Operating System Project 1 Report

戴佑全
黃子賢
劉彥廷

April 25, 2016

1 FIFO

1.1 Design

資料結構 : 一開始打算以link list來當作ready queue，但後來想到有給process的數量再 加上並沒有用到link list可以在中間隨意插入的優點，所以改用array當作ready queue， 並有2個變數紀錄現在執行的child與下一個要被fork的child。

FIFO是一個很直觀的 scheduler，因爲只有一顆CPU，所以我的作法是parent將child要跑多少時間跟child說，之後將child的priority提到最高，而 parent 會重新拿到CPU有2種情況，

1. 因爲可能中途要fork其他的child，所以child需要在那時將CPU還給parent，這裡 使用的方式是將parent的priority設爲98，換child時將child提成99，當child要回給 parent時將自己設爲97，就能完成context switch了，而因爲此時child還沒執行完，只是 暫時換parent，所以在parent中現在執行的變數不變，所以fork完之後會接續做。

2. 當child執行完後，輪到parent時，將parent中現在執行的變數增加並wait child 以防有zombie。 所以parent的流程是，先檢查是否在idle(現在執行的child = 下一個要被 fork的child)，如果是則做for loop直到下一個child被出來，再來是檢查需不需要fork， 因爲並非每次輪到parent時都要fork，再來是將child需要跑的時間告訴child，時間爲 min(下一個要被fork的child的ready time, 現在執行的child的execution time)。

1.2 Result

1. `FIFO_1.txt`

```
P1 3507
P2 3508
P3 3509
```

```
P4 3510
P5 3511

[ 4145.108524] [Project1] 3507 1461500329.095585756 1461500330.144344924
[ 4145.108528] [Project1] 3508 1461500330.144345975 1461500331.188365292
[ 4145.108530] [Project1] 3509 1461500331.188366306 1461500332.216749967
[ 4145.108532] [Project1] 3510 1461500332.216751307 1461500333.245824809
[ 4145.108534] [Project1] 3511 1461500333.245826289 1461500334.286764881
```

2. FIFO_2.txt

```
P1 3515
P2 3516
P3 3517
P4 3518

[ 4517.340523] [Project1] 3515 1461500526.987120466 1461500691.939705154
[ 4517.340527] [Project1] 3516 1461500691.939706083 1461500702.258074897
[ 4517.340529] [Project1] 3517 1461500702.258075945 1461500704.295347989
[ 4517.340531] [Project1] 3518 1461500704.295349110 1461500706.332647694
```

3. FIFO_3.txt

```
P1 3523
P2 3524
P3 3525
P4 3526
P5 3527
P6 3528
P6 3529

[ 4736.784469] [Project1] 3523 1461500878.186589428 1461500894.640836343
[ 4736.784473] [Project1] 3524 1461500894.640837652 1461500904.988557021
[ 4736.784474] [Project1] 3525 1461500904.988558103 1461500911.196864232
[ 4736.784476] [Project1] 3526 1461500911.196865326 1461500913.265837919
[ 4736.784478] [Project1] 3527 1461500913.265839038 1461500915.317630553
[ 4736.784480] [Project1] 3528 1461500915.317631986 1461500917.408211062
[ 4736.784481] [Project1] 3529 1461500917.408212108 1461500925.666872035
```

1.3 Comparison

與理想中的差不多，但是在dmesg稍微有點誤差，原因大概是parent 與child之間context switch造成的，除了priority有保護順序外，因爲FIFO並不考慮preemptive，所以只有上一 個做完下一個才會做，再加上每次child死亡都會wait所以這也會保護到答案順序的正確。

2 SJF

2.1 Design

設計跟FIFO差別不大，唯一的差別是在fork完之後要做一次sort，有2種情況，

1. 現在有child正在執行中，那麼只sort除了這支child的其他已被fork出來的child。
2. 現在沒有child正在執行中，那麼sort所有已被fork出來的child。除此之外剩下皆與FIFO相同。

2.2 Result

1. SJF_1.txt

```
P2 2558
P3 2559
P4 2560
P1 2557

[ 1331.110283] [Project1] 2558 1461506228.616022414 1461506232.915866507
[ 1331.110286] [Project1] 2559 1461506232.915868797 1461506235.073549386
[ 1331.110288] [Project1] 2560 1461506235.073552214 1461506243.838296353
[ 1331.110290] [Project1] 2557 1461506243.838298302 1461506258.987460851
```

2. SJF_2.txt

```
P2 2710
P5 2714
P4 2711
P3 2713
P1 2712

[ 1520.327165] [Project1] 2710 1461506414.776505277 1461506414.976428066
[ 1520.327168] [Project1] 2714 1461506414.976541437 1461506415.378525063
[ 1520.327170] [Project1] 2711 1461506415.378527639 1461506424.004852075
[ 1520.327172] [Project1] 2713 1461506424.004854531 1461506432.734845293
[ 1520.327173] [Project1] 2712 1461506432.734847707 1461506448.109725066
```

3. SJF_3.txt

```
P1 2730
P4 2733
P5 2734
P6 2735
P7 2736
```

```
P2 2731
P3 2732
P8 2737

[ 1733.221024] [Project1] 2730 1461506590.620579045 1461506597.056205686
[ 1733.221027] [Project1] 2733 1461506597.056208329 1461506597.077219594
[ 1733.221029] [Project1] 2734 1461506597.077221749 1461506597.098146308
[ 1733.221030] [Project1] 2735 1461506597.098147463 1461506605.898578152
[ 1733.221032] [Project1] 2736 1461506605.898581016 1461506614.651940234
[ 1733.221033] [Project1] 2731 1461506614.651943364 1461506625.677029525
[ 1733.221035] [Project1] 2732 1461506625.677032127 1461506641.046276916
[ 1733.221036] [Project1] 2737 1461506641.046279188 1461506660.897146737
```

2.3 Comparison

SJF與FIFO很像，都是不preemptive，所以同FIFO，順序並不太會錯誤，在來就是因爲要頻繁的sort所以在時間誤差方面會比較大。

3 Contribution of Each Member