CMSC 216 Process Worksheet I Spring 2020

1. What is the kernel?
2. Provide two alternatives that allow us to interact with the kernel.
3. What is the difference between a thread and a process? Draw a diagram that illustrates two threads in a process (similar to what we did in lecture).
4. What is context switching?
5. Which process has pid #1?
6. Give two examples of signals we have seen in class.
7. Why can fork() fail?
8. How many processes are created (not including the process associated with main) by the following code fragment? You can assume fork calls will always be successful.

```
int main() {
   fork();
   fork();

   return 0;
}
```

9. Implement the process function below. The function creates a child process that computes and prints the square of the child_value parameter. The parent process performs the same computation, but using the parent_value parameter. For example, running the program with input values 4 and 8 will generate the following results:

```
4 8
Child: 64
Parent: 16
Processing Done
```

- You may not modify the main function.
- The child will print the square value by using the message "Child: " followed by the result.
- The parent will print the square value by using the message "Parent: " followed by the result.
- We don't care about the order in which the output appears. For example, "Processing Done" can appear before the child output.

```
int main() {
   int parent_value, child_value;

   scanf("%d%d", &parent_value, &child_value);
   process(parent_value, child_value);
   printf("Processing Done\n");

   return 0;
}

void process(int parent_value, int child_value) {
```