

Operating Systems Programming Assignment 2

睡覺的助教

授課老師：廖峻鋒

TA：林俊安 廖宇凡

作業情境說明

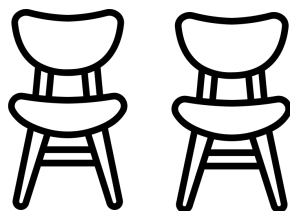
政大資科作業系統有一個助教(TA)小林，他在TA時間會在實驗室幫學生解決程式作業問題。實驗室很小，只放得下一張椅子和桌子。實驗室外走廊有 n 把椅子。當小林正在幫助學生時，其他學生可以坐在走廊等待。實驗室沒有學生需要幫忙時，小林會趴在桌上小睡。如果學生在TA時間抵達實驗室，又發現小林在睡覺，學生必須去叫醒小林。如果學生到達，發現小林正在幫助其他學生時，這個學生就會坐在走廊的一把椅子上等待。如果沒有椅子坐時，學生就會離開。

作業情境圖

Student



來問問題



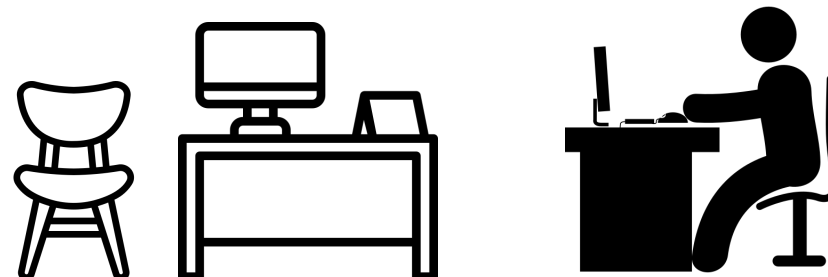
...
n把椅子



走廊

- 最多n個學生在走廊椅子上等待
- 所有椅子被佔據，則新來學生直接離開

TA



實驗室

- 最多一個學生
- 沒有學生來，TA睡覺

演算法

演算法為參考用，可以有不同的實作方法

TA

```
void *TA(void *temp)
{
    while (1)
    {
        1 sem_wait(&availableStudentsSem);    //等待是否有坐在實驗室外椅子上的學生
        sem_wait(&chairMutex); 5

        //...code
        //TA讓學生離開椅子

        8 sem_post(&TA_sem);    // TA available
        sem_post(&chairMutex); //此時已學生離開椅子，進實驗室 9

        //...code
        //開始幫忙學生。提示：調用usleep()。
    }
}
```

availableStudentsSem (初始值0，範圍0-n)

TA_sem (初始值0，範圍0-1)

chairMutex (初始值1，範圍0-1)

numOfOccupiedChairs 被佔據椅子的計數器
student

```
void *Student(void *temp)
{
    2 sem_wait(&chairMutex);
    3 if (numOfOccupiedChairs < MAX_CHAIRS_NUM) //檢查實驗室外是否還有多多的椅子可以坐
    {
        //...code
        //教室外有位子，學生佔據椅子

        4 sem_post(&availableStudentsSem);    // 等待學生加一
        sem_post(&chairMutex);
        7 sem_wait(&TA_sem);    // 看TA是否有再忙 6

        //...code
        //開始接受助教幫忙(更動時間)。
        //提示：可調用waitingTime，timeMutex，usleep()與numOfOccupiedChairs等等。
    }
    else
    {
        4 sem_post(&chairMutex); // 教室外沒有位子，學生離開
    }
}
```

POSIX

- **可移植作業系統介面**（英語：Portable Operating System Interface，縮寫為**POSIX**）是IEEE為要在各種UNIX作業系統上執行軟體，而定義API的一系列互相關聯的標準的總稱，其正式稱呼為IEEE Std 1003，而國際標準名稱為ISO/IEC 9945
- **POSIX**（英語：POSIX Threads，常被縮寫為Pthreads）是POSIX的執行緒標準，定義了創建和操縱執行緒的一套API。
- 實現POSIX thread標準函示庫常被稱作**Pthreads**，一般用於Unix-like POSIX 系統，如Linux、Solaris。但是Microsoft Windows上的實現也存在，例如直接使用Windows API實現的第三方庫pthreads-w32；而利用Windows的SFU/SUA子系統，則可以使用微軟提供的一部分原生POSIX API。

資料來源：維基百科

可參考教學

- Pointer與Memory address : <https://openhome.cc/Gossip/CGossip/Pointer.html>
- C 語言 Pthread平行化程式設計入門教學與範例
<https://blog.gtwang.org/programming/pthread-multithreading-programming-in-c-tutorial/>
 - 需要理解的實作部分
 - Pthread
 - Mutex
 - Semaphore
- 相關演算法教學影片 (sleeping-barber problem)
 - <https://www.youtube.com/watch?v=OvJFpsN5czg>

參數設置與作業要求

➤ 參數設置

1. $n = 10$ (走廊上椅子數)
2. 小林每幫助一個學生需要花2單位時間
3. TA時間為100單位時間以內，每個單位時間是否有學生來請參考作業資料中(input.txt)，1表示有學生來，0表示沒有學生來。程式需直接讀取input.txt檔。

EX：

1
1
0
0

表示只有第一個與第二個單位時間有學生來

4. 最後小林需要幫助完在走廊上等待的學生，程式才會結束

➤ 輸出參數要求

同學的程式最後需要在終端機print出有進實驗室的每位學生（沒等待的不算）平均待在實驗室內外的時間長度（也就是從佔據走廊上椅子到從實驗室離開的時間，資料型別為double），時間單位為單位時間(下一張投影片中waitingTime所定義的單位時間。)

Sample code 中變數定義與hint

➤ Sample code 中主要之參數定義(同學可以另外自訂參數，以下不一定要全部用到)

- chairMutex, timeMutex：mutex(更改實驗室外的椅子與時間值時需要用到)，初始值1
- availableStudentsSem：表示學生之semaphore，初始值0
- TA_sem：表示助教之semaphore，初始值0
- allStudents：表示100個單位時間內，來問問題學生總數
- waitingStudents:表示100個單位時間內，有進實驗室的學生總數
- waitingTime：所有學生總等待時間計數器(waitTime = 1表示一個單位時間)
- numOfOccupiedChairs：表示目前走廊上被佔據椅子的數量(初始值為0)。要避免numOfOccupiedChairs同時被兩個學生使用(可以使用chairMutex)

➤ 時間部分說明

- 在sample code中，每個單位時間設置為1毫秒，亦即每1毫秒都可能會有新的學生來。
- 輸出結果之算法「平均單位時間」：所有學生總等待時間/進實驗室的學生總數
- 記得要確保所有thread執行完之後再印

➤ usleep(1) 中參數單位為1微秒(10^{-6} s)

- 用意：使該process/thread在參數內之時間暫停執行
- 舉例：usleep(1000000)代表1秒

➤ 題目期待同學參考作業說明提供的演算法。在sample code中加入並調整演算法，適當的加入usleep()關鍵字與維護waitingTime、waitingStudents、numOfOccpiedChairs變數。

系統安裝與執行結果

➤ Windows系統：需要安裝virtual box(使用linux環境執行)

- 請參考「moodle上的VM 安裝教學」

➤ Linux/MacOS系統：請直接使用gcc進行編譯

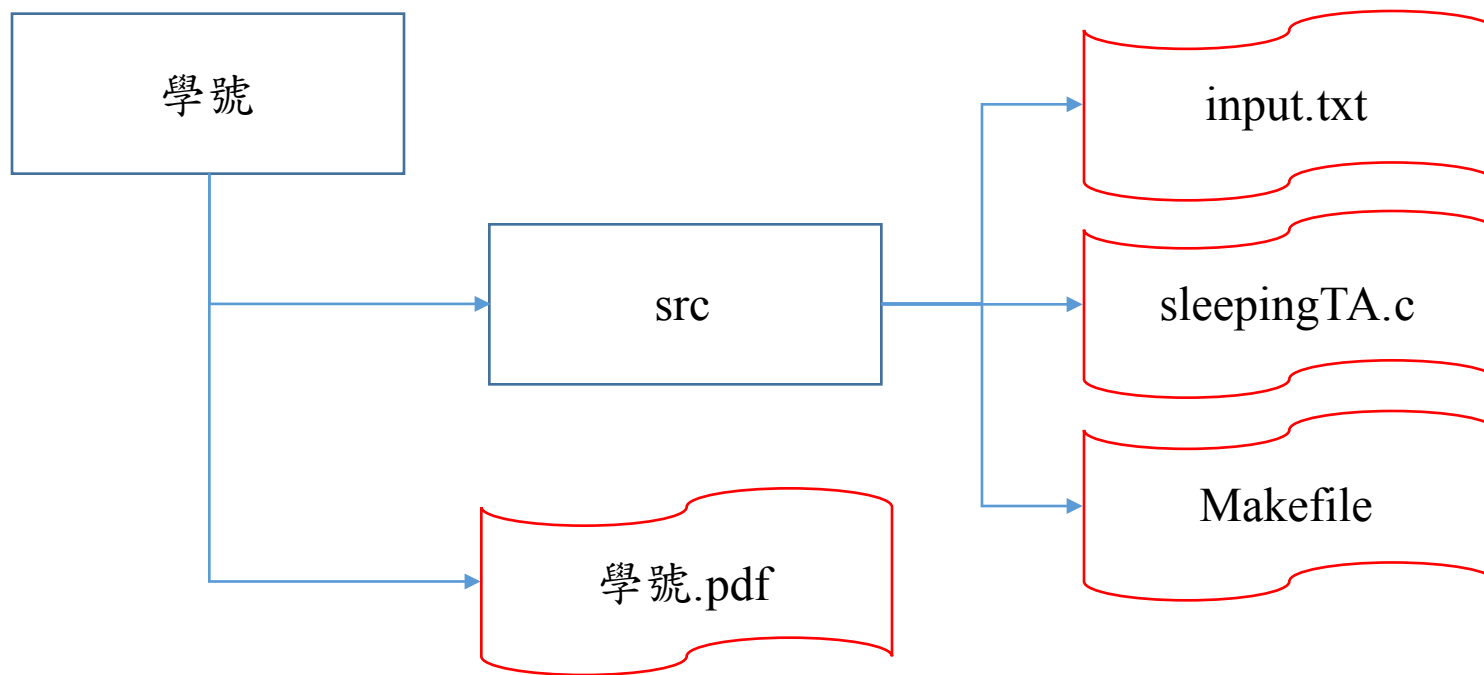
- Makefile已經寫好，請直接在終端機下“make”指令。如下圖：

```
nccude-MacBook-Pro:src nccu$ make  
rm -f sleepingTA  
gcc sleepingTA.c -o sleepingTA -lm -pthread
```

➤ 執行結果範例(由於每台電腦的thread跑的時間可能不同，所以非絕對答案)

```
Avg waiting Time 14.190476
```

作業繳交格式



請將檔案壓縮成.zip檔繳交

About Programming Assignment

1. 作業評分規則

- 有依照作業要求繳交: 40分
- 程式碼compile無誤: 60分
- 未按規定繳交作業: 一項扣10分
- 遲交0分

2. 程式作業請獨立完成，請勿抄襲同學之程式碼

3. 程式碼不一定需要按照助教提供的sample code完成，只要確保在終端機中輸入make指令後能夠輸出作業所要求的結果(作業統一以c撰寫)

About Programming Assignment

3. 繳交作業內容：

- 實驗結果圖
- 程式運作流程
- 心得