

Multithread Programming

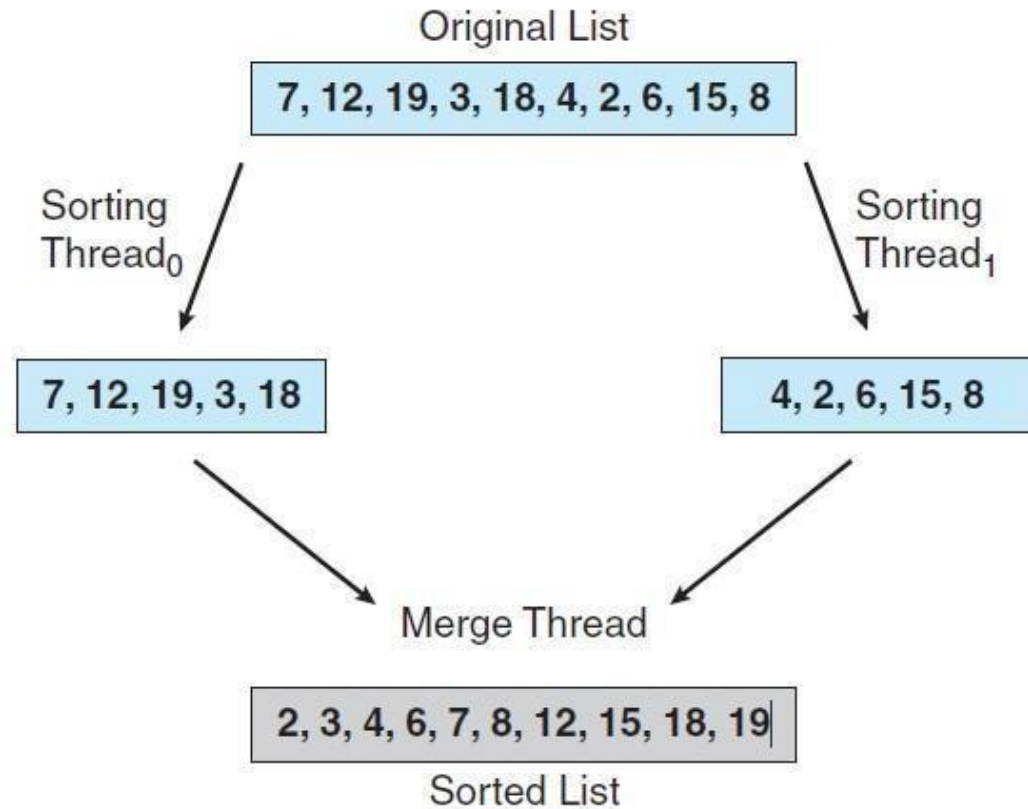
- In this project, you need to write a multithreaded sorting program that works as follow:

- A list of integers is divided into **two smaller lists of equal size**. You have to create **two separate threads(sorting threads)** to sort each sublist using the **merge sort** algorithm.

Multithreaded Sorting Application

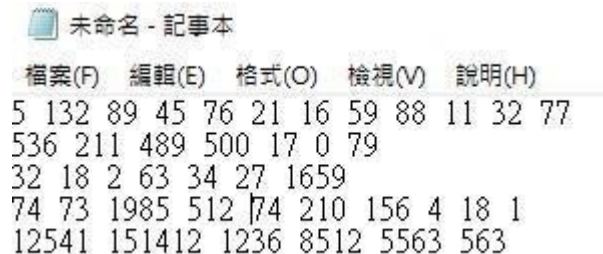
- Then, the two sublists are merged into a single sorted list by a **third thread(merging thread)**.

Multithreaded Sorting Application



Multithreaded Sorting Application

- We will give you a .txt file which contains several lines of **integers** as your input test data. Each line represents one test case which is composed by several numbers. So, you should figure out **how to use argc, argv way to read in the test data line by line first.**
- E.g.



```
未命名 - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
5 132 89 45 76 21 16 59 88 11 32 77
536 211 489 500 17 0 79
32 18 2 63 34 27 1659
74 73 1985 512 74 210 156 4 18 1
12541 151412 1236 8512 5563 563
```

Multithreaded Sorting Application

- Then, you need to implement the merge sort using multithreaded programming : **two threads for sorting, and one for merging.**
- Make sure that the third thread(merging thread) get started after both sorting threads are done, which means you have to **keep the merging thread waiting until the sorting threads finish their jobs by pthread condition wait.**
- Trace **pthread.h** first.
- Finally, you should **print out your running time for each test case on the screen, and output a .txt file for your sorting result.**

Multithreaded Sorting Application

- The output filename must be **output.txt**.
- Please output your result with **the same format as the input** we gave you.
- We will use more difficult test cases to test your program.
- **At most 10000 integers for one test case.**
- You can find some test cases to test your program by yourself.
- You should **implement under Linux**.
- Only **merge sort** is acceptable.

Multithreaded Sorting Application

- Must use `argc, argv` way to input the test file.
- Use `gcc hw3.c -pthread -o hw3.o` to compile your code.
- Your command to run and output your code **must** be `./hw3.o testcase.txt output.txt`
- Hw3_{studentID}.rar :
- hw3.c (90%)
- 10% for each hidden test cases, so you need to pass 9 test cases for full score 😊.

Multithreaded Sorting Application

- hw3_report (10%)
- Tell us how you implement your homework and show us your time and result with some screenshots.
- We will randomly pick 1/4 of all students to demo in person after the midterm.

Multithreaded Sorting Application

- **0 will be given to cheaters**, so don't copy & paste your friend's code directly.
- Make sure that you totally understand your code 😊.
- Deadline: **5/9(THU) 23:59**