OS Project 1 Report
第四組

1. 設計：

整體架構設計：

從 stdin 讀進 process，讓 parent 的 process 跑 core2，fork 出來的 child process 跑 core1，所以 parent process 不會佔用 child process 的資源 (core1)，c hild process再據排程的需求 (i.e: FIFO, SJF, PSJF, RR) 來計算

FIFO 設計方法：

根據 ready time 從小到大排列，排在最前頭的有最大的 priority，起床之後parent process fork 一個child，接者call exec，執行time.out這個程式，執行所要求的時間，然後以此類推到最後一個 process。

SJF 設計方法:

首先按ready time從小到大排列，如果 ready time 一樣，執行時間的順序照 execution time 從小到大排列， 接者我們模擬程式的進行(先排好再做事)，把每個 process 進行的順序排出來，進行方法如下，第一個 process 所需要的時間是他的 ready time 加 execution time (finish_time = R[0] + T[0])，若後面有一些的 process 的 ready time 小於等於第一個 process 的 ready time 加 execution time，就要從這些 process 的 execution time 去排順序，時間小的順位排前面，第二個 process 的 R[P[1]] 加上 finish_time，再去看後面哪些 process 的 ready time 小於 finish_time，再去排順序，以此類推做到最後，就可以得到 process 進行的順序。

```
int finish_time = R[0];
for(int i = 0; i < p_num; i++){
    point = i;
    finish_time += T[P[i]];
    for(int j = i ; j < p_num; j++){
        if(R[P[j]] > finish_time) break;
            point++;
    }
    for(int j = point - 1; j >= i + 1; j--)
        for(int k = i + 1; k < j; k++)
            if(T[P[k]] > T[P[k+1]]) swap(&P[k],&P[k+1]);
 }
```

PSJF 設計方法:

會被搶先的 priority 排低一點，當做到有 process 在 ready queue 裏面的時候，priority 高的 process 就會搶先，目前的 process 就會被踢出去，先把拿到的排程資料做排序，排序的方式是每一個 process 來的時間點去算說哪個 child process 的 priority 最高，再來才 fork child process。

RR 設計方法:

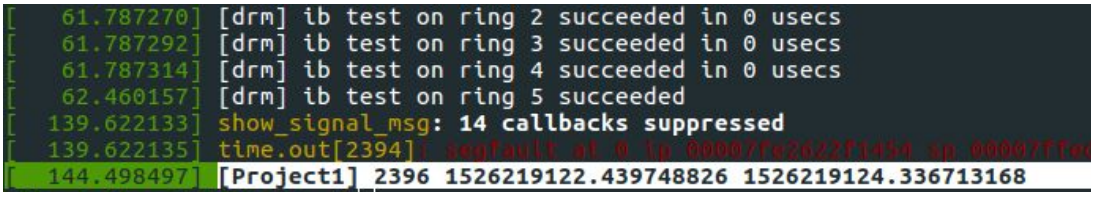
依照FIFO的做法，只是要加入記時間的變數 (time_count) ，從該process開始時間算起 (令作start_time，每改變一次process的優先序，start_time 就會變成現在的 time_count) ，每到 500 時間單位就要兌換所有正在執行階段的 process 的 priority ，把剛剛執行的 process 優先序改到最後，其餘的 priority 往前一位。至於記執行中的 process，我們是用 queue 去記，當有 process 執行完成，就把該 process 從 queue 中排出，queue 中的順序代表該 process 的 priority，所以要換 priority 時，process 的優先序是在 queue 換。

2. 執行範例測資的結果

```
$ time ./time.out 1000

real      0m1.898s
user      0m1.897s
sys       0m0.000s
```

```
$ dmesg
```



-1526219122.439748826 +1526219124.336713168 = 1.896964342s

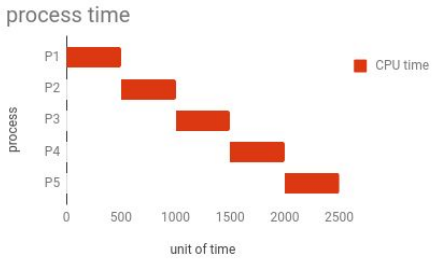
1000 units 跑 1.896 秒

・FIFO

Testcase 1

```
FIFO
5
P1  0  500
P2  0  500
P3  0  500
P4  0  500
P5  0  500
```

理論值:



理論總時間：2500 / 1000 * 1.896 = 4.74s

實際值:

```
$ sudo ./schedule < ./OS_PJ1_Test/FIFO_1.txt
$ dmesg
```

```
[ 4106.322757] [Project1] 4175 1526223085.194522191 1526223086.197902491
[ 4107.326794] [Project1] 4176 1526223086.198402508 1526223087.201949245
[ 4108.330625] [Project1] 4177 1526223087.202452199 1526223088.205791408
[ 4109.334419] [Project1] 4178 1526223088.206310321 1526223089.209593344
[ 4110.338198] [Project1] 4179 1526223089.210093476 1526223090.213382519
```

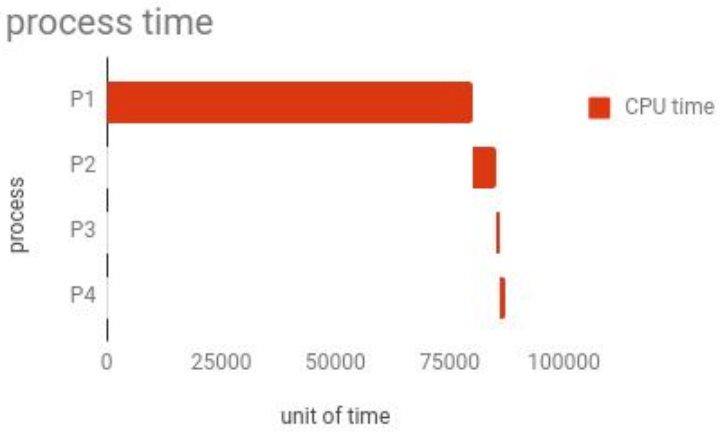
```
pid:4175        -1526223085.194522191 + 1526223086.197902491 = 1.0033803
pid 4176        -1526223086.198402508 + 1526223087.201949245 = 1.003546737
pid 4177        -1526223087.202452199 + 1526223088.205791408 = 1.003339209
pid 4178        -1526223088.206310321 + 1526223089.209593344 = 1.003283023
pid 4179        -1526223089.210093476 + 1526223090.213382519 = 1.003289043
```

實際總時間：1.0033803 + 1.003546737 + 1.003339209 + 1.003283023 + 1.003289043 = 5.016838312s

Testcase 2

```
FIFO
4
P1 0 80000
P2 100 5000
P3 200 1000
P4 300 1000
```

理論值:



理論總時間：87000 / 1000 * 1.896 = 164.952s

實際值:

```
$ sudo ./schedule < ./OS_PJ1_Test/FIFO_2.txt
$ dmesg
```

```
[ 8623.833579] perf: interrupt took too long (2511 > 2500), lowering kernel.perf_event_max_sample_rate to 79500
[ 8645.135756] [Project1] 5966 1526227464.422861513 1526227625.47870329
[ 8655.212283] [Project1] 5967 1526227625.48378737 1526227635.124557393
[ 8657.242749] [Project1] 5968 1526227635.125079886 1526227637.155055502
[ 8659.265249] [Project1] 5969 1526227637.155553520 1526227639.177587301
```

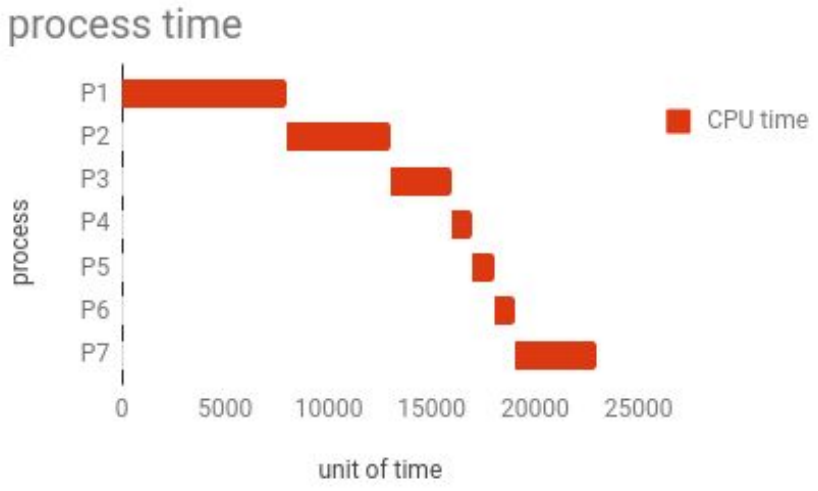
pid:5966 -1526227464.422861513 + 1526227625.47870329 = 161.055841777
pid:5967 -1526227625.48378737 + 1526227635.124557393 = 9.640770023
pid:5968 -1526227635.125079886 + 1526227637.155055502 = 2.029975616
pid:5969 -1526227637.155553520 + 1526227639.177587301 = 2.022033781

實際總時間：161.055841777 + 9.640770023 + 2.029975616 + 2.022033781 = 174.748621197

Testcase 3

```
FIFO
7
P1 0 8000
P2 200 5000
P3 300 3000
P4 400 1000
P5 500 1000
P6 500 1000
P7 600 4000
```

理論值:



理論總時間：23000/1000 * 1.896 = 43.608s

實際值:

```
$ sudo ./schedule < ./OS_PJ1_Test/FIFO_2.txt
$ dmesg
```

```
[35870.861401] [Project1] 13333 1526257152.415356650 1526257168.492123564
[35880.908185] [Project1] 13334 1526257168.492664626 1526257178.538906794
[35886.938456] [Project1] 13335 1526257178.539417475 1526257184.569176517
[35888.957024] [Project1] 13336 1526257184.569844970 1526257186.587746341
[35890.966493] [Project1] 13337 1526257186.588286774 1526257188.597214439
[35892.975615] [Project1] 13338 1526257188.597754593 1526257190.606337190
[35901.012407] [Project1] 13339 1526257190.606865122 1526257198.643129684
```

pid:13333 -1526257152.415356650 + 1526257168.492123564 = 16.076766914
pid:13334 -1526257168.492664626 + 1526257178.538906794 = 10.046242168
pid:13335 -1526257178.539417475 + 1526257184.569176517 = 6.029759042
pid:13336 -1526257184.569844970 + 1526257186.587746341 = 2.017901371
pid:13337 -1526257186.588286774 + 1526257188.597214439 = 2.008927665
pid:13338 -1526257188.597754593 + 1526257190.606337190 = 2.008582597
pid:13339 -1526257190.606865122 + 1526257198.643129684 = 8.036264562

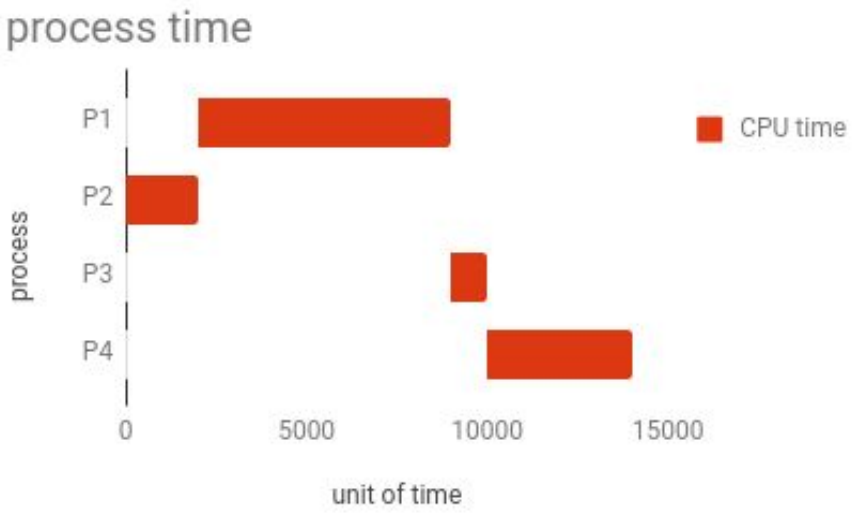
實際總時間：16.076766914 + 10.046242168 + 6.029759042 + 2.017901371 + 2.008927665 + 2.008582597 + 8.036264562 = 46.224444319

・SJF

Testcase 1

```
SJF
4
P1 0 7000
P2 0 2000
P3 100 1000
P4 200 4000
```

理論值:



理論總時間：14000/1000 * 1.896 = 26.544

實際值:

```
$ sudo ./schedule < ./OS_PJ1_Test/SJF_1.txt
$ dmesg
```

```
[57544.936473] [Project1] 23206 1526285011.578023087 1526285015.599228577
[57546.945252] [Project1] 23208 1526285015.599770069 1526285017.608008178
[57554.977766] [Project1] 23209 1526285017.608527321 1526285025.640521945
[57569.035112] [Project1] 23207 1526285025.641043253 1526285039.697869061
```

pid:23206 -1526285011.578023087 + 1526285015.599228577 = 4.02120549
pid:23208 -1526285015.599770069 + 1526285017.608008178 = 2.008238109
pid:23209 -1526285017.608527321 + 1526285025.640521945 = 8.031994624

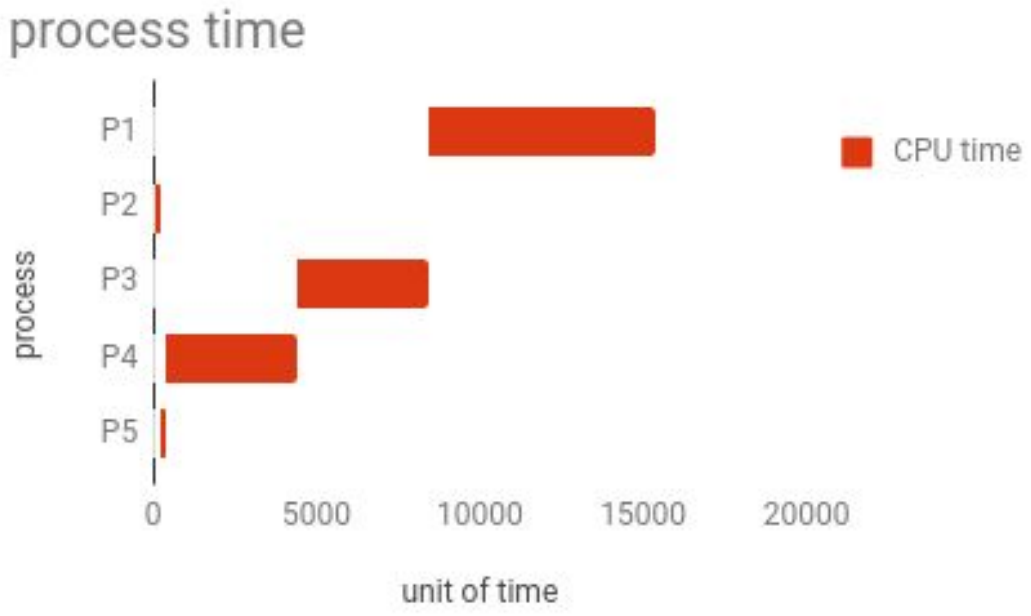
pid:23207 -1526285025.641043253 + 1526285039.69786906 = 14.056825807

實際總時間：4.02120549 + 2.008238109 + 8.031994624 + 14.056825807 = 28.11826403

Testcase 2

```
SJF
5
P1 200 7000
P2 100 100
P3 200 4000
P4 100 4000
P5 200 200
```

理論值



理論總時間：15300/1000 * 1.896 = 29.0088

實際值

```
$ sudo ./schedule < ./OS_PJ1_Test/SJF_2.txt
$ dmesg
```

```
[58639.044553] [Project1] 23691 1526286110.269406434 1526286110.470578709
[58639.446758] [Project1] 23693 1526286110.471041750 1526286110.872784627
[58647.478770] [Project1] 23692 1526286110.873275046 1526286118.904796767
[58655.511147] [Project1] 23694 1526286118.905296688 1526286126.937173053
[58669.569848] [Project1] 23695 1526286126.937728432 1526286140.995874403
```

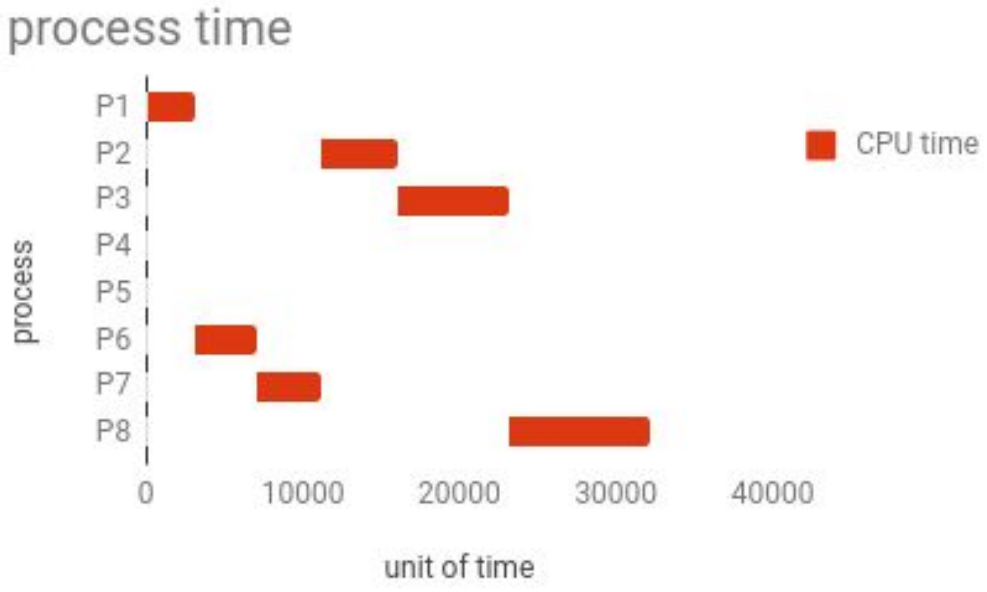
pid:23691 -1526286110.269406434 + 1526286110.470578709 = 0.201172275
pid:23693 -1526286110.471041750 + 1526286110.872784627 = 0.401742877
pid:23692 -1526286110.873275046 + 1526286118.904796767 = 8.031521721
pid:23694 -1526286118.905296688 + 1526286126.937173053 = 8.031876365
pid:23695 -1526286126.937728432 + 1526286140.995874403 = 14.058145971

實際總時間：0.201172275 + 0.401742877 + 8.031521721 + 8.031876365 + 14.058145971 = 30.724459209

Testcase 3

```
SJF
8
P1 100 3000
P2 100 5000
P3 100 7000
P4 200 10
P5 200 10
P6 300 4000
P7 400 4000
P8 500 9000
```

理論值:



理論總時間：32020/1000 * 1.896 = 60.89952

實際值:

```
$ sudo ./schedule < ./OS_PJ1_Test/SJF_3.txt
$ dmesg
```

```
[70644.457369] [Project1] 29437 1526304041.201440297 1526304047.227684490
[70644.478244] [Project1] 29440 1526304047.228222693 1526304047.248558117
[70644.498953] [Project1] 29441 1526304047.249077043 1526304047.269266986
[70652.537208] [Project1] 29442 1526304047.269845907 1526304055.307522928
[70660.574007] [Project1] 29443 1526304055.308078661 1526304063.344321715
[70670.614085] [Project1] 29438 1526304063.344831422 1526304073.384399651
[70684.680083] [Project1] 29439 1526304073.384916901 1526304087.450398045
[70702.774421] [Project1] 29444 1526304087.450900419 1526304105.544735923
```

```
pid:29437       -1526304041.201440297 + 1526304047.227684490 = 6.026244193
pid:29440       -1526304047.228222693 + 1526304047.248558117 = 0.020335424
pid:29441       -1526304047.249077043 + 1526304047.269266986 = 0.020189943
pid:29442       -1526304047.269845907 + 1526304055.307522928 = 8.037677021
pid:29443       -1526304055.308078661 + 1526304063.344321715 = 8.036243054
pid:29438       -1526304063.344831422 + 1526304073.384399651 = 10.039568229
pid:29439       -1526304073.384916901 + 1526304087.450398045 = 14.065481144
pid:29444       -1526304087.450900419 + 1526304105.544735923 = 18.093835504
```

實際總時間：6.026244193 + 0.020335424 + 8.037677021 + 8.036243054 + 10.039568229 + 14.065481144 + 18.093835504 = 64.319384569s

・PSJF

Testcase1

```
PSJF
4
P1 0 10000
P2 1000 7000
P3 2000 5000
P4 3000 3000
```

理論值:

理論總時間：25000/1000 * 1.896 = 47.4s

實際值:

```
$ sudo ./schedule < ./OS_PJ1_Test/SJF_1.txt
$ dmesg
```

```
[116932.004031] [Project1] 19030 1526403067.164162720 1526403073.194791151
[116940.059479] [Project1] 19029 1526403065.154495807 1526403081.250240101
[116952.111604] [Project1] 19027 1526403063.145507626 1526403093.302364313
[116970.194835] [Project1] 19026 1526403061.132798456 1526403111.385594801
```

pid:19026 -1526403061.132798456 + 1526403111.385594801 = 50.252796345s

實際總時間：25000/1000 * 1.896 = 47.4s

Testcase2

```
PSJF
5
P1 0 3000
P2 1000 1000
P3 2000 4000
P4 5000 2000
P5 7000 1000
```

理論值：

理論總時間：11000/1000 * 1.896 = 20.856s

實際值：

```
$ sudo ./schedule < ./OS_PJ1_Test/SJF_1.txt
$ dmesg
```

```
[117783.870713] [Project1] 19356 1526403923.53118498 1526403925.61473352
[117787.890243] [Project1] 19355 1526403921.44224612 1526403929.81003823
[117793.946196] [Project1] 19361 1526403931.108076685 1526403935.136957407
[117795.955742] [Project1] 19366 1526403935.137483523 1526403937.146502303
[117802.005022] [Project1] 19357 1526403929.81507799 1526403943.195782837
```

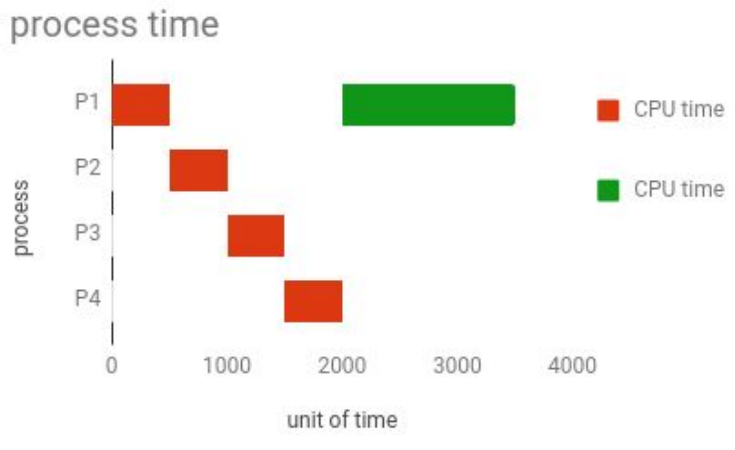
pid19355: -1526403921.44224612 + 1526403929.81003823 = 8.36779211
pid19357: -1526403929.81507799 + 1526403943.195782837 = 13.380704847

實際總時間：8.36779211 + 13.380704847 = 21.748496957s

Testcase3

```
PSJF
4
P1 0 2000
P2 500 500
P3 1000 500
P4 1500 500
```

理論值：



理論總時間：3500/1000 * 1.896 = 6.636s

實際值：

```
$ sudo ./schedule < ./OS_PJ1_Test/SJF_1.txt
$ dmesg
```

```
[116046.075079] [Project1] 18616 1526402186.261530210 1526402187.265839383
[116047.079893] [Project1] 18617 1526402187.266377588 1526402188.270652473
[116048.085083] [Project1] 18618 1526402188.271142137 1526402189.275843483
[116051.099080] [Project1] 18615 1526402185.253798491 1526402192.289840967
```

pid:18615 -1526402185.253798491 + 1526402192.289840967 = 7.036042476s

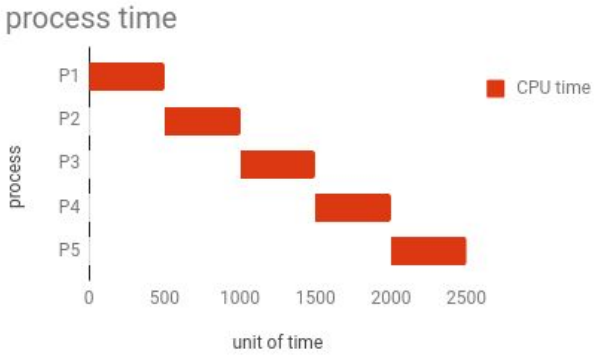
實際總時間：7.036042476s

- RR

Testcase1

```
RR
5
P1 0 500
P2 0 500
P3 0 500
P4 0 500
P5 0 500
```

理論值：



理論總時間：2500/1000 * 1.896 = 4.74s

實際值：

```
$ sudo ./schedule < ./OS_PJ1_Test/RR_1.txt
$ dmesg
```

```
[22512.703244] [Project1] 9791 1526442510.462989308 1526442511.480980999
[22513.809380] [Project1] 9792 1526442511.481716401 1526442512.587175764
[22514.863980] [Project1] 9793 1526442512.587813178 1526442513.641831251
[22515.928405] [Project1] 9794 1526442513.642461448 1526442514.706313522
[22516.981547] [Project1] 9795 1526442514.706969298 1526442515.759510209
```

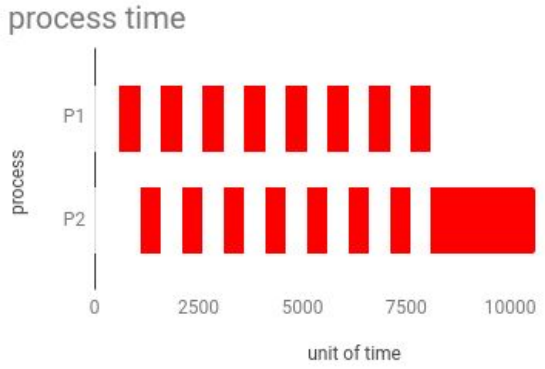
pid:9791 -1526442510.462989308 + 1526442511.480980999 = 1.017991691
pid:9792 -1526442511.481716401 + 1526442512.587175764 = 1.105459363
pid:9793 -1526442512.587813178 + 1526442513.641831251 = 1.054018073
pid:9794 -1526442513.642461448 + 1526442514.706313522 = 1.063852074
pid:9795 -1526442514.706969298 + 1526442515.759510209 = 1.052540911

實際總時間：1.017991691 + 1.105459363 +1.054018073 + 1.063852074 + 1.052540911 = 5.293862112s

Testcase2

```
RR
2
P1 600 4000
P2 800 5000
```

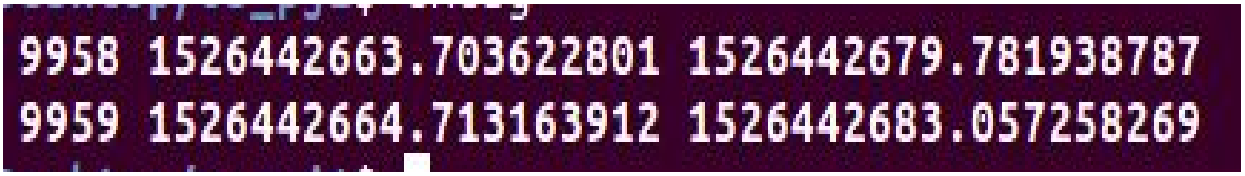
理論值：



理論總時間：9000/1000 * 1.896 = 17.064s

實際值：

```
$ sudo ./schedule < ./OS_PJ1_Test/RR_2.txt
$ dmesg
```

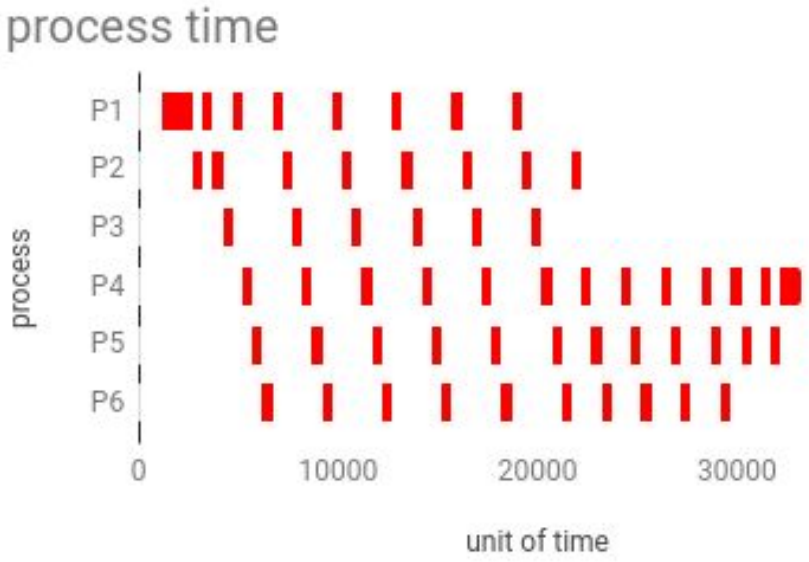


實際總時間：-1526442663.703622801 + 1526442683.057258269 = 19.353635468s

Testcase3

```
RR
6
P1 1200 5000
P2 2400 4000
P3 3600 3000
P4 4800 7000
P5 5200 6000
P6 5800 5000
```

理論值：



理論總時間：30000/1000 * 1.896 = 56.88s
實際值：

```
$ sudo ./schedule < ./OS_PJ1_Test/RR_2.txt
$ dmesg
```

```
10018 1526442807.512801599 1526442837.801354221
10016 1526442801.058431203 1526442839.873171341
10017 1526442804.272358866 1526442842.053116266
10021 1526442816.118244977 1526442859.356216800
10020 1526442813.998612234 1526442863.660314465
10019 1526442811.808930259 1526442865.764441917
```

實際總時間：-1526442801.058431203 + 1526442865.764441917 = 64.706010714s

3. 比較實際結果與理論結果，並解釋造成差異的原因

我們試著使用雙核以減少 parent process 對 children processes 造成的延遲，但虛擬機系統本身還是會佔用到兩顆核心的效能，這是各種 scheduling 都無法避免的延遲原因。

我們所有 scheduling 中，parent 都是以 busy waiting 來估計時間，這可能會出現些許延遲，導致出現了一段段空白時間。而且我們做的是 ”user space” 的 scheduler ，在 kernal 當中還是以 round robin 為排序方式，這也會對真實 processes 的運行造成影響。

4. 各組員貢獻

double_core.c: 張修瑞，張凱程
time.c、kernel file: 張修瑞，梁偉傑
report.pdf: 梁偉傑、洪敦敏