

# OS Project 1 report By B07902109張翔文

## 設計

這題我用雙核心來分別跑Scheduler跟fork出來的Child，每當一個process進來的時候就為他fork一個process然後馬上Block住以免他偷跑，之後再用scheduler來控制誰可以跑誰不可以跑。

### Block的方法:

如果是想要讓他跑的程式就設他為SCHED\_FIFO(99)，想要block住的就設SCHED\_IDLE，由於兩者之間cpu優先順序差距太大故priority小的process不會搶priority高的process，測試時發現process有可能會偷跑，光改priority可能不夠，所以再加上while(lock)來預防偷跑的狀況。

### 控制各種不同的Cpu排程:

FIFO, SJF等不同排程的差別就在選擇process的順序，故我分別為每一個排程都設一個函式回傳下一個要換誰跑。

#### FIFO

1. 要做到non preemtive
2. 先進來的先跑

#### SJF

1. 要做到non preemtive
2. execute\_time短的先跑

#### PSJF

1. preemtive
2. execute\_time短的先跑

#### RR

1. preemtive
2. 每500單位時間就換人跑

## 核心版本

核心: Linux ubuntu 4.14.25  
OS: Ubuntu 16.04

## 比較實際結果與理論結果，並解釋造成差異的原因

最後我的程式跑得比理論結果還要多一些時間，這是由於理論結果是完全沒考慮到process之間轉換時花的時間，實際上這一部分也要算進去才會準，assign cpu, 函式間切換等都會花時間，再加上整個VM也要跑別的程式也會占用cpu發生context switch，這就是造成誤差的原因。