

# 2020 OS Project 1 - Process Scheduling

B06902049 林首志

## 核心版本

我使用的核心是 Linux 5.6.7

## 設計

我使用`sched_setscheduler`和`sched_setaffinity`作為控制排程的方法。`sched_setscheduler`固定使用`SCHED_FIFO`，每次我會將需要運行的程序的priority設為97 (High Priority)，將下次（可能）要運行的程序的priority設為50 (Medium Priority)，將其餘程序的priority設為3 (Low Priority)。為了讓新fork出來的程序不要影響到其他正在運行的程序，控制排程的主程序的Priority固定設為3，這樣fork出來的子程序就會在最先開始的時候設成Low Priority。使用三種Priority Level的原因是，當前正在運行的程序跑完的時候，排程主程序沒辦法那麼快判斷出下個需要運行的程序是那一個。如果只用兩種Level，在High Priority的程序運行完的瞬間，可能會不小心跑到理論上不該運行的程序。因此使用三種Level，將下次可能要運行的程序先找出來可以增加準確度。控制排程的主程序固定在CPU 0執行，而其他運行的子程序固定在CPU 1執行。

由於四種排程演算法有一定的共通性，我盡量讓他們的實做結合在一起。我的排程演算法會使用一個用circular array實做的queue，存放正在運行的processes。這個queue可以作為簡單的FIFO queue，適合給FIFO和RR使用，也可以用naive的 $O(N)$ 時間複雜度找到queue中的最小值及次小值（比較方法是先依執行時間為基準，如果執行時間一樣則比較其輸入順序），適合給SJF和PSJF使用，因此四種排程演算法我都使用同樣的資料結構。

當主程序讀完輸入之後，就會進入一個大迴圈，每跑完一次主程序就會經過一個unit time。這一個大迴圈主要會依序做五件事情：一、將達到ready time的程序fork出來，並放進queue。二、當有新程序被fork出來的時候，或者是有程序跑完的時候，重新選擇要運行的程序，還有下次會運行的程序。三、跑一個unit time。四、如果當前的程序運行完了，則將程序pop出queue，並waitpid該子程序。（我會在主程式中對每個子程序紀錄其剩下的時間。經過我實際測試發現，相較於主程式不紀錄剩下時間，只用SIGCHLD判斷子程序是否運行結束的作法，由主程序紀錄時間並判斷子程序是否即將結束，實驗結果會比較準確。）五、如果是RR排程演算法，則判斷當前程序是否已經跑完了一個time quantum，如果是的話則切換程序運行。跑完numProcess個程序之後，主程序就會離開這個大迴圈，並正常的結束。

以下我將對四種排程演算法會經過的過程做更詳細的說明。當演算法是FIFO時，我會將fork出來的程序push進queue的尾端，每次在挑選要執行的程式時，當前要執行的程式即是queue的head，而下次要執行的程式即是queue的head的下一個元素（如果存在的話）。

當演算法是RR時，大致上和FIFO一樣，只是當程序跑完一個time quantum時，就會將queue的head pop出來，再放進queue的尾端，並重新選擇要跑的程序。

當演算法是SJF時，我一樣會將fork出來的程序放進queue的尾端，每次在挑選要執行的程序時，我會找出最小值和次小值。如果當前沒有運行的程序，則我會依照最小值和次小值決定這次要跑和下次要跑的程序，並將queue的最小值swap的queue的head，這樣可以讓下次的pop能正確的pop掉跑完的程式。若當前有正在運行的程序，如果最小值和當前運行的程序不同，則我會依照最小值來重新決定下次要跑的程序，否則我會依照次小值重新決定下次要跑的程序。

當演算法是PSJF時，實做方法和SJF類似，只是每次都會重新依照queue的最小值和次小值決定現在要跑和下次要跑的程序，就算現在正在運行的程序還沒跑完也一樣。我一樣會讓queue的最小值保持在queue的head，方便pop掉跑完的程序。

在dmesg裡，我的開始時間是紀錄程序開始運行的時間，而不是剛fork出來的時間。

### 一些實作要注意的小細節

由於我的kernel版本比較新，因此實作syscall的方法和HW1的投影片有不少的差異。我的作法是參考這篇stackoverflow的解答：<https://stackoverflow.com/questions/53735886/how-to-pass-parameters-to-linux-system-call>。另外，project網頁上提到的getnstimeofday在比較新的核心上是無法使用的deprecated function，因此我參考了這篇文件：<https://www.kernel.org/doc/html/latest/core-api/timekeeping.html>，使用ktime\_get\_real\_ts64這個類似的function達到同樣的效果。

### 結果分析

為了分析我的程式與理論解的差異，我寫了一個C++的程式來算出理論解。其實作的方式主要是使用C++內建的queue和priority\_queue對整數做操作，計算出理論解。

接下來，我寫了一個python script，用來分析程式的運行結果，其中一次的運行結果我列在最底下的那些表格。由於我程式的寫法常常會受誤差影響，每次跑的結果可能都有些微的差異，但通常來說運行結果離理論解都十分接近。

我先用TIME\_MEASUREMENT.txt算出Unit Time，我的Unit Time是0.0017422656059265136 ns。在接下來的表格，我會列出每個檔案，每個程序的名字(Name)、理論運行時間長度(TD)（單位是Unit Time）、實際運行時間長度(RD)（單位是Unit Time）、與理論的差值(Error)以及與理論的相對誤差(RError)（實際時間減理論時間，再除以理論時間）。

觀察這些表格的結果，可以發現大部分的程序運行結果和理論值都有些微的差異，而且通常運行時間是比理論久的。這可能是因為排程式在運行的時候有許多overhead，需要處理很多其他的運算，導致排程的結果不夠準確。也有可能是因為每次跑unit time的迴圈時花費的時間都有一些差異，造成結果的誤差。另外，排程式計算的時間和子程式計算的時間也可能會不同，由於之間沒有同步機制，多少會造成一些差異。

#### FIFO\_1.txt

平均誤差（取絕對值再取平均）：3.809326180735127

平均相對誤差（取絕對值再取平均）：0.007618652361470255

Name	TD	RD	Error	RError
P1	500	506.516	6.515717161862199	0.013031434323724398
P2	500	503.590	3.5902663197543916	0.0071805326395087835
P3	500	503.880	3.8795545169765546	0.0077591090339531096
P4	500	502.943	2.9427205537868417	0.005885441107573683
P5	500	502.118	2.1183723512956476	0.004236744702591295

#### FIFO\_2.txt

平均誤差（取絕對值再取平均）：51.23359657920696

平均相對誤差（取絕對值再取平均）：0.0018298299538250377

Name	TD	RD	Error	RError
P1	80000	79803.328	-196.67218403393053	-0.0024584023004241315
P2	5000	4995.748	-4.251605960026609	-0.0008503211920053218
P3	1000	1003.890	3.8899820290997695	0.0038899820290997696
P4	1000	1000.121	0.12061429377092736	0.00012061429377092736

#### FIFO\_3.txt

平均誤差（取絕對值再取平均）：10.636078004922151

平均相對誤差（取絕對值再取平均）：0.0032904450450444564

Name	TD	RD	Error	RError
P1	8000	7981.252	-18.747736258276746	-0.002343467032284593
P2	5000	5032.084	32.083757937430164	0.006416751587486033
P3	3000	3009.024	9.024478051017468	0.003008159350339156
P4	1000	1001.061	1.0611430447207795	0.0010611430447207794
P5	1000	1004.112	4.111806138701013	0.004111806138701013
P6	1000	1004.981	4.981176014269863	0.004981176014269863
P7	4000	4004.442	4.442448590039021	0.0011106121475097552

#### FIFO\_4.txt

平均誤差（取絕對值再取平均）：47.34432203354114

平均相對誤差（取絕對值再取平均）：0.08105942182100057

Name	TD	RD	Error	RError
P1	2000	2077.601	77.6007788941024	0.0388003894470512
P2	500	543.525	43.525037715572466	0.08705007543114493
P3	200	220.628	20.628093118942246	0.10314046559471123
P4	500	547.623	47.62337840554744	0.09524675681109489

#### FIFO\_5.txt

平均誤差（取絕對值再取平均）：58.766991224382615

平均相對誤差（取絕對值再取平均）：0.013325298649173376

Name	TD	RD	Error	RError
P1	8000	8208.754	208.75375684651044	0.026094219605813804
P2	5000	5067.899	67.89884098065886	0.013579768196131771
P3	3000	3084.450	84.45001223111603	0.028150004077038677
P4	1000	1000.652	0.6517058232975614	0.0006517058232975615
P5	1000	1007.000	7.000035415224943	0.007000035415224943
P6	1000	1009.530	9.530280810985687	0.009530280810985687
P7	4000	4033.084	33.0843064628848	0.008271076615721198

### PSJF\_1.txt

平均誤差（取絕對值再取平均）：18.47443143921805

平均相對誤差（取絕對值再取平均）：0.0013939090867275267

Name	TD	RD	Error	RError
P1	25000	25038.120	38.120083229612646	0.0015248033291845057
P2	15000	15016.770	16.769957423455708	0.001117997161563714
P3	8000	8016.335	16.334684056509104	0.002041835507063638
P4	3000	3002.673	2.673001047294747	0.000891000349098249

### PSJF\_2.txt

平均誤差（取絕對值再取平均）：4.84519140341663

平均相對誤差（取絕對值再取平均）：0.0024649598644784416

Name	TD	RD	Error	RError
P1	4000	3999.589	-0.41086039893070847	-0.00010271509973267712
P2	1000	995.968	-4.0316429287044	-0.0040316429287044
P3	7000	7012.586	12.586335558215978	0.001798047936887997
P4	2000	1998.391	-1.6094495483298488	-0.0008047247741649244
P5	1000	1005.588	5.587668582902211	0.0055876685829022105

### PSJF\_3.txt

平均誤差（取絕對值再取平均）：5.232455792959243

平均相對誤差（取絕對值再取平均）：0.005351125924599156

Name	TD	RD	Error	RError
P1	3500	3511.932	11.932166543078438	0.0034091904408795537
P2	500	498.814	-1.1862731901985057	-0.0023725463803970113
P3	500	503.432	3.432074666022686	0.006864149332045372
P4	500	504.379	4.3793087725373425	0.008758617545074685

#### PSJF\_4.txt

平均誤差（取絕對值再取平均）：7.424764952655522

平均相對誤差（取絕對值再取平均）：0.001975162996767581

Name	TD	RD	Error	RError
P2	3000	3010.990	10.989968295436938	0.003663322765145646
P3	1000	998.196	-1.8035490653309125	-0.0018035490653309125
P4	4000	4000.175	0.1745581950694941	4.363954876737352e-05
P1	7000	6983.269	-16.730984254784744	-0.002390140607826392

#### PSJF\_5.txt

平均誤差（取絕對值再取平均）：47.897965179116014

平均相對誤差（取絕對值再取平均）：0.020275809181391997

Name	TD	RD	Error	RError
P1	100	97.557	-2.4432345124486687	-0.024432345124486686
P2	4000	4209.476	209.47634785720675	0.05236908696430169
P3	200	204.085	4.085433564732881	0.020427167823664404
P4	4000	4007.424	7.424416000478232	0.001856104000119558
P5	7000	7016.060	16.060393960713554	0.0022943419943876504

#### RR\_1.txt

平均誤差（取絕對值再取平均）：15.089940165785947

平均相對誤差（取絕對值再取平均）：0.030179880331571894

Name	TD	RD	Error	RError
P1	500	499.519	-0.481389791952779	-0.000962779583905558
P2	500	512.415	12.41465108696002	0.024829302173920043
P3	500	509.076	9.076341923651057	0.018152683847302113
P4	500	514.865	14.86552696787271	0.029731053935745422
P5	500	538.612	38.61179105849317	0.07722358211698634

### RR\_2.txt

平均誤差 ( 取絕對值再取平均 ) : 25.31973735368547

平均相對誤差 ( 取絕對值再取平均 ) : 0.003146872469382277

Name	TD	RD	Error	RError
P1	7500	7478.570	-21.430179540958306	-0.0028573572721277743
P2	8500	8470.791	-29.20929516641263	-0.00343638766663678

### RR\_3.txt

平均誤差 ( 取絕對值再取平均 ) : 509.22395097540385

平均相對誤差 ( 取絕對值再取平均 ) : 0.027877791119877444

Name	TD	RD	Error	RError
P1	18000	19539.499	1539.4986222273292	0.08552770123485162
P2	17500	17524.814	24.813647221337305	0.0014179226983621316
P3	14000	14508.482	508.48224742061393	0.03632016053004385
P4	25000	25283.002	283.0021316462189	0.011320085265848757
P5	23000	23352.126	352.12630109214297	0.01530983917791926
P6	20000	20347.421	347.42075624478093	0.017371037812239046

### RR\_4.txt

平均誤差 ( 取絕對值再取平均 ) : 49.23817268918791

平均相對誤差 ( 取絕對值再取平均 ) : 0.004942440090160768

Name	TD	RD	Error	RError
P1	23000	23095.248	95.24782419422627	0.0041412097475750555
P2	19500	19558.987	58.986985534331325	0.0030249736171451963
P3	13500	13552.522	52.52173643639435	0.00389049899528847
P4	4000	4018.450	18.450483740771688	0.004612620935192922
P5	4000	4026.173	26.173274379482336	0.0065433185948705844
P6	4000	4033.629	33.62908239152284	0.00840727059788071
P7	15000	15059.658	59.65782214758656	0.003977188143172437

### RR\_5.txt

平均誤差 ( 取絕對值再取平均 ) : 53.7168810270443

平均相對誤差 ( 取絕對值再取平均 ) : 0.004286698198625894

Name	TD	RD	Error	RError
P1	23000	23079.395	79.39485835524829	0.0034519503632716646
P2	19500	19600.761	100.7608033267752	0.005167220683424369
P3	13500	13574.925	74.92542413016054	0.005550031417048929
P4	4000	4008.893	8.89329940113248	0.00222332485028312
P5	4000	4011.263	11.262753107330354	0.0028156882768325888
P6	4000	4022.254	22.254199317871098	0.005563549829467775
P7	15000	15078.527	78.52682955079217	0.005235121970052811

### SJF\_1.txt

平均誤差（取絕對值再取平均）：28.234528048885693

平均相對誤差（取絕對值再取平均）：0.011622183135842339

Name	TD	RD	Error	RError
P2	2000	2061.252	61.25202727530632	0.03062601363765316
P3	1000	1005.010	5.010186940386802	0.005010186940386802
P4	4000	4039.056	39.055767703274796	0.0097639419258187
P1	7000	7007.620	7.620130276574855	0.0010885900395106935

### SJF\_2.txt

平均誤差（取絕對值再取平均）：49.9082767570099

平均相對誤差（取絕對值再取平均）：0.034317034278239046

Name	TD	RD	Error	RError
P1	100	108.504	8.504284776943479	0.08504284776943478
P2	4000	4212.897	212.89662657119516	0.05322415664279879
P3	200	205.893	5.893142687393748	0.02946571343696874
P4	4000	4006.293	6.293126725910952	0.001573281681477738
P5	7000	7015.954	15.95420302360617	0.002279171860515167

### SJF\_3.txt

平均誤差（取絕對值再取平均）：19.39681412956901

平均相對誤差（取絕對值再取平均）：0.006017675370377195

Name	TD	RD	Error	RError
P1	3000	3004.226	4.2259066584947504	0.0014086355528315836
P4	10	10.096	0.09593913266855303	0.009593913266855303
P5	10	10.101	0.10072867235766125	0.010072867235766125
P6	4000	4031.718	31.717919211577282	0.00792947980289432
P7	4000	4020.138	20.138043839231614	0.005034510959807903
P2	5000	5023.450	23.4498600058605	0.0046899720011721
P3	7000	7032.417	32.41735621897624	0.004631050888425177
P8	9000	9043.029	43.02875929738548	0.004780973255265053

#### SJF\_4.txt

平均誤差（取絕對值再取平均）：7.377943781094655

平均相對誤差（取絕對值再取平均）：0.004206774043949925

Name	TD	RD	Error	RError
P1	3000	2992.929	-7.0709753402202296	-0.00235699178007341
P2	1000	999.708	-0.2918334954568991	-0.00029183349545689906
P3	4000	3987.687	-12.312729682375448	-0.003078182420593862
P5	1000	1013.400	13.399544659830212	0.013399544659830212
P4	2000	2003.815	3.8146357275904847	0.0019073178637952423

#### SJF\_5.txt

平均誤差（取絕對值再取平均）：4.107570817175926

平均相對誤差（取絕對值再取平均）：0.006053783164346527

Name	TD	RD	Error	RError
P1	2000	1994.236	-5.763622586680867	-0.0028818112933404337
P2	500	504.098	4.097546994826587	0.008195093989653174
P3	500	503.737	3.737099922223365	0.0074741998444467295
P4	500	502.832	2.832013764972885	0.00566402752994577

#### TIME\_MEASUREMENT.txt



平均誤差（取絕對值再取平均）：2.4232279209473004

平均相對誤差（取絕對值再取平均）：0.0048464558418946

Name	TD	RD	Error	RError
P0	500	505.743	5.743095988013351	0.011486191976026704
P1	500	498.546	-1.4539400368241786	-0.0029078800736483572
P2	500	496.319	-3.6813496802420786	-0.007362699360484158
P3	500	498.859	-1.1411146731297777	-0.0022822293462595555
P4	500	497.854	-2.14623378788707	-0.00429246757577414
P5	500	497.564	-2.4363430490560063	-0.004872686098112013
P6	500	503.341	3.3407997239473843	0.006681599447894769
P7	500	501.054	1.053589256411101	0.0021071785128222017
P8	500	501.979	1.9786546363647517	0.003957309272729503
P9	500	498.743	-1.2571583775973068	-0.0025143167551946134

所有檔案的平均誤差之平均：50.863946926749804

所有檔案的平均相對誤差之平均：0.013561787354342585