

# 2020 OS Project1

b07902027 高偉倫

## 1. 設計

### Scheduler

- 在每一個time unit結束時，執行下列步驟：
  - 檢查是否有已完成的process
    - 如果有，wait(process)，印出資訊
  - 檢查是否有process ready
    - 如果有，創造process並將其移進waiting queue，記錄結束時間
    - **將start time設成process產生時間**
  - 依據不同的scheduling policy，找出下一個該執行的process
    - FIFO\_next(), RR\_next(), SJF\_next(), PSJF\_next()
  - 將此process移進CPU，並將CPU內的process移出(如果是同個process就不移)
  - 經過1 unit time，把執行中的 process.t\_exec--
- 持續到所有process.t\_exec皆為0

### Scheduling Policy

- FIFO\_next()：
  - 如果當前時間有process在執行，回傳其id
  - 若無，回傳ready time最小的未執行process
- RR\_next()：
  - 如果當前無執行process，直接把第一個ready的process送入CPU
  - 若當前有執行process，檢查是否已經過一個time quantum
    - 若是，找出下一個該執行的process
    - 若否，回傳當前process id
- SJF\_next()：
  - 因為non preemptive，若當前有process執行，回傳其id
  - 若無，找出剩餘執行時間最短的process id
- PSJF\_next()：
  - 因為preemptive，在每一個時間點都回傳剩餘執行時間最短的process id

## 2. 核心版本

```
b07902027@b07902027-VirtualBox:~/p1$ uname -mrs  
Linux 4.14.25 x86_64
```

## 3. 比較

- 觀察輸出結果可以發現，儘管順序符合預期，process的結束時間還是會比預期結束時間來的長一些。我想這可能是因為，scheduler在每一個timestamps時找下一個process時，在我的實作上需要花費的時間是 $O(N)$ ，因此增加了誤差。