

Operating System Project Phase 2 Report

Zheng Li

July 30, 2019

1 Description

The program developed in phase 2 aims to accomplish the following tasks: 1) It is able to be compiled as kernel module. 2) Run a multi-thread sort-merge algorithm under kernel mode.

NOTE:

1. Since the `pthread.h` used in project phase 1 is a library used at POSIX level(user mode), it is very difficult to be imported in kernel mode. So in the second phase of this project, a new library `kthread.h` is used to support the multi-thread implementation in kernel mode.
2. The input list developed in phase 1 requires a local filename and the number of threads user input and the program then reads the content and execute. In the phase 2, because the program is running as a kernel module which cannot acquire user input. All input information (unsorted list, number threads) were hard coded inside the code. If any of the parameters need to be changed, please modify the source code accordingly.

2 Program structure

The overall structure of the phase 2 program is very similar to the phase 1 program with a few exceptions (e.g. *sort_fn()* and *merge_fn()*) which are mostly caused by the change of the threading library (< *pthread.h* > to < *kthread.h* >). Please refer phase 1 report for the pseudocode of each routine inside the program.

NOTE: This program follows the example code from [this tutorial](#) to execute the kthread library.

3 Test Run

The code can be successfully compiled with a few minor warnings (variable array declaration). Like the phase 1 program, this kernel module prompts several information during execution:

1. Unsorted list information (Array Size, Number of threads).
2. Each thread's ID (e.g. 0 , 1, 2) and its assigned task (SORT/MERGE).
3. Final sorted list.

The log files of the program execution can be found inside the "OS_PRJ_PHASE2_ZHENGLI.zip" and a screenshot of kernel module execution is also shown below:

NOTE: The source code is included in the file: "OS_PRJ_PHASE2_ZHENGLI.zip".

```

[root@os scratch]# insmod sort_kernel.ko
[10613.705895] =====
[10613.706485] |
[10613.707078] | Array Size:          12
[10613.707707] | Input File:          Hard coded Array
[10613.708224] | Number of Threads:    3
[10613.709665] |
[10613.710236] =====
[10613.710236]
[10613.711421] Thread#0 Created successfully
[10613.712293] Thread#0 -> SORT
[10613.712823] BEFORE SORT:2 5 3 1
[10613.712825] AFTER SORT:1 2 3 5
[10613.713349] -----
[10614.732021] Thread#1 Created successfully
[10614.733294] Thread#1 -> SORT
[10614.733793] BEFORE SORT:6 8 7 9
[10614.733795] AFTER SORT:6 7 8 9
[10614.734245] -----
[10615.745653] Thread#2 Created successfully
[10615.747350] Thread#2 -> SORT
[10615.748068] BEFORE SORT:53 23 3 4
[10615.748071] AFTER SORT:3 4 23 53
[10615.748861] -----
[10616.758928] Thread#3 Created successfully
[10616.760441] Thread#3 -> MERGE
[10616.761081] -----
[10618.972478]
[10618.972478] INPUT ARRAY:
[10618.973720] 2 5 3 1 6 8 7 9 53 23 3 4
[10618.973724]
[10618.973724] FINAL RESULT:
[10618.974923] 1 2 3 3 4 5 6 7 8 9 23 53
[10618.974926]
[root@os scratch]#

```