

BIM306 Operating Systems

Project-I

1MRandomNumber

Due: March 18, 2017

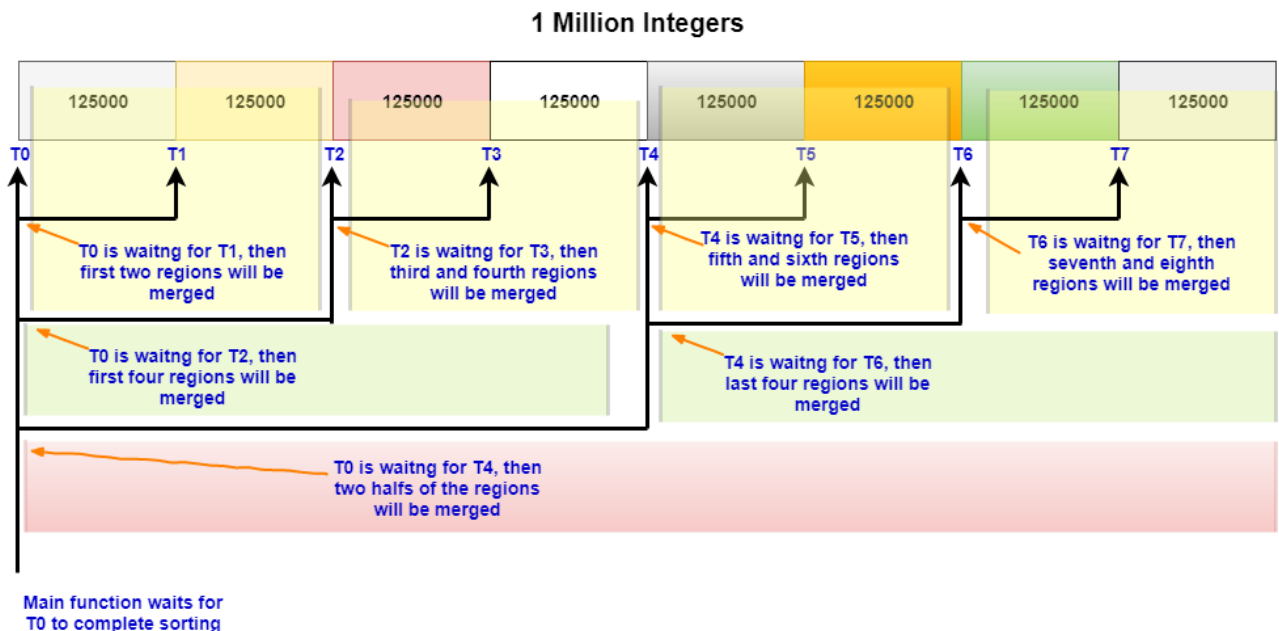
Note: The groups should consist of 2 students exactly.

Project Definition

You are asked to implement an integer-sorting program using threads in C programming languages. The program have several responsibilities you have to implement. First, the program should create a binary file including random generated 1 Million integers. Then, you may load these integers to the buffer (array). The focus in the project is that the numbers have to be sorted with using threads. Eight threads, you have created, have to deal with their own piece of array to sort each pieces. After sorting work is done, you need to merge all pieces to construct completely sorted array like. While doing your project, you need to use POSIX library. Your job is illustrated as in Figure below. Last, you need to write your sorted array to the binary file as in creating random number file.

File name for random numbers: "1MRandomNumber.dat"

File name for sorted numbers: "1MRandomNumberSorted.dat"



Hints:

- The functions may be an idea for your project implementation. You may change, remove, or add any function.

```
void *runner(void *params); /* the thread */  
void create1MBinaryFile(); /* 1M random file being sorted */  
void load1MBinaryFile(); /* 1M random file being sorted */  
int control(); /* Controls sorted file */  
int compare (const void * a, const void * b);  
void create1MBinaryFileSorted();  
void merge(int *parameter1,int *parameter2,int N);
```

- **gcc -o 1MRandomNumber 1MRandomNumber.c -pthread**

- Sample output. You can give more information than the output

1000000 Numbers is written

1000000 numbers is read

Thread 1 is created

Argument = 0 offset = 0 length = 125000

Thread 2 is created

Thread 3 is created

Thread 4 is created

Thread 5 is created

Thread 6 is created

Thread 7 is created

Thread 8 is created

Argument = 1 offset = 125000 length = 125000

Argument = 2 offset = 250000 length = 125000

Argument = 3 offset = 375000 length = 125000

Argument = 4 offset = 500000 length = 125000

Argument = 5 offset = 625000 length = 125000

Argument = 6 offset = 750000 length = 125000

Argument = 7 offset = 875000 length = 125000

Thread 1 is joined

Thread 5 is joined

Thread 3 is joined

Thread 2 is joined

Thread 7 is joined

Thread 6 is joined

Thread 4 is joined

Last Thread (tid 0) has joined

1000000 Sorted Numbers is written

You have sorted number successfully. To make sure, run it several times!

- You can use *qsort* to sort each 1/8 array blocks. **Do not use** for whole array and sorting two sorted blocks.
- This is my control function. You can implement your own function.

```
int control(){
    FILE *fp = fopen("1MRandomNumberSorted.dat","rb");
    int i,next=1;

    if(fp==NULL){
        printf("File couldn't be opened!\n");
        return -1;
    }

    for(i=0;i<NUMBER-1;i++,next++){
        fread(&i,sizeof(int),1,fp);
        fread(&next,sizeof(int),1,fp);
        if(i>next){
            fclose(fp);
            return -1;
        }
    }
    fclose(fp);
    return 1;
}
```