Roll Num: Name:

CS3510: Operating Systems I

Quiz 1: Autumn 2020

Instructions for submission:

1. You must submit your final answer copy as pdf.
2. You should avoid submitting scan copies of hand-written notes. Only if you wish to attach any figure, you can attach the scans of the figures in your pdf.

1. System calls are executed in the *kernel-mode* (supervisor or privileged mode) by the Operating System to satisfy some request on behalf of the normal user. In this case, how does the OS ensure safety of the system when it is executing a system call in the privileged mode on the behalf of the user. As you might recall normally all the user programs run in *user-mode* for greater security? **(4 pts)**

2. Consider an SMP system similar to the one shown in Figure 1.8 (page 17 of the 10th edition) of Galvin book. Illustrate with an example how data residing in memory could in fact have a different value in each of the local caches. **(4 pts)**

3. Explain the circumstances under which the line of code marked printf("LINE J") shown in the Figure below will be reached (Qn 3.33 of the book, 10th Edition). **(4 pts)**

```
#include < sys/types.h >
#include < stdio.h >
#include < unistd.h >
int main()
{
        pid t pid;
        /* fork a child process */
        pid = fork();
        if (pid < 0) { /* error occurred */
                fprintf(stderr, "Fork Failed");
                return 1;
        }
        else if (pid == 0) { /* child process */
                execlp("/bin/ls","ls",NULL);
                printf("LINE J");
        }
        else { /* parent process */
                /* parent will wait for the child to complete */
                wait(NULL);
                printf("Child Complete");
        }
        return 0;
}
```

4. Using the program shown in Figure below, explain what the output will be at lines X and Y.

Please justify your answer. **(5 pts)**

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

#define SIZE 5

int nums[SIZE] = { 0,1,2,3,4 } ;

int main()
{
        int i;
        pid t pid;

        // Initialize the values
        for (i = 0; i < SIZE; i++) {
                nums[i] = 1;
        }

        pid = fork();
        if (pid == 0) {
                for (i = 0; i < SIZE; i++) {
                        nums[i] += -i;
                        printf("CHILD: %d %d", i, nums[i]); /* LINE X */
                }
        }
        else if (pid > 0) {
                wait(NULL);
                for (i = 0; i < SIZE; i++)
                        nums[i] *= -i;
                        printf("PARENT: %d %d", i,nums[i]); /* LINE Y */
                }
        }
        return 0;
}
```

5. What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level.
a. Synchronous and asynchronous communication
b. Fixed-sized and variable-sized messages
Please justify your answers. **(6 pts)**