2019 Fall Introduction to Operating Systems

Homework 3

0616015 劉姿利

1 Briefly describe the design for the sort and merge function and the thread management in the multi-thread program.

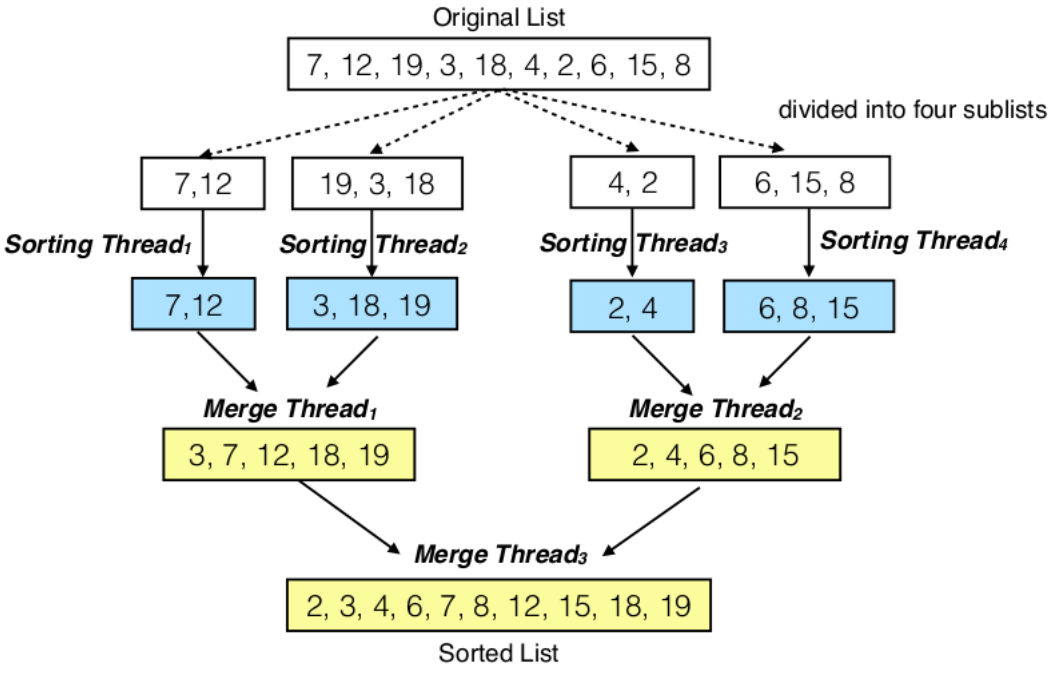
1.1 Sort

Brute-force bubble sort.

1.2 Merge

Two-pointer sorting.

1.3 Thread Management



- Let each thread be either sorting_thread or merge_thread.
- For each sorting_thread, just simple creat it and execute.
- For each merge_thread, we have first wait for other 2 threads before execution.

- merge_thread_1 waits for sorting_thread_1 and sorting_thread_2.
- merge_thread_2 waits for sorting_thread_3 and sorting_thread_4.
- merge_thread_3 waits for merge_thread_1 and merge_thread_2.

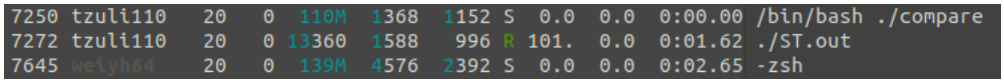
1.3.1 Waiting Design for Merge_threads

I pass the thread indexes to wait for by passing a structure S.

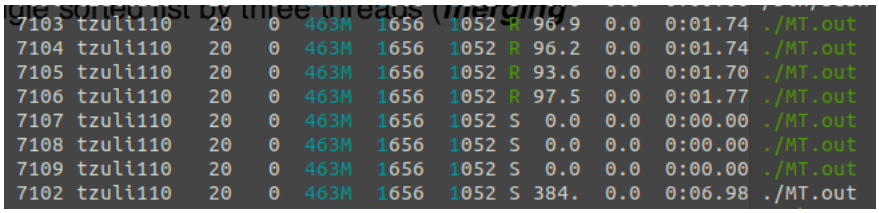
```
void *_merge(void* data) {
    S* ptr = (S*)data;
    int id1 = ptr->id1;
    int id2 = ptr->id2;

    pthread_join(id[id1] ,NULL);
    pthread_join(id[id2] ,NULL);

    ...
}
```

2 Show the thread information screenshot while running the single-thread/multi-thread program.

2.1 Single-thread

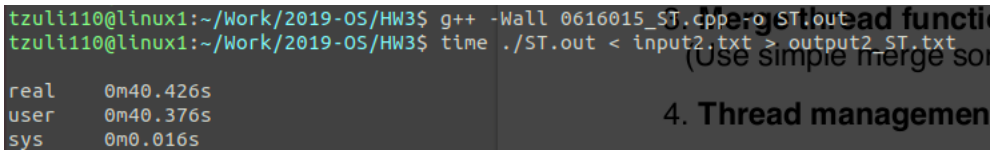


2.2 Multi-thread

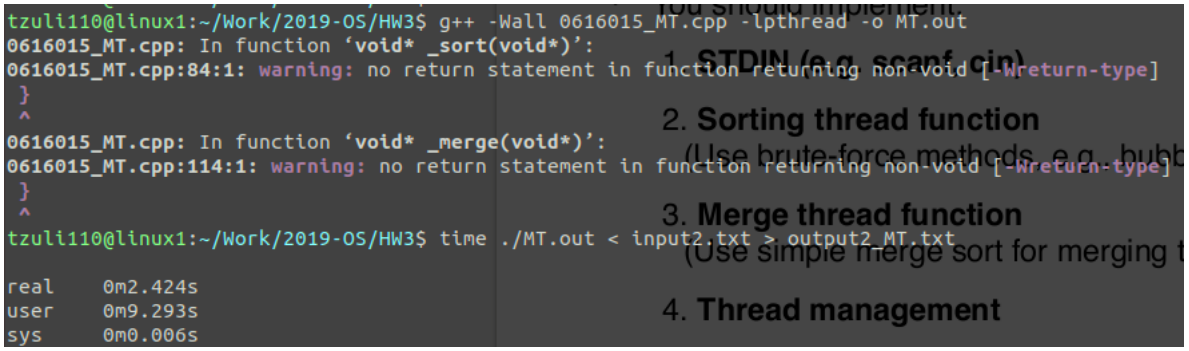


3 Show the time speedup between single-thread and multi-thread.

3.1 Single-thread



3.2 Multi-thread



4 What I learned from doing homework 3.

- How to use pthread api.
- How to check the number of threads by using htop.
- How to use stucture and type conversion to pass arguments into threads.