Midterm Review
CMSC 242
Spring 2017

Name: ______________________________

1. List at least two *syntactic* differences between C and C++.

2. Matching: For each of the following system calls in the left column, write the letter corresponding to the description of that call in the right column.

malloc ______ a. Causes a process to sleep until one of its children exits or sleeps

fork ______ b. Returns heap memory to the operating system

setuid ______ c. Reserves memory on the stack

exec ______ d. Reserves memory on the heap

free ______ e. Returns the unique id number of the current process

kill ______ f. Loads a program into memory and schedules it forexecution

wait ______ g. Terminates a process and sets its exit status

alloca ______ h. Creates a new process which is a copy of the current process

exit ______ j. Sends a signal to a process

getpid ______ k. Changes which user "owns" the current process

(Note that there is no option "i")

3. Suppose we declare an array of characters to hold a string named "id". Ho**w large** should we make the array if we plan to execute the following instructions?

a. strcpy(id, "Doodle"); //Strcpy copies the second string into the first string

b. scanf("%20s", id);

4. What is the difference between the following two blocks of code? Be detailed and specific.

Block 1

```
int num;
scanf(“%d”, &num);
printf(“%o”, num);
```

Block 2

```
int num;
scanf(“%d”, &num);
printf(“%d”, num);
```

The difference between Block 1 and Block 2 is:

5. Write code for allocating ten integers on the heap. Name the array "nums".

6. What is an environment variable? Give an example of an environment variable which includes both its name and how it is used.

7. What are two different ways a programmer can access the environment variables?

8. What does the UNIX "ulimit" command do?

9. What is the difference between pressing CTRL-Z and CTRL-C while running a program?

10. Give an example of a signal we can send to a process. Give both its name and explain the circumstances under which it would be used.

11. In what ways do the functions “execl” and “execvpe” differ from each other? Give at least **three**.

12. What is the difference between static linking and dynamic linking?

13. What is the difference between a strong symbol and a weak symbol? Why does it matter?

14. Show how I can declare a global variable "num_values" that cannot be seen from any other file linked into my project.

15. Show how I can tell the compiler to use a global floating point variable named "score" which is defined in another .cpp file of my project.

16. Below is a function named "times". Is “times” a strong symbol or a weak symbol?

```
int times(int num, char* str) {
        for (int i=0; i < num; ++i) {
                printf("%s\n", str);
        }
}
```

17. Suppose I have a file named "secret". What command could I use to set permissions on this file so that no one by the owner of the file can read it?

18. Suppose I want to use the system library named "libzip". What flag do I need to add to my compiler so that I can link in the system libraries?

19. What does the “whoami” command do?

20. Using dynamic linking, I can load libraries at runtime. Why would I want to do this? Give an example of a program that might use runtime dynamic loading and tell what advantage it gets from loading the library "on-the-fly".