

## OS Project1 Report

B04902067 林育志

### Environment and design:

Using Oracle virtual box 6.1 on Ubuntu 16.04 LTS, kernel 4.14.25(modified)

Scheduler runs on CPU0, all children run on CPU1. The speed of running “unit of time” of the two CPU has an approximate 2% difference, which causes the answers sometimes to be unstable( especially when testing RR, where the first child exits after tens of thousands units of time.

My machine has approximately 1.5~1.8s/500 units of time.

### FIFO:

Sort by ready time, child are then given priority equal to  $99-i$ , which  $i$  is there number of place of ready time.

### RR:

When a child has run 500 unit time, it is then put to state IDLE, and the one that's suppose to run next has it's priority set to 99.

The main data structure is a queue, and all new child are placed at the end of the queue when spawned.

### SJF:

The children are first sorted by ready time. There's another vector storing ready processes. When a new child is spawned, it's put to sleep, and the vector is sorted by execution time. When there's no child running, the scheduler pick the child with least execution time to wake.

### PSJF:

Basically same as SJF, but if the new child has less execution time then that current running, the original running one is put to sleep ( at the front of the vector), and the new process starts running instead. If the new process has higher execution time then the on running has left, it has no difference to SJF.

### Comparison and Explanation:

FIFO : The result is close to theory.

RR: The processes that are supposed to finish in the same round robin are sometimes reversed, especially when the first process is involved. This happens more often when the total time is more than 10000 units of time. It's could be caused by the two CPU running with different speed, so the child terminates earlier/later.

SJF/PSJF: result is close to theory.