Project 3: Multithreaded Sorting Application & Fork-Join Sorting Application

Name: 韩冰

Number: 516030910523

1.Multithreaded Sorting Application

Write a multithreaded sorting program that works as follows: A list of integers is divided into two smaller lists of equal size. Two separate threads (which we will term sorting threads) sort each sublist using a sorting algorithm of your choice. The two sublists are then merged by a third thread—a merging thread—which merges the two sublists into a single sorted list.

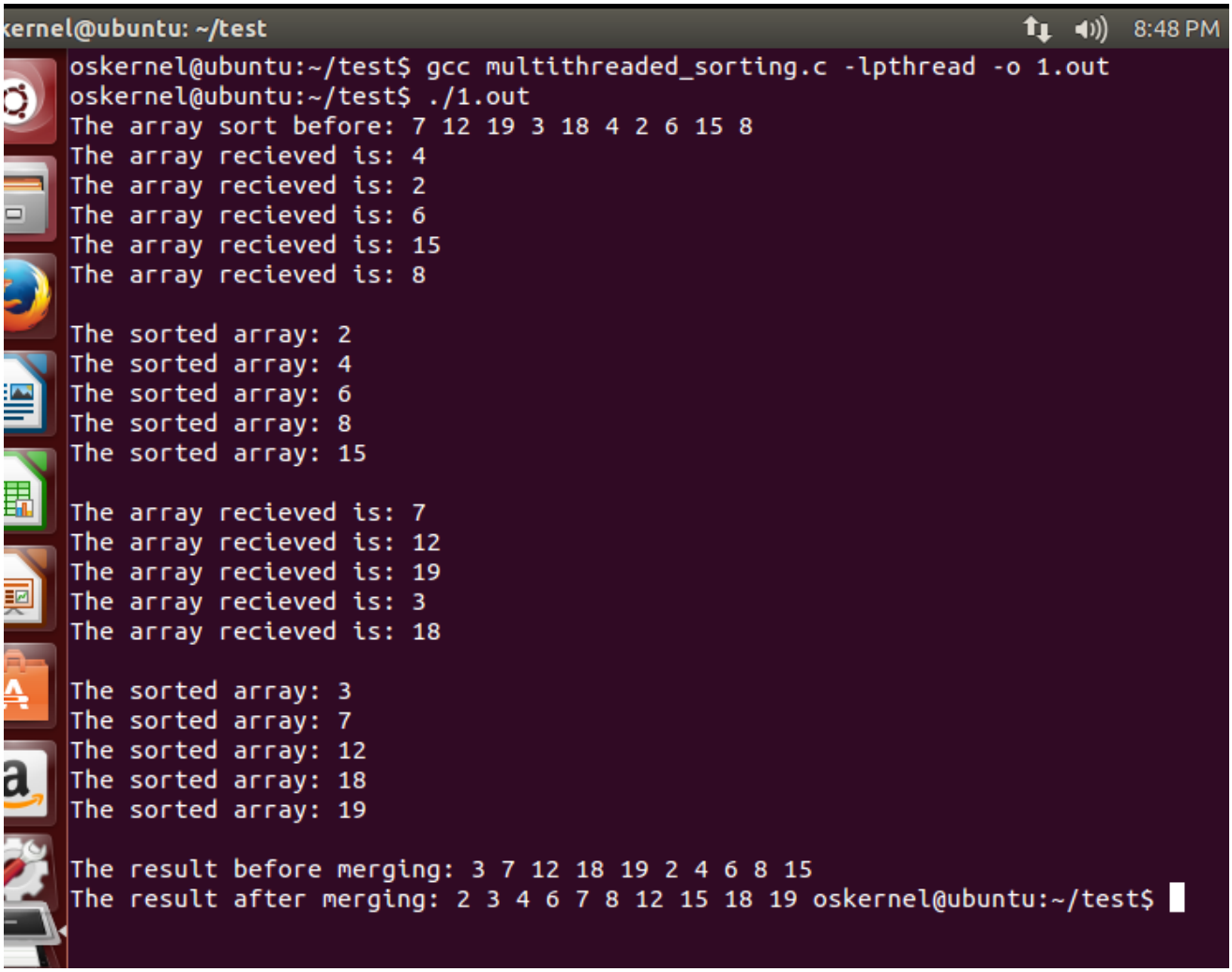
Pthread's using is so easy.

Firstly, we need to create the function which is need to running in a new thread. Such as the sorter and merger function in this project. Then we create a struct to store the parameters of the function. After, using pthread_create() function will start a new thread to run the function.

I store the array need to be sorted in a global array. Because that all thread share the same global variables.

In the sorter, I apply the Bubble Sort algorithm, after sort two subarray, I use the merge sort to merge two subarray which have been sorted.

Result:



2. Fork-Join Sorting Application

Implement the preceding project (Multithreaded Sorting Application) using Java's fork-join parallelism API. This project will be developed in two different versions.

1.QuickSort

This algorithm usually is realize base in recursive. In this project, I will extends the RecursiveTask class. Overrride the compute method.

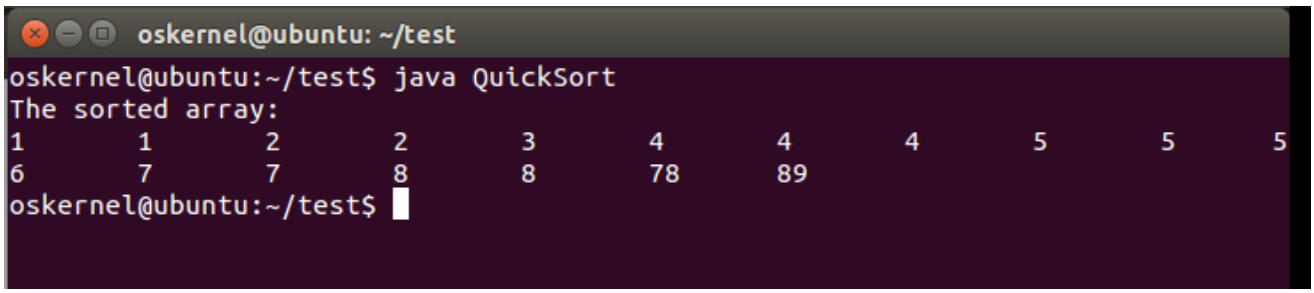
Quicksort is based in the partition function. In this function ,we will divide the array into two subarrays. Everytime I have divided the array, I will fork a thread to divide two array again untill it is sorted.

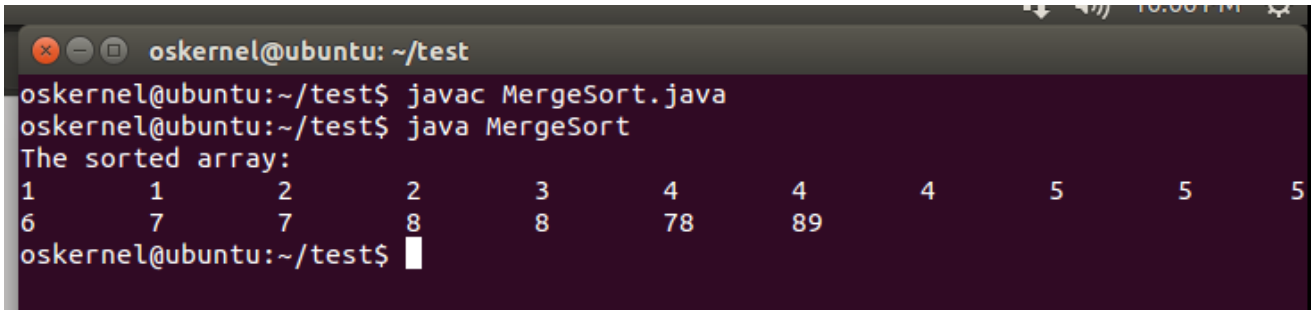
2. MergeSort

This algorithm is also realized based in recursive.

Everytime I divide the array into two array, them divide it into four array, untill the subarray's length is 1. Then we merge subarrays which are neibor. At last, we will get the result.

Result:


oskernel@ubuntu: ~/test
oskernel@ubuntu:~/test$ java QuickSort
The sorted array:
1 1 2 2 3 4 4 4 5 5 5
6 7 7 8 8 78 89
oskernel@ubuntu:~/test$


oskernel@ubuntu: ~/test
oskernel@ubuntu:~/test$ javac MergeSort.java
oskernel@ubuntu:~/test$ java MergeSort
The sorted array:
1 1 2 2 3 4 4 4 5 5 5
6 7 7 8 8 78 89
oskernel@ubuntu:~/test$