# OS Project1第一組 -Process scheduing / Report

#### 0 Member

簡辰宇、黃文鴻、徐子秦、倪嘉宏

## 1 The design

### /General

將四種方法之函數做成標頭檔，由 main include。

實作 printk 部分， test\_module.ko 會創造一個 /dev/osproj 裝置來接收由 main 傳進的 pid 號碼，同時以 OSPROJ\_PROC\_START 或是 OSPROJ\_PROC\_END 作為 ioctl 的參數:一個表示程式的開始，一個表示程式的結束；當 pid 號碼傳入時，test\_module 會判斷系統時間，首先會存下程式開始時間，待程式結束時再一起跟一起印出。

接著將所有檔案合併為一 .src 檔。

### /main.c

接受來自 STDIN 的 input，判斷 task 類型(FIFO/SJF/RR/PSJF)，並開啟對應函數。

### /FIFO.c

1 在 input 中找出最早進入的程序並執行，若系統時間早於最早進入時間，則空轉到該時間為止再開始。

2 active\_num(仍待執行程序總量)歸零時，return。

/SJF.c

1 在 input 找出所有已進入程序。

2 找出其中具最短執行時間者執行。

3 回到判斷區間，再次找出所有已進入程序。

4 active\_num(仍待執行程序總量)歸零時，return。

### /RR.c

1 對input的行程依序執行

2 給予該程序 500 單位的執行時間

3 若未在500單位時間跑完，則用kill( pid, SIGSTOP); 暫停行程

並放到residue\_Queue

4 執行第一round完畢

5 取出residue\_Queue資料，喚醒暫停的行程，一樣最多給500單位的時間

跑不完再塞回residue\_Queue

6 直到行程執行完畢

### /PSJF.c

1.檢查是否有行程就緒且仍未完成

2.若有則將pi設定為 符合條件者且剩餘時間最短者的編號，若無則不改變pi

3.若pi可執行，則執行一次

4.若Pi不可執行，則浪費一個單位時間

5.若有行程完成，將其標記為已完成

6.循環以上步驟直至所有行程被標記為已完成

## 2 The results



## 3 The comparison between your output and theoretical result, and explanation

輸出格式與作業 spec 要求不同，作業要求中，開始執行時間與結束執行時間印在同一行，但我們是行程開始和結束時，各 call 一次 printk，因此分成兩行做輸出。

## 4 The contribution of each member

/結案報告：共同編輯

/System call device construction/code combination：黃文鴻

/FIFO/SJF implementation：簡辰宇

/Round Robin implementation：徐子秦

/PSJF implementation：倪嘉宏