

作業系統 Project1

第 9 組

R07922118 翁子騰、R07922107 連少甫、R07922105 林東逸

B05611038 張育堂、B04502138 霍家琦、B04502017 王靖棻

一、設計理念

當資料讀進來後，processes 會先依照 ready time 排序，固定將 parent process 指定給 0 號 CPU，而 child processes 指定給 1 號 CPU。一旦時間數到 process 的 ready time，就 fork 一個 child process 開始執行。parent process 每經過一單位的時間，都要檢查 process 有沒有 ready 和可不可以執行。並且每一單位時間都會看有沒有需要做 context switch，來換另一個 process 來執行。

1. FIFO

- 1) 將讀測資檔結果丟進 FIFO()變數中
- 2) 將進程依照 ready time 排列
- 3) 將各進程的 pid 初始化為-1，代表尚未 ready
- 4) 給母進程一個 CPU，避免它被其他進程搶占資源
- 5) 進入 while loop
 - i. 判斷：若有進程剛好執行完成；做：waitpid()收集此子進程 exit 資訊，若為最後一個進程則 break
 - ii. 將當下時間剛好 ready 的子進程全部 fork 出來，紀錄他們的 pid，並將他們全部丟到另一個 CPU，並暫停他們的運行
 - iii. 決定下一個時間單位該做什麼，若需要執行一個新的子進程，就把該進程的權限拉高
 - iv. 執行一個 time unit，若此時有子進程在運行則將該進程的 exec time-1
- 6) 回到 main

2. RR

資料讀進來後，先用 sort 把 process 的 ready time 先排序過一次。當 main 開始跑後，以 while loop 當作每一次的 unit time，每次迴圈中先檢查是否有程序的執行時間歸 0 (process finish)，之後看是否有程序需要 fork 進子程序序列中。因為 RR 可能需要 context switch，所以每次 loop 中會先檢查下一個 time step 中需要>

執行的 process，而此函式中有儲存上一個 context switch 的時間，確認完後，若要 context switch，就將 child cpu 中正在執行的 process 降低 priority，然後讓下一個要執行的程序提高 priority，此迴圈一直進行到所有程序跑完為止。

3. SJF

檢查各 process 的 execution time：

- i. 若有多個 processes 都已經 ready → execution time 短的會先執行
- ii. 若大家 execution time 都相同 → 則用 FIFO。

下一個要執行的 process 用 sched_setscheduler 把他的 priority 排到最優先；而剛生成的 child process 則是先使其 idle。

4. PSJF

當 readytime 等於系統時間時則 fork 一個新的 child process 至 cpu1，並在每次系統時間增加時，選擇一個 execution time 最低的作為下次執行的 process。

二、範例測資結果

1. FIFO

測試：FIFO_1.txt

FIFO

5

P1 0 500

P2 0 500

P3 0 500

P4 0 500

P5 0 500

結果：

```
[50386.885522] [project1] 7160 1556605919.125540544 1556605921.581604033
[50389.342341] [project1] 7161 1556605919.125727046 1556605924.039648492
[50391.737984] [project1] 7162 1556605919.125705249 1556605926.436492706
[50394.085706] [project1] 7163 1556605919.129407093 1556605928.785387597
[50396.394261] [project1] 7164 1556605919.151190387 1556605931.095093462
```

2. RR

測試 : RR_3.txt

RR

6

P1 1200 5000

P2 2400 4000

P3 3600 3000

P4 4800 7000

P5 5200 6000

P6 5800 5000

結果 :

[2005.419856]	[project1]	2452	1556544737.723564672	1556544800.084102358
[2008.537686]	[project1]	2450	1556544724.884969612	1556544803.203493922
[2011.411529]	[project1]	2451	1556544731.539740827	1556544806.078770934
[2051.619281]	[project1]	2455	1556544747.219257127	1556544846.306630577
[2061.552951]	[project1]	2454	1556544744.975531956	1556544856.245265956
[2067.272330]	[project1]	2453	1556544743.583076210	1556544861.967504898

3. SJF

測試 : SJF_5.txt

SJF

4

P1 0 2000

P2 500 500

P3 1000 500

P4 1500 500

結果 :

[34047.512346]	[project1]	5499	1556589563.967873434	1556589574.038733806
[34050.049432]	[project1]	5500	1556589574.039078366	1556589576.577011311
[34052.566048]	[project1]	5501	1556589574.038975110	1556589579.094966313
[34055.084470]	[project1]	5502	1556589579.095186588	1556589581.614647984

4. PSJF

測試 : PSJF_3.txt

PSJF

4

P1 0 2000

P2 500 500

P3 1000 500

P4 1500 500

結果 :

[39760.455927]	[project1]	5918	1556595265.163758384	1556595267.6312500
[39785.311468]	[project1]	5920	1556595289.986332922	1556595292.4583700
[39810.143954]	[project1]	5923	1556595314.854970037	1556595317.3246400
[39883.975830]	[project1]	5915	1556595240.269689428	1556595257.5847600

三、 測資結果分析

1. FIFO：平均 error = -18.3362 (ideal unit time)

理論	Start Time	Finish Time	Finish-Start Time
P1	0	500	500
P2	0	1000	1000
P3	0	1500	1500
P4	0	2000	2000
P5	0	2500	2500

*unit time = 0.004897285

實際	PID	Start Time	Finish Time	Finish-Start Time	Finish-Start Time/Unit Time
P1	7160	1556605919.125540544	1556605921.581604033	2.45606	501.5146
P2	7161	1556605919.125727046	1556605924.039648492	4.91392	1003.397
P3	7162	1556605919.125705249	1556605926.436492706	7.31079	1492.825
P4	7163	1556605919.129407093	1556605928.785387597	9.65598	1971.701
P5	7164	1556605919.151190387	1556605931.095093462	11.94390	2438.882

2. RR：平均誤差 = -1937.714 (ideal unit time)

理論	Start Time	Finish Time	Finish-Start Time
P1	1200	18700	17500
P2	2400	19200	16800
P3	3600	18200	14600
P4	4800	31200	26400
P5	5200	30200	25000
P6	5800	28200	22400

實際	PID	Start Time	Finish Time	Finish-Start Time	Finish-Start Time/Unit Time
P1	2450	1556544724.884969612	1556544803.203493922	78.31853	15992.23331
P2	2451	1556544731.539740827	1556544806.078770934	74.53903	15220.48035
P3	2452	1556544737.723564672	1556544800.084102358	62.36054	12733.69585
P4	2453	1556544743.583076210	1556544861.967504898	118.3844	24173.48182
P5	2454	1556544744.975531956	1556544856.245265956	111.2697	22720.69810
P6	2455	1556544747.219257127	1556544846.306630577	99.08738	20233.12375

3. SJF : 平均 error = 155.3795113 (ideal unit time)

理論	Start Time	Finish Time	Finish-Start Time
P1	0	2000	2000
P2	2000	2500	500
P3	2500	3000	500
P4	3000	3500	500

實際	PID	Start Time	Finish Time	Finish-Start Time	Finish-Start Time/Unit Time
P1	5499	1556589563.967873434	1556589574.038733806	10.07085991	2056.416815
P2	5500	1556589574.039078366	1556589576.577011311	2.537940025	518.2340526
P3	5501	1556589574.038975110	1556589579.094966313	5.055989981	1032.406657
P4	5502	1556589579.095186588	1556589581.614647984	2.519459963	514.4605207

4. PSJF : 平均 error = 12.14320318 (ideal unit time)

理論	Start Time	Finish Time	Finish-Start Time
P1	0	3500	3500
P2	500	1000	500
P3	1000	1500	500
P4	1500	2000	500

實際	PID	Start Time	Finish Time	Finish-Start Time	Finish-Start Time/Unit Time
P1	5915	1556595240.2696800	1556595257.5847600	17.3150799	3535.649
P2	5918	1556595265.1637500	1556595267.6312500	2.4675050	503.8516
P3	5920	1556595289.9863300	1556595292.4583700	2.4720440	504.7785
P4	5923	1556595314.8549700	1556595317.3246400	2.4696710	504.2939

誤差成因：

- 1) 從 SJF 和 PSJF 的數據可以看到其實際值較理論值大，其原因在於實際排程時不同 process 間產生了 context switch，導致了執行時間的增加。另外，在執行測資時背景環境中可能尚有其他 priority 較高的(ex. System call)在執行，導致測資時間的增加。
- 2) FIFO 和 RR 的實際值較理論值小，推測其原因為 Unit time 在測試時可能沒有

考慮到每個人寫的 main 不一樣，因此造成些微的時間差異，而當執行時間變長時此差異就明顯了起來。而 RR 在執行時時間消耗越來越短，除了 Unit time 的誤差以外，因為 RR 演算法在設計時，每一次預測下一個 time step 要跑的 function 中，有一個要跑過所有存在程序的 for loop，而當程序執行到越後面時，因為先執行完的 process 已經消失了，此時 for loop 要跑得 iteration 變少了，再加上 context switch 時也需要時間，而程式執行到越後面要 context switch 的次數也越來越少，因此在長時間的執行下，這些時間差異就顯現出來了。

四、 分工

FIFO	霍家琦
RR	張育堂
SJF	翁子騰
PSJF	連少甫/林東逸
程式統整+測資	翁子騰/林東逸
Report 彙整	王靖棻