

Name: Sean Lossef

RCS ID: losses @rpi.edu

CSCI 4210 — Operating Systems ♠♠♠
Spring 2020 Quiz 5 (March 26, 2020)

- This is an open book, open notes quiz. Please do not use any other people as resources including your classmates.
- This quiz is designed to take 25 minutes; therefore, for 50% extra time accommodations, the expected time is 38 minutes and 100% extra time accommodations is 50 minutes (i.e., the end of class). Please self police and stay within these guidelines.
- Questions will not be answered except when there is a glaring mistake or ambiguity in a question. Please do your best to interpret and answer each question.
- Below is an honor code pledge for this course. By submitting this quiz for grading, you are asserting that you agree with and will abide by this pledge.

Honor Pledge: On my honor, I have neither given nor received any unauthorized aid on this quiz.

1. (4 POINTS) What is the most likely terminal output of the code below? Note that there are no compilation warnings or errors in the given code. The `#include` directives are omitted to save space on the page. Assume that all system calls succeed. Circle the best answer.

```
void * go( void * arg )
{
    char * s = calloc( strlen( arg ) + 1, sizeof( char ) );
    strcpy( s, arg );
    s[7] = '\\0';
    strcpy((char*)arg,"aaaa");
    fprintf( stderr, "%s", s );
    free( s );
    return NULL;
}

int main()
{
    pthread_t tid;
    char q[] = "ABCDEFGHJKLMNOPQRSTUVWXYZ";
    pthread_create( &tid, NULL, go, q );
    pthread_join(tid, NULL);
    fprintf( stderr, "%s", q );
    return EXIT_SUCCESS;
}
```

- (a) No output
(b) aaaaABCDEFGG
(c) ABCDEFGaaaa
(d) ABCDEFGHJKLMNOPQRSTUVWXYZABCDEFGG

2. (4 POINTS) When you call `pthread_mutex_unlock()`, what happens? Circle the best answer.

- (a) The thread creates a shared memory segment
- (b) The thread blocks waiting for the lock to release
- (c) The thread's parent is no longer required to call `pthread_join()`
- (d) The thread memory is copied from the parent thread to a child thread
- ☒ (e) Access to the critical section is granted to the next thread

3. (12 POINTS) What is the **exact** terminal output for the code below? If multiple outputs are possible, please use a diagram to show all possibilities. Note that there are no compilation warnings or errors in the given code. The `#include` directives are omitted to save space on the page. Assume that all system calls succeed. Also assume that child thread IDs are assigned sequentially starting at 777.

```
void * sonny( void * arg )
{
    int * s = (int *)arg;
    printf( "%ld lucky %d\n", pthread_self(), *s );
    return NULL;
}

int main()
{
    pthread_t tid1, tid2;
    int x = 13;
    int rc = pthread_create( &tid1, NULL, sonny, &x );
    x = 7;
    rc = pthread_join( tid1, NULL );
    rc = pthread_create( &tid2, NULL, sonny, &x );
    printf( "%d unlucky %d\n", rc, x );
    rc = pthread_join( tid2, NULL );
    return EXIT_SUCCESS;
}
```

