

# OS Project 1 Report



B06902020 唐浩

## 設計



### - 核心架構



1. 主程序綁到第1顆CPU( `sched_setaffinity()` )，先讀進所有測資，並根據各排程sort子程序的資料，然後儲存至shared memory( `#include<sys/ipc.h> & #include<sys/shm.h>` )
2. 主程序根據測資，在達到ready time時依序fork()子程序，並根據各個排程決定是否讓該子程序執行。(使用一個變數 `turn` 代表，由於該變數存在shared memory，故所有程序都能存取。)
3. 子程序誕生後馬上把自己綁到第2顆CPU，並根據 `turn` 變數決定是否讓自己 `pause()`，即執行或沉睡。
4. 主程序fork()完所有子程序後，會進入wait()等待它們結束。
5. 根據各個排程，主程序和子程序之間會以signal的方式溝通，控制各個子程序的喚醒與沉睡。

### - Signals

1. `SIGUSR1`：當子程序 `exit()` / `pause()` 時，通知主程序。
2. `SIGUSR2`：主程序喚醒子程序。
3. `SIGUSR3` (自定義，取代 `SIGXFSZ`)：子程序preempt其它子程序。



### - 排程



- 以下所有排程的第1.點都是從「主程序判斷達到ready time時」開始，描述各子程序的運作。
- 所有子程序的資訊都存在shard memory，供主程序排程。

#### (1) FIFO



1. 主程序會判斷當前是否有其它子程序正在執行，若有則 `turn=0` (阻止新子程序執行)，否則 `turn=1`。
2. 子程序執行結束後，會送 `SIGUSR1` 給主程序，通知主程序喚醒下一個子程序，並讓自己 `pause()`。
3. 主程序根據FIFO性質，送 `SIGUSR2` 喚醒下一個子程序。(若沒有子程序在ready queue則不做任何事。)



#### (2) RR



1. 主程序會判斷當前是否有其它子程序正在執行，若有則 `turn=0` (阻止新子程序執行)，否則 `turn=1`。
2. 子程序執行結束後或執行完一個time quantum後，會送 `SIGUSR1` 給主程序，通知主程序喚醒下一個子程序，並讓自己 `pause()`。
3. 主程序根據RR性質，送 `SIGUSR2` 喚醒下一個子程序(有可能又喚醒原本的程序)。(若沒有子程序在ready queue則不做任何事。)



### (3) SJF

1. 主程序會判斷當前是否有其它子程序正在執行，若有則  $turn=0$  (阻止新子程序執行)，否則  $turn=1$ 。
2. 子程序執行結束後，會送 SIGUSR1 給主程序，通知主程序喚醒下一個子程序，並讓自己 pause()。
3. 主程序根據SJF性質，送 SIGUSR2 喚醒下一個子程序。(若沒有子程序在ready queue則不做任何事。)

### (4) PSJF

1. 主程序會判斷當前是否有其它子程序的remain time小於等於新子程序的execution time，若有則  $turn=0$  (阻止新子程序執行)，否則  $turn=1$ 。
2. 若  $turn=1$ ，則新子程序會送 SIGUSR3 給當前正在執行的子程序，通知其 pause()。
3. 子程序執行結束後，會送 SIGUSR1 給主程序，通知主程序喚醒下一個子程序，並讓自己 pause()。
4. 主程序根據PSJF性質，送 SIGUSR2 喚醒下一個子程序。(若沒有子程序在ready queue則不做任何事。)

## 核心版本

- 5.4
- Reference: [Compile 5.4.2 Linux Kernel with a new Syscall for Ubuntu 18.04](https://fenghe.us/compile-5-4-2-linux-kernel-with-a-new-syscall-for-ubuntu-18-04-in-vmware/) (<https://fenghe.us/compile-5-4-2-linux-kernel-with-a-new-syscall-for-ubuntu-18-04-in-vmware/>)

## 比較

- 以 TIME\_MEASUREMENT\_dmesg.txt 之平均執行時間為基準計算理論unit time。(約為 0.00147秒)
  - 每個比較表格將第一個執行之子程序的開始時間對齊至0。
  - 每個比較表格以子程序執行結束之時間排序。
  - 所有時間四捨五入到小數點後第5位。
  - 所有理論時間與實際時間的差距中，差距最大者約相差 7.56016%，最小者約相差 0.0047%。(所有表格下方有略為討論。)
- (公式為:  $(|\text{Practical running} - \text{Theo. running}| / \text{Theo. running}) \times 100\%$ )

#### 1. FIFO\_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.733450	0.733450	0.000000	0.731750	0.731750
P2	0.733450	1.466900	0.733450	0.746150	1.476970	0.730820
P3	1.466900	2.200350	0.733450	1.486110	2.216570	0.730460
P4	2.200350	2.933800	0.733450	2.219290	2.951560	0.732270
P5	2.933800	3.667250	0.733450	2.956280	3.689680	0.733400

## 2. FIFO\_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	117.3521	117.3521	0.000000	119.34512	119.34512
P2	117.3521	124.68661	7.334510	119.34781	126.93986	7.592050
P3	124.68661	126.15351	1.466900	126.94028	128.43203	1.491750
P4	126.15351	127.62041	1.466900	128.43359	129.92428	1.490690

## 3. FIFO\_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	11.73521	11.73521	0.000000	12.17888	12.17888
P2	11.73521	19.06972	7.334510	12.17963	19.63652	7.456890
P3	19.06972	23.47042	4.400700	19.64048	24.05530	4.414820
P4	23.47042	24.93732	1.466900	24.06038	25.53226	1.471880
P5	24.93732	26.40422	1.466900	25.53342	26.99667	1.463250
P6	26.40422	27.87112	1.466900	27.00325	28.46705	1.463800
P7	27.87112	33.73873	5.867610	28.47124	34.34465	5.873410

## 4. FIFO\_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	2.933800	2.933800	0.000000	3.047120	3.047120
P2	2.933800	3.667250	0.733450	3.049800	3.819990	0.770190
P3	3.667250	3.960630	0.293380	3.821910	4.132930	0.311020
P4	3.960630	4.694080	0.733450	4.133150	4.904830	0.771680

## 5. FIFO\_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	11.73521	11.73521	0.000000	11.89758	11.89758
P2	11.73521	19.06972	7.334510	11.90347	19.36605	7.462580
P3	19.06972	23.47042	4.400700	19.36749	23.78907	4.421580
P4	23.47042	24.93732	1.466900	23.79104	25.25663	1.465590
P5	24.93732	26.40422	1.466900	25.26792	26.75906	1.491140
P6	26.40422	27.87112	1.466900	26.77111	28.28110	1.509990
P7	27.87112	33.73873	5.867610	28.28248	34.22203	5.939550

## 6. PSJF\_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P4	4.400700	8.801410	4.400710	4.665600	9.217150	4.551550
P3	2.933800	14.66901	11.73521	3.147960	15.20070	12.05274
P2	1.466900	23.47042	22.00352	1.607300	24.18484	22.57754
P1	0.000000	36.67253	36.67253	0.000000	37.43523	37.43523

## 7. PSJF\_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P2	1.466900	2.933800	1.466900	1.483950	2.955870	1.471920
P1	0.000000	5.867610	5.867610	0.000000	5.907430	5.907430
P4	7.334510	10.26831	2.933800	7.415950	10.36805	2.952100
P5	10.26831	11.73521	1.466900	10.37859	11.84479	1.466200
P3	5.867610	16.13591	10.26830	5.925540	16.22900	10.30346

## 8. PSJF\_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P2	0.733450	1.466900	0.733450	0.752430	1.496460	0.744030
P3	1.466900	2.200350	0.733450	1.530630	2.276440	0.745810
P4	2.200350	2.933800	0.733450	2.297490	3.038810	0.741320
P1	0.000000	5.134150	5.134150	0.000000	5.216230	5.216230

9. PSJF\_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P3	0.146690	1.613590	1.466900	0.151460	1.622830	1.471370
P2	0.000000	4.400700	4.400700	0.000000	4.459560	4.459560
P4	4.400700	10.26831	5.867610	4.467920	10.47064	6.002720
P1	10.26831	20.53662	10.26831	10.47083	20.91135	10.44052

10. PSJF\_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.146690	0.146690	0.000000	0.157780	0.157780
P3	0.146690	0.440070	0.293380	0.158360	0.457720	0.299360
P2	0.440070	6.307680	5.867610	0.457980	6.445880	5.987900
P4	6.307680	12.17528	5.867600	6.447110	12.47086	6.023750
P5	12.17528	22.44359	10.26831	12.47187	22.93413	10.46226

11. RR\_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.733450	0.733450	0.000000	0.737410	0.737410
P2	0.733450	1.466900	0.733450	0.740900	1.480250	0.739350
P3	1.466900	2.200350	0.733450	1.483140	2.215900	0.732760
P4	2.200350	2.933800	0.733450	2.218770	2.951820	0.733050
P5	2.933800	3.667250	0.733450	2.954240	3.688080	0.733840

12. RR\_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	11.00176	11.00176	0.000000	11.08228	11.08228
P2	0.733450	13.20211	12.46866	0.734140	13.31292	12.57878

## 13. RR\_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P3	4.400710	24.93732	20.53661	4.491850	25.25136	20.75951
P1	0.000000	26.40423	26.40423	0.000000	26.72187	26.72187
P2	2.200350	27.87113	25.67078	2.231240	28.20156	25.97032
P6	10.26831	39.60634	29.33803	10.41025	40.12434	29.71409
P5	8.801410	42.54014	33.73873	8.933280	43.07536	34.14208
P4	7.334510	44.00704	36.67253	7.455170	44.55068	37.09551

## 14. RR\_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P4	2.200350	8.067960	5.867610	2.214180	8.125390	5.911210
P5	2.933800	8.801410	5.867610	2.946060	8.869710	5.923650
P6	3.667250	9.534860	5.867610	3.694500	9.611610	5.917110
P3	1.466900	21.27007	19.80317	1.469060	21.52831	20.05925
P7	5.134150	27.13767	22.00352	5.183930	27.38943	22.20550
P2	0.733450	29.33803	28.60458	0.731930	29.60366	28.87173
P1	0.000000	33.73873	33.73873	0.000000	34.01371	34.01371

## 15. RR\_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P4	2.200350	8.067960	5.867610	2.213670	8.112780	5.899110
P5	2.933800	8.801410	5.867610	2.951920	8.848260	5.896340
P6	4.400700	10.26831	5.867610	4.425760	10.34171	5.915950
P3	1.466900	21.27007	19.80317	1.478300	21.39184	19.91354
P7	5.134150	27.13767	22.00352	5.167920	27.31739	22.14947
P2	0.733450	29.33803	28.60458	0.739920	29.57492	28.83500
P1	0.000000	33.73873	33.73873	0.000000	33.98972	33.98972

## 16. SJF\_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P2	0.000000	2.933800	2.933800	0.000000	2.927660	2.927660
P3	2.933800	4.400700	1.466900	2.931680	4.399780	1.468100
P4	4.400700	10.26831	5.867610	4.400910	10.31228	5.911370
P1	10.26831	20.53662	10.26831	10.32446	20.80249	10.47803

## 17. SJF\_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.146690	0.146690	0.000000	0.150870	0.150870
P3	0.146690	0.440070	0.293380	0.153340	0.457730	0.304390
P2	0.440070	6.307680	5.867610	0.457910	6.382280	5.924370
P4	6.307680	12.17528	5.867600	6.385660	12.32427	5.938610
P5	12.17528	22.44359	10.26831	12.32513	22.70835	10.38322

## 18. SJF\_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	4.400700	4.400700	0.000000	4.443310	4.443310
P4	4.400700	4.415370	0.014670	4.446640	4.462190	0.015550
P5	4.415370	4.430040	0.014670	4.463150	4.478610	0.015460
P6	4.430040	10.29765	5.867610	4.481030	10.38922	5.908190
P7	10.29765	16.16525	5.867600	10.39156	16.24776	5.856200
P2	16.16525	23.49976	7.334510	16.25193	23.61259	7.360660
P3	23.49976	33.76807	10.26831	23.61362	34.02381	10.41019
P8	33.76807	46.97018	13.20211	34.02794	47.22943	13.20149

## 19. SJF\_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	4.400700	4.400700	0.000000	4.440520	4.440520
P2	4.400700	5.867610	1.466910	4.443980	5.918350	1.474370
P3	5.867610	11.73521	5.867600	5.919900	11.78583	5.865930
P5	11.73521	13.20211	1.466900	11.80563	13.26818	1.462550
P4	13.20211	16.13591	2.933800	13.27194	16.19545	2.923510

## 20. SJF\_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	2.933800	2.933800	0.000000	2.926860	2.926860
P2	2.933800	3.667250	0.733450	2.927810	3.658270	0.730460
P3	3.667250	4.400700	0.733450	3.659800	4.391250	0.731450
P4	4.400700	5.134150	0.733450	4.396040	5.127220	0.731180

- 討論

在理論時間的估計算法下：每個unit的時間都是固定的，且一個程序結束後可以立即執行下一個程序，context switch也可以瞬間完成。

但實際執行時，我的作法是以signal模擬context switch等切換程序的功能，而這些都需要時間，就會造成誤差。

（即使不以signal模擬，真實情況下的context switch也一樣需要時間。）

另外，我們也無法保證CPU每次執行unit的時間在取到奈秒精度後仍然相同，而執行多個unit勢必也會使誤差擴大。

舉例說明：FIFO\_2.txt 的 80000 units 在實際與理論時間的差距便來到0.95秒。

然而，若我們以百分比的衡量方式來看，該筆測資的誤差僅為 1.69832%，由此可見本模擬方法的準確度。

（如上所述，最大誤差為 7.56016%，也並未超過10%。）