Multithreading Sorting

Submitted by:

Ailen Grace T. Aspe
BSEC-4
2013-1364

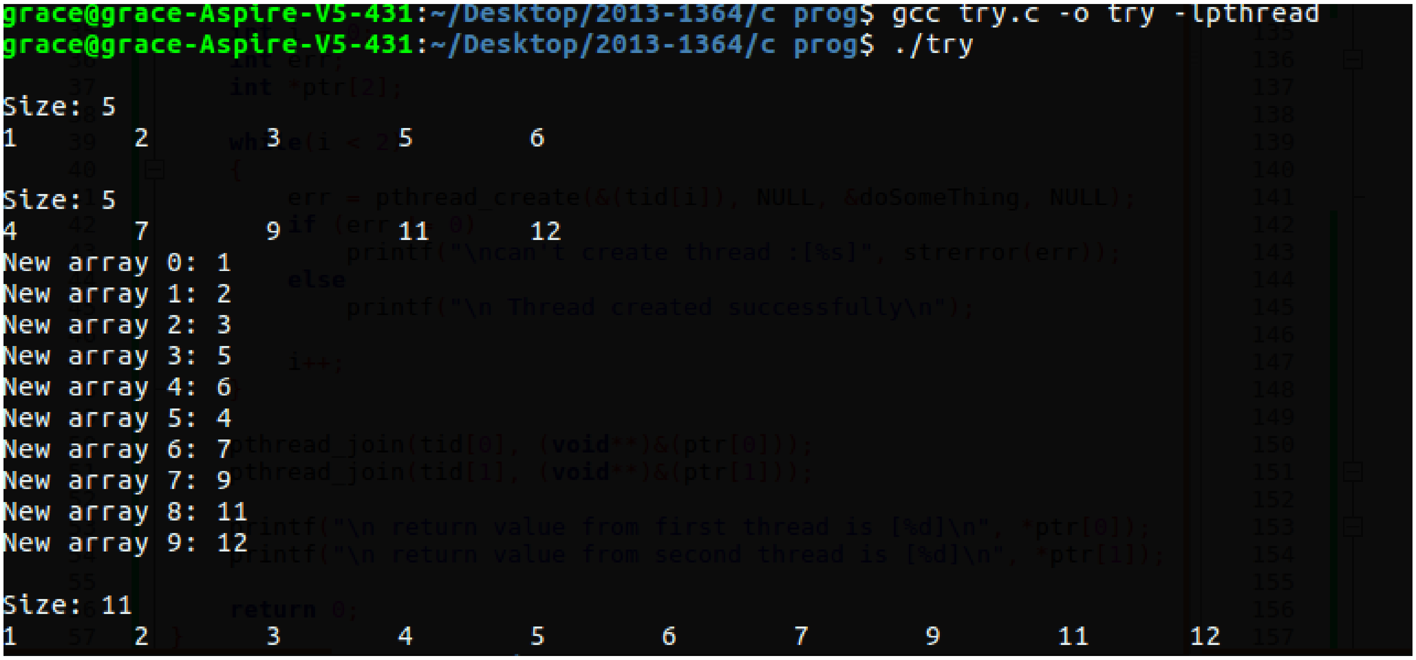
Submitted to:

Ms. Margie S. Arda
CSC 155 Instructor

SOURCE CODE

```
/*
    Author: Ailen Grace Aspe
    ID: 2013-1364
*/

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

int count;
int array1[20], array2[20];
void *mythread1(void *arg){
     int *tmp=(int*)arg;
     //int arr1[20];
     int i=0;
     int j, temp;



     while(tmp[i] !=0){

        i++;
     }
     printf("\nSize: %d\n", i);

     for(int k=0; k<i; k++){
        //printf("%d\n", arr1[k]);
        for(int z=k+1; z<i; z++){
            if(tmp[k] > tmp[z]){
                temp = tmp[k];
                tmp[k]=tmp[z];
                tmp[z]=temp;

            }

        }

     }
     for(int y=0; y<i; y++){
         array1[y]=tmp[y];
         printf("%d\t", tmp[y]);
     }
     printf("\n");




}

void *mythread2(void *arg){
     int *tmp=(int*)arg;
     //int arr1[20];
     int i=0;
     int j, temp;
     while(tmp[i] !=0){

        i++;
     }
     printf("\nSize: %d\n", i);

     for(int k=0; k<i; k++){
        //printf("%d\n", arr1[k]);
        for(int z=k+1; z<i; z++){
            if(tmp[k] > tmp[z]){
                temp = tmp[k];
                tmp[k]=tmp[z];
                tmp[z]=temp;

            }

        }

     }
     for(int y=0; y<i; y++){
         array2[y]=tmp[y];
         printf("%d\t", tmp[y]);
     }
     printf("\n");


}
void *mythread3(void *arg){
     int *tmp=(int*)arg;
     int i=0;
     int j, temp;

     while(tmp[i] !=0){

        i++;
     }
     printf("\nSize: %d\n", i);

     for(int k=0; k<i-1; k++){
        //printf("%d\n", arr1[k]);
        for(int z=k+1; z<i-1; z++){
            if(tmp[k] > tmp[z]){
                temp = tmp[k];
                tmp[k]=tmp[z];
                tmp[z]=temp;

            }

        }

     }
     for(int y=0; y<i-1; y++){
         array2[y]=tmp[y];
         printf("%d\t", tmp[y]);
     }
     printf("\n");

}

int main(){



    int array[] ={5, 1, 3, 2, 6, 7, 11, 4,9,12, 0};
    int newarray[20];
    int size = (sizeof(array)/sizeof(int))-1;
    int arr1[20], arr2[20];
    int lim =size/2;
    int j =0;

    for(int i=0; i<size/2; i++){
        arr1[i] =array[i];
        arr2[i]=array[lim];
        lim++;

    }

    pthread_t thread1, thread2, thread;

    pthread_create(&thread1, NULL, mythread1,(void*) arr1);
    pthread_create(&thread2, NULL, mythread2,(void*) arr2);

    pthread_join(thread2, NULL);
    pthread_join(thread1, NULL);

    for(int i=0; i<size; i++){

        if(array1[i]==0){
            newarray[i]=array2[j];
            j++;
        }
        else{
            newarray[i]=array1[i];
        }

    }

    for(int i=0; i<size; i++){
        printf("New array %d: %d\n",i, newarray[i]);
    }
    pthread_create(&thread, NULL, mythread3,(void*) newarray);
    pthread_join(thread, NULL);

    pthread_exit(NULL);




}
```

Wireframe



```
grace@grace-Aspire-V5-431:~/Desktop/2013-1364/c prog$ gcc try.c -o try -lpthread
grace@grace-Aspire-V5-431:~/Desktop/2013-1364/c prog$ ./try

Size: 5
1       2       3       5       6

Size: 5
4       7       9       11      12
New array 0: 1
New array 1: 2
New array 2: 3
New array 3: 5
New array 4: 6
New array 5: 4
New array 6: 7
New array 7: 9
New array 8: 11
New array 9: 12

Size: 11
1       2       3       4       5       6       7       9       11      12
```