

系統程式

作業四

● 作業內容

- 設計一個程式，允許多名助教可同時輸入成績(可以同時操作，但程式寫入的動作可以錯開)
- 多個助教，無限多個同學，每個同學 5 個成績
- addGrade 為執行檔，執行之後會等待使用者輸入資料，第一個參數是學號，後面五個是成績
- 所有的成績都刊登在 grade 檔案內

● 作業要求

- code 需排版，限用 c 語言（不含 c++）
- 不需要登入助教帳號，直接輸入資料

```
alex@alex-B85M-D3H:~/文件/test$ ./a.out
101 1 2 3 4 5
```

- 可同時以多個 terminal 執行 addGrade。
- 本作業重點為對檔案操作、上鎖解鎖、練習 critical section 的概念，登入登出、排序、資料格式只要輸出成績結果正確即可，實作方法細節並不納入計分。
- 限用 open()、close()、write()等上課所用到的 function，不可使用其他相同效果的 function，ex: 用 fprintf() 取代 write()。
- 輸入成績無限 loop，每一次都是一個同學的成績，直到 ctrl + c 跳出
- 已有的成績不必刪除、覆蓋
- 所有成績從資料末端接續，若每次 addGrade 都從第 1 行覆蓋或建立新檔案，檔案操作 -15
- 模擬 critical section，設計一個機制假裝檔案使用中，期間其他助教無法使用檔案，並能以人眼觀察到。
ex: 倒數 5 秒才離開 critical section、按任意鍵離開 critical section
- 有 A,B 兩個助教，輸入的順序是 A1->A2->B1->A3->B2->A4->B3
那麼最後 grade 裡面的順序也要是 A1->A2->B1->A3->B2->A4->B3
然而去年半數的人，grade 裡的順序是 A1->A2->A3->A4->B1->B2->B3，也就是 A 關閉終端機 B 才能寫入成績，A 從頭到尾都鎖著 grade 不放，這樣並沒有達成「AB 可交錯輸入」的目的，因此去年半數的人沒完成「critical section」80 分，只有 10 個 100 分。大概是去年 SP 中最低分的作業。因為曾經很多人錯在這裡（或直接放生），所以重複一遍，

要有可觀測的 critical section 且 grade 裡的成績也要是交錯的

code (90%) = 其他基礎	(30%)
+ 檔案操作	(30%)
+ 可觀測的上鎖解鎖 critical section 且成績交錯輸入	(20%)
+ 可閱讀的成績	(10%)

makefile (10%)

包含所有 code、makefile，助教執行 make 後必須產生執行檔，執行檔名稱「addGrade」。

將 code、makefile 加入壓縮檔，**格式為 zip**

請上傳之後再下載一次檢查有無損毀，若評分時有損毀視同未交。

● 繳交期限

2018/4/17