Operating System Project 2

1. Part One : Invoke FIFO Scheduler

1.1. sched_test.c outline

```
10    thread function () {
            // block 0.5 seconds
            . . .
      }
20    main () {
            // set CPU affinity
            . . .
            // set scheduling policy
            . . .
            // create two threads
            . . .
            // delete two threads
            . . .
      }
```

1.2. Functions implemented

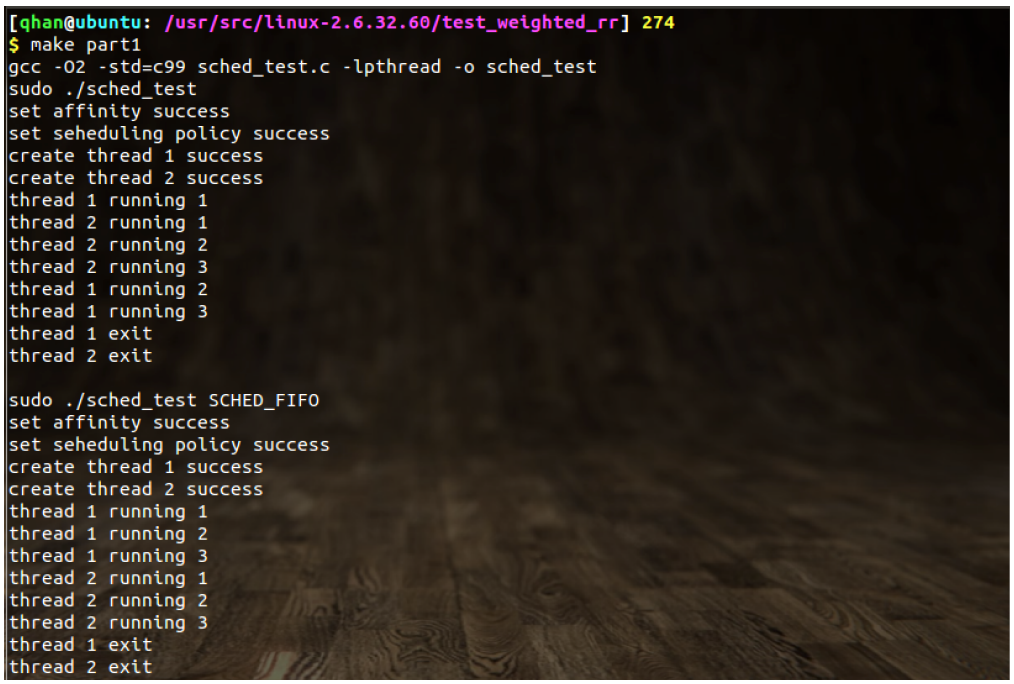
a. CPU_ZERO( ), CPU_SET( ), sched_setaffinity( ) :
   Set the affinity of the CPU, implemented with only one CPU.

b. sched_setscheduler( ) :
   Set the scheduling policy if “SCHED_FIFO” is activated.

c. pthread_create( ), pthread_join( ), pthread_exit( ) :
   Threads implementations.

d. clock( ) :
   Blocking methods in the thread function.

1.3. Result



2. Part Two : Weighted Round-Robin Scheduler

2.1. sched_weighted_rr.c outline

```
      . . .
34    enqueue_task_weighted_rr () {
            // add task to the tail, count++
            . . .
      }
42    dequeue_task_weighted_rr () {
            // detele task, count--
            . . .
      }
      . . .
66    yield_task_weighted_rr () {
            // move task to the tail
            . . .
      }
      . . .
84    pick_next_task_weighted_rr () {
            // if count is zero, return null
            . . .
            // else, pick out the first task in queue
            . . .
      }
      . . .
186   task_tick_weighted_rr () {
            // time slice--
            . . .
            // if time slice is zero
            . . .
      }
      . . .
```

2.2. Data structures

a. rq (in kernel/sched.c) :
   - weighted_rr (type : struct weighted_rr_rq) to specify the run queue for weighted RR.

b. weighted_rr_rq (in kernel/sched.c) :
   - queue (type : struct list_head) for queuing tasks.
   - nr_running (type : unsigned long) for counting the number of currently running tasks.

c. task_struct (in include/linux/sched.h) :
   - task_time_slice (type : unsigned int) for recording current available time slices for using CPU resources.
   - weighted_time_slice (type : unsigned int) for relocating the time slices of the tasks which have no more time slices to use.
   - weighted_rr_list_item (type : struct list_head) for reprensenting the task queuing unit.

d. sched_class (in kernel_weight_rr.c) :
   Accomplish the implementaion of weighted RR scheduler by assigning custom functions.

2.3. Functions implemented

a. list_add_tail( ), list_del( ), list_move_tail( ) :
   Managing the task queue.

b. list_first_entry( ) :
   Read the first task from the task queue.

2.4. Result

```
[qhan@ubuntu: /usr/src/linux-2.6.32.60/test_weighted_rr] 49
$ ./test_weighted_rr weighted_rr 10 5 100000000
sched_policy: 6, quantum: 10, num_threads: 5, buffer_size: 100000000
abcdeabcdabcababa

[qhan@ubuntu: /usr/src/linux-2.6.32.60/test_weighted_rr] 50
$ ./test_weighted_rr weighted_rr 10 5 200000000
sched_policy: 6, quantum: 10, num_threads: 5, buffer_size: 200000000
abcdeabcdeabcdabcabcababababa

[qhan@ubuntu: /usr/src/linux-2.6.32.60/test_weighted_rr] 51
$ ./test_weighted_rr weighted_rr 10 5 300000000
sched_policy: 6, quantum: 10, num_threads: 5, buffer_size: 300000000
abcdeabcdeabcdabcdabcabcabcababababababa

[qhan@ubuntu: /usr/src/linux-2.6.32.60/test_weighted_rr] 52
$ ./test_weighted_rr weighted_rr 10 5 500000000
sched_policy: 6, quantum: 10, num_threads: 5, buffer_size: 500000000
abcdeabcdeabcdeabcdabcdabcdabcabcabcabcabababababababababababa
```