

CS305 作業系統概論 Prog. #3 Task Coordination

2021.11.22

一、作業目的

在本作業中，利用pthread的方式來熟悉 synchronization 的觀念。

二、作業內容

使用 pthread 來模擬哲學家問題。從命令列讀入兩個整數，第一個1~3的整數 m 表示模擬的演算法，第二個數字是一個 3~11的整數 n ，程式就會產生 n 個執行緒來代表有 n 個哲學家。程式的一些變數定義如下：

1. 模擬的演算法如下：

- a. 7.1.3 的基本演算法 ($m=1$) : One simple solution is to represent each chopstick with a semaphore. A philosopher tries to grab a chopstick by executing a wait() operation on that semaphore. She releases her chopsticks by executing the signal() operation on the appropriate semaphores.
- b. 7.1.3 的Asymmetric 演算法 ($m=2$) : An odd-numbered philosopher picks up first her left chopstick and then her right chopstick, whereas an even-numbered philosopher picks up her right chopstick and then her left chopstick.
- c. 7.1.3 的同時拿筷子的演算法 ($m=3$) : Allow a philosopher to pick up her chopsticks only if both chopsticks are available (to do this, she must pick them up in a critical section).

2. 每位哲學家第一次要開始吃飯時，先隨機等待 1~5 秒。

3. 每位哲學家每次吃飯所花的時間長度是一個 1~5 秒的隨機時間。

4. 每位哲學家每次思考所花的時間長度是一個 5~10 秒的隨機時間。

在程式進行中，請將哲學家的狀態詳細輸出，至少需要印出下面六種情況，以便助教檢查。

- d. 拿到一隻筷子
- e. 拿到另一隻筷子
- f. 用餐
- g. 放下一隻筷子
- h. 放下另一隻筷子
- i. 思考

哲學家狀態輸出時，同時要印出當下電腦的Wall time 時間，以秒為單位。當所有哲學家都吃過10次餐點以後，程式就停止。

例如，下面是一個可能的輸出：

```
> prog3 1 5
```

```
15:03:01-Phi 1-taking right chopstick
```

```
15:03:01-Phi 1-taking left chopstick
```

```
15:03:01-Phi 1-eating
```

```
15:03:02-Phi 2-taking right chopstick
```

```
15:03:02-Phi 3-taking right chopstick
```

```
15:03:02-Phi 4-taking right chopstick
```

```
15:03:03-Phi 1-putting right chopstick
```

```
15:03:03-Phi 2-taking left chopstick
```

```
15:03:03-Phi 1-putting left chopstick
```

15:03:03-Phi 2-eating

15:03:03-Phi 5-taking right chopstick

15:03:03-Phi 1-thinking

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 21.04 64-bit。使用的編譯程式為gcc/g++ 編譯器：10.3。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。
2. 請注意，本作業必須要用pthread API中的 mutex機制來進行，例如pthread_mutex_init()。不能使用semaphore機制，例如sem_init()。**任何不用pthread API mutex機制的作業，本作業只能最多得到[基礎]部份的30分。**
3. 本作業的評分方式如下：
 - a. [基礎] 正確產生模擬所有哲學家的 threads者，最多可得20分。
 - b. [基礎] 亂數的產生必須在主程式一開始就用 srand(0) 的方式設定亂數種子，每次執行都會產生相同的亂數序列。（助教會修改 seed 值，進行測試），最多可得10分。
 - c. 正確使用pthread API 中的 mutex 機制形成critical section來設計，最多可得10分。不使用此機制者，此部分 0分，以下各演算法的實作也不計分。
 - d. 各個演算法正確執行，可得以下分數：
 - i. 基本演算法：完成本演算法者，最多可得20分。
 - ii. Asymmetric 演算法：完成本演算法者，最多可得20分。
 - iii. 同時拿筷子的演算法：完成本演算法者，本演算法最多可得20分。
 - e. 進階功能
 - i. 死結偵測：如果所有哲學家都停止動作維持在某一個固定狀態，在等待20秒鐘以後，程式判定為死結發生，此時印出死結訊息，同時程式也結束。**如果有實作死結偵測，必須要再說明文件檔中註記說明，不然助教不會進行判讀。**本項死結偵測實作的部份，最多可得20分。
4. 本作業需繳交檔案：
 - a. 說明報告：檔案為docx或pdf格式。
 - i. 報告中必須說明程式的設計理念、程式如何編譯，以及如何操作。
 - ii. 報告中同時必須詳細說明你完成哪些部份。如有用到特殊程式庫，請務必說明。
 - iii. 請務必讓助教明白如何編譯及測試你的程式。助教如果無法編譯或測試，會寄信（**最多兩次**）通知你來說明，但每說明一次，助教會少給你10分。
 - b. 完整原始程式碼檔案（.c 或 .cpp）。不可含執行檔。助教會重新編譯你們的程式。請注意：不可用 .txt檔或是 .docx檔等非正常方式繳交程式碼，如有類似情形，助教也會扣10分。
 - c. **不可以含有病毒，如果含有病毒等惡意程式，本作業0分。**
5. 所有相關檔案，例如報告檔、程式檔、參考資料等，請壓縮成一個壓縮檔（不可超過2MB）後上傳至portal。請注意，不可抄襲。助教不會區分何者為原始版本，被判定抄襲者，一律0分。

四、繳交方式：

1. 最終繳交時間：
 - a. 電子檔在 2021.12.22 以前，上傳至個人portal。如有多個檔案，將所有檔案壓縮成zip（rar 亦可）格式，然後上傳。

b.上傳檔名格式：「學號_作業號碼.doc」或「學號_作業號碼.rar」。例如：912233_01.doc 或 912233_01.rar。

2. 如有違規事項者，依照課程規定處理。

3. 如需請假，請上portal請假，並持相關證明文件，在請假結束後的第一次上課時完成請假手續，並在一週內完成補交。補交作業將以8折計算。

4. 老師不接受「門縫」方式繳交，助教也不接受任何作業。

五、 如有未盡事宜，將在個人portal板面公告通知。

六、 If you need **an English version** of this assignment or **any assistance in English**, please contact Prof. Yang.

七、 參考資料

1. 課本第6-7章

2. “How to Use C Mutex Lock Examples for Linux Thread Synchronization”,
<http://www.thegeekstuff.com/2012/05/c-mutex-examples/?refcom>

3. “pthread：mutex”，<http://angelonotes.blogspot.tw/2012/02/pthread-mutex.html>