COSC1112/1114: Operating Systems Principles

Lab 5 (week 6)

1. Run the following two C programs and observe the concurrency processing of two functions when using thread, and the sequential processing of the same two functions when not using thread.

Code-1 (using thread):

```
#include <pthread.h>
#include <stdio.h>
#include <time.h>

int sum; /* this data is shared by the thread(s) */
int multiply;
void *runner(void *param); /* threads call this function */
void *second(void *param);

int main(int argc, char *argv[])
{
        pthread_t tid; /* the thread identifier */
        pthread_t tid1;
        pthread_attr_t attr; /* set of thread attributes */
        pthread_attr_t attr1;

        if (argc != 2) {
                fprintf(stderr,"usage: a.out <integer value>\n");
                return -1;
        }

        if (atoi(argv[1]) < 0) {
                fprintf(stderr,"%d must be >= 0\n",atoi(argv[1]));
                return -1;
        }

        /* get the default attributes */
        pthread_attr_init(&attr);
        pthread_attr_init(&attr1);
        /* create the thread */
        pthread_create(&tid,&attr,runner,argv[1]);
        pthread_create(&tid1,&attr1,second,argv[1]);
        /* wait for the thread to exit */
```

```
        pthread_join(tid,NULL);
        pthread_join(tid1,NULL);

        printf("final sum = %d\n",sum);
        printf("final multiply=%d\n",multiply);
}

/* The thread will begin control in this function */
void *runner(void *param)
{
        int i, upper = atoi(param);
        sum = 0;
        for (i = 1; i <= upper; i++){
                sum += i;
                sleep(1);
                printf("sum = %d\n",sum);
        }
        pthread_exit(0);
}

void *second(void *param)
{
        int i, upper = atoi(param);
        multiply = 1;
        for(i=1; i <= upper; i++){
                multiply += i*i;
                sleep(1);
                printf("multiply = %d\n",multiply);
        }
        pthread_exit(0);
}
```

Code-2 (no thread)

```
#include <pthread.h>
#include <stdio.h>
#include <time.h>

int sum; /* this data is shared by the thread(s) */
int multiply;
void runner(void *param); /* threads call this function */
void second(void *param);
```

```
int main(int argc, char *argv[])
{
        pthread_t tid; /* the thread identifier */
        pthread_t tid1;
        pthread_attr_t attr; /* set of thread attributes */
        pthread_attr_t attr1;

        if (argc != 2) {
                fprintf(stderr,"usage: a.out <integer value>\n");
                return -1;
        }

        if (atoi(argv[1]) < 0) {
                fprintf(stderr,"%d must be >= 0\n",atoi(argv[1]));
                return -1;
        }

        runner(argv[1]);

        second(argv[1]);

  printf("final sum = %d\n",sum);
        printf("final multiply=%d\n",multiply);
}

/* The thread will begin control in this function */
void runner(void *param)
{
        int i, upper = atoi(param);
        sum = 0;
        for (i = 1; i <= upper; i++){
                sum += i;
                sleep(1);
                printf("runner sum = %d\n",sum);
        }
        //pthread_exit(0);
}

void second(void *param)
{
        int i, upper = atoi(param);
        multiply = 1;
        for(i=1; i <= upper; i++){
                multiply += i*i;
```

```
                sleep(1);
                printf("second multiply = %d\n",multiply);
        }
        //pthread_exit(0);
}
```