

# CS305 作業系統概論 Prog. #4 Thread Coordination

2018.05.23

## 一、 作業目的

在本作業中，利用pthread的方式來練習process coordination的觀念。

## 二、 作業內容

在C市裡，有一家 StartJokes 咖啡專賣店，因應畢業季節的來臨，限量推出一種特製的咖啡，每天只賣出100杯。咖啡有下面四個配料：特調咖啡豆 (B)、牛奶(M)、特製焦糖 (S) 和肉桂 (C)。由於太過暢銷，因此在店裡有三位前台店員 (FC) 負責做好咖啡，及一位店長 (CM) 負責將咖啡的配料送到 FC 的手中。為了加快做好咖啡的速度，三個 FC 各別會先擁有四種配料中的一種配料 100份，而且每位 FC 的配料都沒有重複，當天也不會再有變動。所以每個 FC 都不需再去拿自己擁有的配料，只需要等 CM 在後面的料理台上準備配料，然後動態的去拿取這些配料。在料理台上，CM每次隨機供應一種配料，每種配料桌上一次只有擺放一份。因此如果料理台上某種配料已經有了一份，CM 就不會供應該種配料，而會隨機補其他的配料。

當CM 準備配料時，FC 拿取配料的方式如下：每位 FC 拿的時候，必須一次拿走所缺的三種配料，不能只拿部份的配料。如果可以取得缺的配料，這時便可以加上本身擁有的配料，做成一杯特製咖啡。如果料理台上的配料還不能讓自己完成一杯咖啡，就不拿取，隔一段時間（5ms）再去檢查。當配料被拿走後，CM會一直隨機地補充配料到料理台上但符合之前的規定。這個流程會不斷循環，一直到100 杯限量特製咖啡做完為止。

請設計一個程式，模擬這些店員同步處理的動作，在程式中做出適當的輸出，顯示咖啡店製作特製咖啡的流程，FC 完成限量100杯中的第幾杯咖啡（**請注意，這個流程就是助教要評分的项目，因此務必清楚顯示出製作咖啡的流程**）。因為StartJokes咖啡店限量 100 杯特製咖啡，請在賣出第 100杯特製咖啡時，結束程式，並印出每個 FC 總共做出多少杯特製咖啡，以及他們還剩多少配料。

程式執行時，需要在命令列輸入功能選項（基本功能為 0），並指定三位店員的配料為何。

以下是一個可能的情況：

> prog4 0 B M S

CM: B

CM: M

CM: C

FC(S): 第 1 杯特製咖啡

...

FC(B): 第 10 杯特製咖啡

...

FC(M): 第 29 杯特製咖啡

...

FC(B): 做出 30杯，剩 70 份咖啡豆

FC(M): 做出 40杯，剩 60 份牛奶

FC(S): 做出 30杯，剩 70 份蜜糖...

## 三、 作業要點

1. 請注意，本作業使用的程式語言是C/C++，測試平台的作業系統： Ubuntu 17.10 LTS 64-bit。使用的編譯程式為gcc/g++ 編譯器：7.2。其他平台或程式語言不在本次作業考慮範圍之內。如在測試平台上無法編譯與執行，都不予給分。

# CS305 作業系統概論 Prog. #4 Thread Coordination

1051446 游采蓉

- 編譯

```
parallels@parallels-vm:~/Desktop$ g++ -o out main.cpp -lpthread
parallels@parallels-vm:~/Desktop$ ./out
```

輸入指令： g++ -o out main.cpp -lpthread

- 執行結果：

```
0 B M C
CM: C
CM: S
CM: B
FC(M): 第 1 杯特製咖啡
CM: C
CM: S
CM: M
FC(B): 第 2 杯特製咖啡
CM: C
CM: S
CM: M
FC(B): 第 3 杯特製咖啡
CM: B
CM: S
CM: M
FC(C): 第 4 杯特製咖啡
CM: B
CM: M
CM: C
CM: S
FC(M): 第 5 杯特製咖啡
CM: S
CM: C
FC(B): 第 6 杯特製咖啡
.
.
CM: B
CM: M
CM: S
FC(M): 第 99 杯特製咖啡
CM: C
CM: B
CM: S
FC(M): 第 100 杯特製咖啡
FC(B): 做出 32杯，剩 68 份咖啡豆
FC(M): 做出 37杯，剩 63 份牛奶
FC(C): 做出 31杯，剩 69 份肉桂CM
```

基本功能：

在main function 中用pthread\_create產生共4個thread，並各自去執行它們的function。CM\_Threading裡去做CM隨機的配料順序的功能，CM 擺放四種配料的順序，以亂數函式srand(0)設定亂數序列，如果遇到台上三種原料，FC還不能讓自己完成一杯咖啡，就不拿取，sleep(5)再讓大家去搶。使用pthread API中的 mutex 機制成critical section，來設計FC之間拿取配料的控制，以及CM與FC之間同步的控制。

另外有多做選項(2)的功能

輸入範例：

2 B M C S

多了第4位FC，每位FC所預備的配料都不相同，然後一起完成 100 杯限量特製咖啡。