

Project 3 part 2 (50%)

✧ Question :

某天，有位修習作業系統的學生，ccccc，來問助教 project2 scheduling 的問題，ccccc 根據 [morris](#) 的筆記在 Nachos 實作了 scheduler 並且寫了測試程式，測試資料如下：

圖一、Testing data

```
char *name[thread_num] = {"A", "B", "C", "D"};
int thread_burst[thread_num] = {3, 9, 7, 3};
```

在使用 Round-Robin scheduler (Nachos 預設的 scheduler) 的情況下，ccccc 預期的順序應該是類似 ABCD 這樣的順序，但是程式最終的執行順序卻是 ACBDB，結果如圖二所示：

圖二、Round-Robin Output from ccccc

```
ccccc@ccccc-VirtualBox:~/123/Nachos/nachos-4.0/code/threads$ ./nachos
A: remaining 2
A: remaining 1
A: remaining 0
C: remaining 6
C: remaining 5
C: remaining 4
C: remaining 3
C: remaining 2
C: remaining 1
C: remaining 0
D: remaining 2
B: remaining 8
B: remaining 7
B: remaining 6
B: remaining 5
B: remaining 4
B: remaining 3
B: remaining 2
B: remaining 1
B: remaining 0
D: remaining 1
D: remaining 0
No threads ready or runnable, and no pending interrupts.
Assuming the program completed.
Machine halting!

Ticks: total 400, idle 60, system 340, user 0
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
ccccc@ccccc-VirtualBox:~/123/Nachos/nachos-4.0/code/threads$
```

根據助教的思考以及研究後，認為認真上作業系統和寫 project2 的同學們有能力可以幫助 ccccc 解決這個問題，請同學們回答 ccccc 的問題，寫一篇完整的說明 report，幫助 ccccc 了解背後的原因吧！

對了，ccccc 是一位壞學生，從來沒有去聽過作業系統的課，未閱讀過相關課本，作業也習慣參考網路或學長姐的資源，且對參考資料的程式碼原理一知半解，所以 ccccc 幾乎跟沒修過課的學生一樣什麼都不懂。現在假定改作業的助教是 ccccc，請同學詳細解釋問題推理的過程，盡可能讓 ccccc 理解你的想法，讓 ccccc 能夠

在看完你的 report 後，能完全了解同學們的邏輯，以及理解問題出在哪裡。

✧ Report:

如果同學覺得圖二的 output 有問題，請同學在 report 中，寫下有邏輯且合理的解釋說服 ccccc：

1. 你覺得問題是什麼？
2. 問題的原因為何？
3. 找到這個問題的過程以及思考邏輯。
4. 要怎麼修改才能解決問題，請指出你修改的地方。
5. 為什麼這樣修改後能解決問題。
6. 更多解決思路。
7. 任何可以幫助 ccccc 的說明與想法。

如果同學覺得圖二的 output 沒有問題，請同學在 report 中，寫下有邏輯且合理的解釋說服 ccccc：

1. 為什麼這樣的 output 沒有問題？
2. 判斷的依據是什麼？
3. 更多確認正確性的方法。
4. 任何可以幫助 ccccc 的說明與想法。

- ✧ 上面的各點只是幫助同學們有個方向該怎麼寫 report，同學們不一定要按照上面的方式寫，上面的順序不一定是最好的寫法。
- ✧ 有時候用流程圖，或是其他圖表來輔助說明會讓助教更能夠理解同學們想要表達的意思。
- ✧ 同學們可以去看看恐龍書(課本)中，關於 round-robin 的片段，大概 3~4 頁，或許會給同學們一些想法，或許。
- ✧ **Nachos 內註解有說明不能更改的地方請不要更改！**測試程式與測試資料如果同學有**合理的理由**可做更改，不然請不要更動，有更改的同學也請在報告中說明你更改的理由。
- ✧ 如果同學們最後沒有找到合理的解釋/問題/解決辦法，也請將嘗試過的方法、想過的想法、到目前為止的成果以及思路，寫在 report 中，助教會依據同學們的推理過程給與分數。

針對解釋「不清楚」的舉例：

圖三、Example of unclear examination

```
if (status == IdleMode && !woken && _bedroom.IsEmpty()) { // is it time to quit?  
    if (!interrupt->AnyFutureInterrupts()) {
```

```
timer->Disable(); // turn off the timer  
}  
  
}
```

上面 code 片段是第二次作業中，一段判斷是否要關掉 timer 的邏輯，大部分的同學在第二次作業 report 中，沒有提到或是只用一兩句話帶過為什麼要加 ”!woken” 這個條件式詳細的原因或是理由，像是：

1. 沒人在睡覺
2. 沒有任務需要被喚醒
3. 沒有 Thread 被叫起床

這樣的解釋會讓愚笨的 ccccc 在看過同學們的 report 後，還是不了解加上 ”!woken” 的原因以及背後的意義是什麼，因此糾結在這點上而睡不著覺，cccccc 的心情會不好，給同學們的 report 分數就會降低，請同學們特別注意喔！

✧ File:

請同學們在 NTU COOL 上下載 [hw3_part2.tar.gz](#)，並在另外的資料夾下解壓縮，不要在同學原本的 nachos 資料夾下解壓縮，因為可能會覆蓋掉同學們原本的作業檔案。解壓縮後會看到整個 Nachos 的資料夾，請到 threads 的資料夾下執行 ./nachos RR，看看 output 的結果是不是和圖二的結果相同，然後重新 make 一次看看能不能 make 成功，如果有其他問題，請跟助教聯絡。

✧ 注意事項：

1. 這次作業的目的是想要了解同學們對於 nachos 以及作業系統知識的了解程度，以及在 review 別人的 code 後，是否有獨立思考並從中學習的能力，因此評分的標準會著重在解釋以及推理的過程上，包括詳細說明如何找出問題的過程、思考邏輯和使用的方法。
2. 請務必附上 reference，假如有與任何人討論，請附上他們的學號和姓名，不要說「我跟學長姐討論後的結果」這樣籠統的答案，否則助教會嚴厲制裁，送你這次作業鴨蛋！
3. 有問題可以來跟助教討論，但如果是 TA Hour 以外的時間，請寄信與助教約時間，也麻煩不要去搜尋助教的私人聯絡資訊（如：FB），否則助教會不予

理會。

4. 請各位同學盡早開始想想看本次作業的問題，助教覺得可能要花些時間想通，因為問問題的同学到現在還沒有想通！(因為他現在跟助教一樣要考期中考🙄)