Name : Ramzy Izza Wardhana
NIM : 21/472698/PA/20322
Class : IUP CS B

Activity 7.1 Lab Computer Systems & Networking

Activity 7.1

Change the line that calls "pthread_join" in the "simple_thread.c" into a comment, compile, and run the program.

- What is the output of the program now?
- What happens?

Step 1: Change the pthread_join into comment

```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
void *thread_function(void *arg);
int i,j;
int main() {
        pthread_t a_thread;
        pthread_create(&a_thread, NULL, thread_function, NULL);
        //pthread_join(a_thread, NULL);
        printf("Inside Main Program\n");
        for(j=4;j>=0;j--)
        {
                printf("%d\n",j);
                sleep(1);
        }
}

void *thread_function(void *arg) {
        printf("Inside Thread\n");
        for(i=0;i<5;i++)
        {
                printf("%d\n",i);
                sleep(1);
        }
```

Step 2: Compile the code

```
root@--:~# nano simple_thread.c
root@--:~# gcc -o simple_thread.out simple_thread.c
root@--:~#
```

Step 3: Run the Program and output

```
root@--:~# ./simple_thread.out
Inside Main Program
Inside Thread
0
4
3
1
2
2
1
3
0
4
```

The program will run concurrently in which the function scheduled to run sequentially. Therefore, the result shows that the main function doesn't need to wait for thread function and will be run simultaneously or in other words, in parallel since we remove the pthread_join (commented) that previously caused main function to wait.