

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

- 4.15.0-45-generic (by uname -r)(無需編譯kernel)
- 使用 clock_gettime() 的 CLOCK_REALTIME 得到ns精度的時間。
- 使用 sudo 權限將資訊寫到 /dev/kmsg 。

比較

- 以 TIME_MEASUREMENT_dmesg.txt 之平均執行時間為基準計算理論unit time 。
 - 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4位 。
 - 所有理論時間與實際時間的差距中，差距最大者約相差 8.7669% ，最小者約相差 0.0117% 。
- (所有表格下方有略為討論 。)
- (公式為: (|Practical running – Theo. running| / Theo. running)×100%)

1. FIFO_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.858100	0.858100	0.000000	0.925800	0.925800
P2	0.858100	1.716200	0.858100	0.928700	1.808600	0.879900
P3	1.716200	2.574300	0.858100	1.809300	2.677600	0.868300
P4	2.574300	3.432400	0.858100	2.680200	3.552600	0.872400
P5	3.432400	4.290500	0.858100	3.554000	4.417100	0.863100

2. FIFO_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	137.2947	137.2947	0.000000	143.1623	143.1623
P2	137.2947	145.8756	8.580900	143.1628	152.0195	8.856700
P3	145.8756	147.5918	1.716200	152.0229	153.7419	1.719000
P4	147.5918	149.3080	1.716200	153.7426	155.4260	1.683400

3. FIFO_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	13.72950	13.72950	0.000000	14.20930	14.20930
P2	13.72950	22.31040	8.580900	14.21020	22.29750	8.087300
P3	22.31040	27.45890	5.148500	22.29810	27.03530	4.737200
P4	27.45890	29.17510	1.716200	27.03950	28.69710	1.657600
P5	29.17510	30.89130	1.716200	28.69740	30.30080	1.603400
P6	30.89130	32.60750	1.716200	30.30160	32.03310	1.731500
P7	32.60750	39.47220	6.864700	32.03350	38.54570	6.512200

4. FIFO_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	3.432400	3.432400	0.000000	3.551300	3.551300
P2	3.432400	4.290500	0.858100	3.551900	4.426100	0.874200
P3	4.290500	4.633700	0.343200	4.426400	4.757600	0.331200
P4	4.633700	5.491800	0.858100	4.758500	5.616700	0.858200

5. FIFO_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	13.72950	13.72950	0.000000	14.08130	14.08130
P2	13.72950	22.31040	8.580900	14.08230	22.92420	8.841900
P3	22.31040	27.45890	5.148500	22.92590	28.22300	5.297100
P4	27.45890	29.17510	1.716200	28.22350	30.02880	1.805300
P5	29.17510	30.89130	1.716200	30.02950	31.74940	1.719900
P6	30.89130	32.60750	1.716200	31.74990	33.54350	1.793600
P7	32.60750	39.47220	6.864700	33.54410	40.59740	7.053300

6. PSJF_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P4	5.148600	10.29710	5.148500	5.210000	10.12170	4.911700
P3	3.432400	17.16180	13.72940	3.528800	17.31590	13.78710
P2	1.716200	27.45890	25.74270	1.772800	28.12170	26.34890
P1	0.000000	42.90460	42.90460	0.000000	42.71220	42.71220

7. PSJF_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P2	1.716200	3.432400	1.716200	1.664000	3.333500	1.669500
P1	0.000000	6.864700	6.864700	0.000000	6.746300	6.746300
P4	8.580900	12.01330	3.432400	8.563600	11.99020	3.426600
P5	12.01330	13.72950	1.716200	11.99040	13.65260	1.662200
P3	6.864700	18.87800	12.01330	6.746600	18.82510	12.07850

8. PSJF_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P2	0.858100	1.716200	0.858100	0.864200	1.713600	0.849400
P3	1.716200	2.574300	0.858100	1.723000	2.588200	0.865200
P4	2.574300	3.432400	0.858100	2.590900	3.441900	0.851000
P1	0.000000	6.006600	6.006600	0.000000	5.976500	5.976500

9. PSJF_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P3	0.171600	1.887800	1.716200	0.179500	1.870900	1.691400
P2	0.000000	5.148600	5.148600	0.000000	5.246300	5.246300
P4	5.148600	12.01330	6.864700	5.246500	11.67470	6.428200
P1	12.01330	24.02660	12.01330	11.67510	23.62500	11.94990

10. PSJF_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.171600	0.171600	0.000000	0.176100	0.176100
P3	0.171600	0.514900	0.343300	0.177200	0.503300	0.326100
P2	0.514900	7.379600	6.864700	0.504900	6.954700	6.449800
P4	7.379600	14.24430	6.864700	6.955000	13.42950	6.474500
P5	14.24430	26.25760	12.01330	13.43010	24.39020	10.96010

11. RR_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.858100	0.858100	0.000000	0.857200	0.857200
P2	0.858100	1.716200	0.858100	0.857900	1.695800	0.837900
P3	1.716200	2.574300	0.858100	1.696300	2.527100	0.830800
P4	2.574300	3.432400	0.858100	2.527500	3.381400	0.853900
P5	3.432400	4.290500	0.858100	3.381800	4.245900	0.864100

12. RR_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	12.87140	12.87140	0.000000	12.96120	12.96120
P2	0.858100	15.44570	14.58760	0.826500	15.72180	14.89530

13. RR_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P3	5.148600	29.17510	24.02650	5.153800	29.03670	23.88290
P1	0.000000	32.60750	32.60750	0.000000	32.54790	32.54790
P2	2.574300	33.46560	30.89130	2.595100	33.41330	30.81820
P6	10.29710	46.33700	36.03990	10.28940	46.23030	35.94090
P5	9.439000	49.76930	40.33030	9.391100	49.78580	40.39470
P4	8.580900	51.48550	42.90460	8.530300	51.47830	42.94800

14. RR_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P4	2.574300	9.439000	6.864700	2.663600	9.680300	7.016700
P5	3.432400	10.29710	6.864700	3.574800	10.55900	6.984200
P6	4.290500	11.15520	6.864700	4.501800	11.40650	6.904700
P3	1.716200	24.88470	23.16850	1.759200	25.28630	23.52710
P7	5.148600	30.89130	25.74270	5.355100	31.56550	26.21040
P2	0.858100	34.32370	33.46560	0.892200	34.97530	34.08310
P1	0.000000	39.47220	39.47220	0.000000	40.20340	40.20340

15. RR_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P4	2.574300	9.439000	6.864700	2.549200	9.357300	6.808100
P5	3.432400	10.29710	6.864700	3.430800	10.18770	6.756900
P6	4.290500	11.15520	6.864700	4.227500	11.02350	6.796000
P3	1.716200	24.88470	23.16850	1.680500	24.54380	22.86330
P7	5.148600	30.89130	25.74270	5.060500	30.49070	25.43020
P2	0.858100	34.32370	33.46560	0.869700	33.85700	32.98730
P1	0.000000	39.47220	39.47220	0.000000	39.08830	39.08830

16. SJF_1

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P2	0.000000	3.432400	3.432400	0.000000	3.327200	3.327200
P3	3.432400	5.148600	1.716200	3.327700	5.075200	1.747500
P4	5.148600	12.01330	6.864700	5.075400	11.60170	6.526300
P1	12.01330	24.02660	12.01330	11.60300	23.57330	11.97030

17. SJF_2

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	0.171600	0.171600	0.000000	0.171200	0.171200
P3	0.171600	0.514900	0.343300	0.173100	0.500500	0.327400
P2	0.514900	7.379600	6.864700	0.501500	7.484900	6.983400
P4	7.379600	14.24430	6.864700	7.485500	13.83640	6.350900
P5	14.24430	26.25760	12.01330	13.83660	26.02790	12.19130

18. SJF_3

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	5.148600	5.148600	0.000000	5.039600	5.039600
P4	5.148600	5.165700	0.017100	5.043000	5.058900	0.015900
P5	5.165700	5.182900	0.017200	5.059900	5.075900	0.016000
P6	5.182900	12.04760	6.864700	5.077600	11.90670	6.829100
P7	12.04760	18.91240	6.864800	11.90710	19.11000	7.202900
P2	18.91240	27.49330	8.580900	19.11050	27.51160	8.401100
P3	27.49330	39.50660	12.01330	27.51270	39.28730	11.77460
P8	39.50660	54.95220	15.44560	39.28750	55.10710	15.81960

19. SJF_4

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	5.148600	5.148600	0.000000	5.167600	5.167600
P2	5.148600	6.864700	1.716100	5.168500	6.865800	1.697300
P3	6.864700	13.72950	6.864800	6.867700	13.65130	6.783600
P5	13.72950	15.44570	1.716200	13.65480	15.30600	1.651200
P4	15.44570	18.87800	3.432300	15.30660	18.65960	3.353000

20. SJF_5

Name	Theo. start	Theo. end	Theo. running	Practical start	Practical end	Practical running
P1	0.000000	3.432400	3.432400	0.000000	3.450900	3.450900
P2	3.432400	4.290500	0.858100	3.451200	4.296800	0.845600
P3	4.290500	5.148600	0.858100	4.297800	5.176600	0.878800
P4	5.148600	6.006600	0.858000	5.178900	6.080000	0.901100

- 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6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 0.0427%，由此可見本模擬方法的高準確度。

(如上所述，最大誤差為 8.7669%，也並未超過10%。)