Homework 1 for CS153 (Fall 2019)

Due: Sunday 10/20

* Be brief. You will be graded for correctness, not on the length of your answers.

* Typed electronic submissions are expected. If you submit handwritten answers, you are responsible for legibility.

I. Consider a single CPU system with an active process A. Explain what happens in the following circumstances including any interrupts, system calls, etc. and how they are handled until a process is back to running again (4.5 points)

a) The user presses a key to type in a character

b) The process executes continuously until it exhausts its scheduler allocated time slice

c) The process performs a read operation on an opened file

II. Which of the following involves an event? Explain briefly. (4 points)

a) Compressing some data to get ready to write it to a file (do not consider the file write)
b) The process in (a) writes the data to the file
c) A process experiences a divide by zero
d) A network packet is received

III. Consider the following program:

```
int count = 0;  // shared variable since its global

void twiddledee() {
  int i = 0;  // for part b this will be global and shared
  for (i = 0; i<2; i++)  {
    count = count * count;  // assume count read from memory once
  }
}
void twiddledum()  {
  int i=0; // for part b, this will be global and shared
  for (i=0; i<2; i++) { count = count - 1;}
}
void main() {
  thread_fork(twiddledee);
  thread_fork(twiddledum);
  print count;
}
```

a) What are all the values that could be printed in main? (5 points)

b) Repeat part 1 considering that i is also a shared variable (Bonus 3 points – tricky!)

c) Describe a potential schedule of execution that will result in the value printed out being equal to 0. Assuming there is only one CPU core, clearly specify when the transitions between the Ready and Running states occurs for each thread in this execution. (3 points)

IV. Consider the following program:

```
1    int main() {
2      int count = 1;
3      int pid = 0, pid2 = 0;
4      if ( (pid = fork()) ) {
5        count = count + 2;
6        printf("%d ", count);
7      }
8      if (count == 1)
9      {
10       count++;
11       pid2=fork();
12       printf("%d  ",  count);
13     }
14     if (pid2) {
15       waitpid(pid2, NULL, 0);
16       count = count * 2;
17       printf("%d ", count);
18     }
19   }
```

Note that on line 4, (pid = fork()) in the same line executes the fork, assigns the return value to pid and uses that return value to evaluate the condition. If the condition is above zero, the if is taken, if it is zero, it is not. This is a common but not friendly C hack.

a) How many processes are created during the execution of this program Explain. (2 points)

b) List all the possible outputs of the program. (4 points)

c) If we delete line 15 (the waitpid) show one output that is possible in the new program that is not possible in the old program. (2.5 points)

V. What is virtualization in the context of time-sharing and space-sharing? (2 point)