

# OS PJ2 Report

B04902015王哲愷 B04902103蔡昀達

## Part1:

```
rendybig@rendybig-VirtualBox:~$ cd Desktop/
rendybig@rendybig-VirtualBox:~/Desktop$ gcc sched_test.c -lpthread -std=c99 -o sched_test
rendybig@rendybig-VirtualBox:~/Desktop$ sudo ./sched_test
[sudo] password for rendybig:
Thread 1 was created
Thread 2 was created
Thread 2 is running
Thread 1 is running
Thread 1 is running
Thread 2 is running
Thread 1 is running
Thread 2 is running
rendybig@rendybig-VirtualBox:~/Desktop$ sudo ./sched_test SCHED_FIFO
Thread 1 was created
Thread 2 was created
Thread 1 is running
Thread 1 is running
Thread 1 is running
Thread 2 is running
Thread 2 is running
Thread 2 is running
```

### Details:

(FIFO)直接將先建立的 thread 的 priority 設為最大(99)即可。

(FIFO)以 sudo 執行很重要，否則無法使用 real-time scheduling。

## Part2:

### 1. enqueue\_task\_weighted\_rr()

函數將會給予 task 和 rq，將 task 根據 rq 的資料結構重新將 task 放入。運行 `rq->weighted_rr.nr_running++` 增加 rq 中的計數，隨後回報該 scheduler 中有沒有存在 task 需要運行。

### 2. dequeue\_task\_weighted\_rr()

當 task 完成任務時，呼叫 `update_curr_weighted_rr` 進行統計運行多久，並且更新與 task 相關的排程權重。接著將 task 從 rq 串列中移除，並且更新 scheduler 的 task 計數 (`rq->weighted_rr.nr_running--`)。

### 3. yield\_task\_weighted\_rr()

直接運行 requeue

### 4. pick\_next\_task\_weighted\_rr()

當最上層分配一段時間給 scheduler 運行，運行時會調用這個函數，並拿取要執行的 task，但並不用將其移除串列，並執行 `next->se.exec_start = rq-`

