

# OS-Project1 Report

## 第32組

b05203047 徐衍新

b05902114 陳昕璘

b05902126 詹丰億

## How to compile and run program?

compile: make

run: sudo ./main < [input\_file]

## Implementation

分成三個.c檔，分別是main.c, process.c, scheduler.c, header有process.h跟scheduler.h。main.c僅僅讀入input，再把data交由schedule函式處理。scheduler.c中定義schedule的方式，包含主要的while迴圈，以及policy的判定。process.c處理fork, wait,還有根據policy來決定process處理的次序。在kernel\_files的目錄中，有兩個.c檔my\_gettime.c和my\_printk.c，分別是利用getnstimeofday和printk實作，將這兩份檔案編進kernel code，並且在syscall\_64.tbl和syscalls.h定義原型和增加entry（我們的function number是333跟334），child process就可以呼叫systemcall。

## 執行範例測資結果

```
h1066@h1066-VirtualBox:~/桌面/OS-project1-master$ sudo ./main < input
P1 6132
P2 6133
P3 6134
```

```
[ 9843.680449] [project1] 6132 1556641239.173766737 1556641239.201439831
[ 9843.713375] [project1] 6133 1556641239.201675217 1556641239.234373635
[ 9843.718783] [project1] 6134 1556641239.204998326 1556641239.239784212
```

## 比較實際結果與理論結果

根據提供的測資,平均的單位時間為0.002419534

實際結果比理論結果多出約18%,可能的原因為在程式中while迴圈裡的for loop執行時間未被考慮,致使執行時間較長,以及cpu執行速度並非定值。另外,在理想的計算中, context switch的時間為0,但實際執行context switch必然有耗費時間。

## Discussion

我們在編kernel的時候失敗過,無法順利reboot,後來發現是configuration出問題,就重新調整並編譯。透過VirtualBox,盡量控制在固定的環境,能夠有效提高debug效率。系統相關的程式debug相對困難,必須時常利用error flag跟stderr來除錯。

### 各組員的貢獻

b05203047:程式除錯, system call與編譯kernel, 執行與計算結果

b05902114:編譯kernel, 寫scheduler, 測試測資

b05902126:程式除錯, 修改.c檔裡的幾個function, 測試測資, 計算結果