

# OS Project 1

## B06902052 張集貴

### 1. 設計：

主程式跟他 fork 出的子行程一開始會用 sched\_setaffinity() 把執行的 CPU 設成編號 0 的 CPU。首先把輸入讀進陣列 a，然後按照 R 排序。用陣列 b 儲存所有當前已經創造出來但工作尚未結束的 process。mainloop() 裡的迴圈用 t 表示當前經過了多少單位時間，每執行一次就 t++，當所有 process 都創造出來且 b 裡面沒有待執行的 process 時，mainloop 才結束。每一單位時間的開始，如果 a[i].R == t，則把 a[i] 代表的行程創造出來並把相關資料存進 b 裡。每一單位時間的開始跟結束，mainloop() 會呼叫 update\_queue()，update\_queue() 會把已結束的 process 從 b 移除，並根據當前的 policy(FIFO, RR, SJF, PSJF) 來修改 b 的內容，把要執行的 process 放在 b[0]。b[0] 的 priority 會用 sched\_setscheduler() 調成 SCHED\_FIFO 的 99，其他的 process 的 priority 是 SCHED\_IDLE 的 0。

### 2. 核心版本：4.19.118

### 3. 比較實際結果與理論結果：用行程的產生時間當成 start time，結束時間為 finish time。

FIFO\_1.txt

Processes	R	T	Start Time(ST[1]~ST[5])	Finish Time(FT[1]~FT[5])	Turnaround Time(TT[1]~TT[5])	FT[i] – FT[i -1]
P1	0	500	1588060905.707037494	1588060910.422401066	4.715363502502441	
P2	0	500	1588060905.707071998	1588060911.189348063	5.48227596282959	0.7669470310211182
P3	0	500	1588060905.707107649	1588060911.965659785	6.258552312850952	0.7763118743896484
P4	0	500	1588060905.707145046	1588060912.730970792	7.0238258838653564	0.7653110027313232
P5	0	500	1588060905.707180217	1588060913.493322179	7.786141872406006	0.7623512744903564

雖然 Turnaround Time 偏大但是 FT[i] – FT[i – 1] 都接近 0.76，可以相信 500 單位時間約為 0.76 秒，前面 4 秒是 scheduler 在做預備動作需要的時間。

### PSJF\_2.txt

Processes	R	T	Start Time(ST[1]~ST[5])	Finish Time(FT[1]~FT[5])	Turnaround Time(TT[1]~TT[5])
P1	0	3000	1588061932.845396090	1588061948.353210228	15.5078141689301
P2	1000	1000	1588061934.390112720	1588061939.040340437	4.65022778511047
P3	2000	4000	1588061939.040461565	1588061966.926949412	27.8864879608154
P4	5000	2000	1588061949.899188512	1588061959.181615512	9.28242707252502
P5	7000	1000	1588061959.181747601	1588061962.299578755	3.117830991745

創造順序: P1, P2, P3, P4, P5

結束順序: P2, P1, P4, P5, P3

執行順序跟理論結果吻合。

### RR\_3.txt

Processes	R	T	Start Time(ST[1]~ST[5])	Finish Time(FT[1]~FT[5])	Turnaround Time(TT[1]~TT[5])
P1	1200	5000	1588061380.124001152	1588061449.734181290	69.6101801395416
P2	2400	4000	1588061381.984325191	1588061451.288271915	69.3039467334747
P3	3600	3000	1588061383.849611088	1588061434.212401175	50.3627901077271
P4	4800	7000	1588061385.714467167	1588061473.012646843	87.2981798648834
P5	5200	6000	1588061386.338955946	1588061469.907477500	83.5685217380524
P6	5800	5000	1588061387.276455414	1588061463.715998727	76.4395432472229

創造順序: P1, P2, P3, P4, P5, P6

結束順序: P3, P1, P2, P6, P5, P4

執行順序跟理論結果吻合。

### SJF\_4.txt

Processes	R	T	Start Time(ST[1]~ST[5])	Finish Time(FT[1]~FT[5])	Turnaround Time(TT[1]~TT[5])
P1	0	3000	1588061806.044908144	1588061821.651364719	15.6064567565918
P2	1000	1000	1588061807.596051750	1588061823.206046544	15.6099948883057
P3	2000	4000	1588061809.145391800	1588061833.863397042	24.7180054187775
P4	5000	2000	1588061824.747007542	1588061840.176019964	15.429012298584
P5	7000	1000	1588061827.840858793	1588061835.570246013	7.7293872833252

創造順序: P1, P2, P3, P4, P5

結束順序: P1, P2, P3, P5, P4

執行順序跟理論結果吻合。