece of code that would create and run this function in a new thread. Make sure to include the code that waits for the created thread to return.

**Question 3.** (5 points) Consider the following piece of code, how many threads would we end up with when it executes?

```
1 pthread_t threads[5];  
2 for(int i = 0; i < 5; i++) {  
3     pthread_create(&threads[i], 0, some_function, 0);  
4 }
```

**Question 4.** (5 points) Consider the following snippet of code:

```
1 void thread_fn(void *arg) {  
2     int *id = *(int*)arg;  
3     printf("Thread %d executing\n", id);  
4     return 0;  
5 }  
6  
7 // somewhere else  
8 pthread_t threads[5];  
9 for(int i = 0; i < 5; i++) {  
10     pthread_create(&threads[i], 0, thread_fn, &i);  
11 }
```

In the box below, write down for each thread, the output that will be printed on the screen. For example, write something like Thread 1: Thread 0 executing and so on.

**Question 5.** (5 points) Consider the following snippet of code:

```
1 void thread_fn(void *arg) {  
2     int id = *(int*)arg;  
3     printf("Thread %d is executing \n", id);  
4     return 0;  
5 }  
6  
7 // somewhere else  
8 pthread_t threads[5];  
9 int ids[5];  
10 for(int i = 0; i < 5; i++) {  
11     ids[i] = i+1;  
12     pthread_create(&threads[i], NULL, thread_fn, &ids[i]);  
13 }  
14  
15 for(int i = 0; i < 5; i++) {  
16     pthread_join(threads[i], 0);  
17 }  
18 printf("All the threads have returned\n");
```

In the box below, write the order in which the print outs will show up on the console when this code executes.