CS305 作業系統概論

Prog. #3 Task Coordination說明報告

汪文豪(學號：1071710)

* **如何編譯與測試操作程式：**

1. 編譯：g++ 檔名.cpp –o 檔名 –lrt -pthread

2. 執行：./檔名 m n, 欲讀入資料檔案之檔名 以及 m:演算法的模式1-3 n: 人數 3-11個(threads)，倘若超出範圍，會報錯並且請您重新

執行檔案

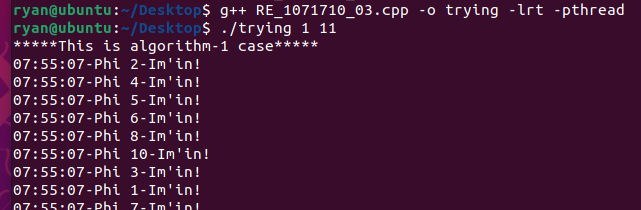
3. 輸出結果

範例(選擇演算法模式為1、人數為11人)：

一. g++ RE\_1071710\_03.cpp –o trying –lrt -pthread

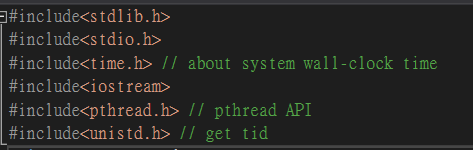
二. ./trying 1 11

三. 輸出結果



* **設計理念：**

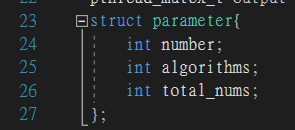
**本次使用到的特殊函式庫**

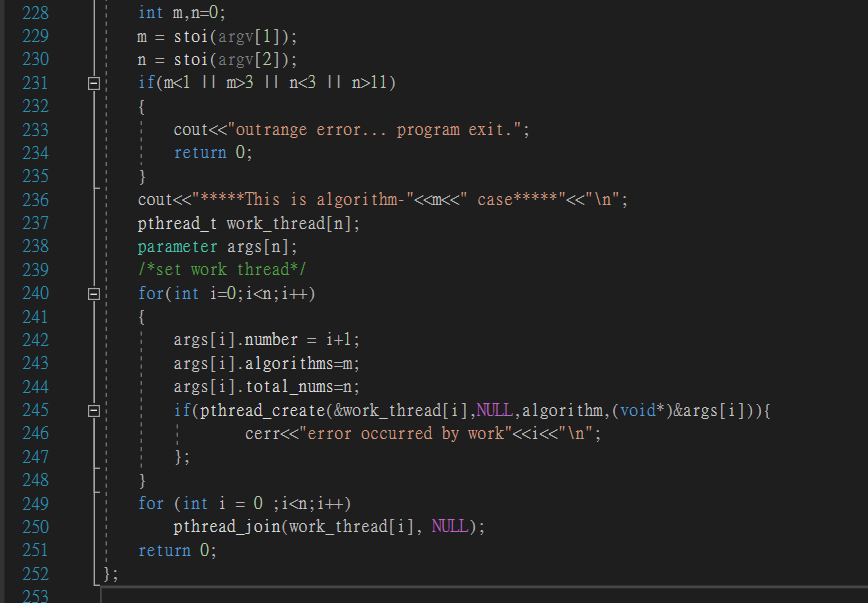


**本次程式說明：**

* 完成部分：

1. **[基礎] 正確產生模擬所有哲學家的 threads者，最多可得20分。**

****

****

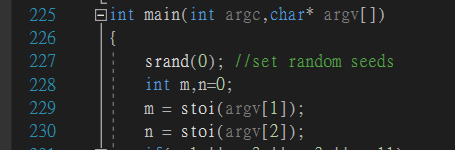
Parameter struct儲存著

* + - number => 該哲學家的編號
    - algorithms => 1 – 3哪一種演算法
    - total\_nums => 3-11人的總人數，用來計算筷子用

**為求程式方便進行，因此多開一個空間，讓0的地方空著，以1開始起數。**

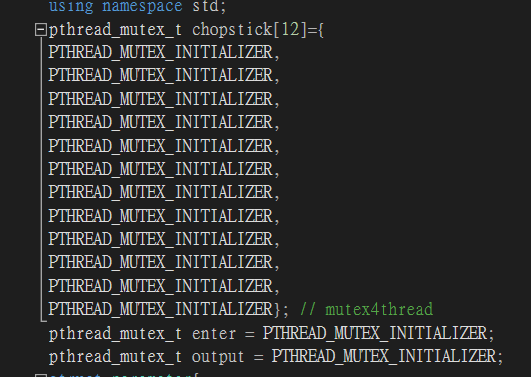
可依照需求正確產生threads的數量

1. **[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。**



227行設置random seeds.

1. **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。**

****

使用pthread作為筷子的設計。

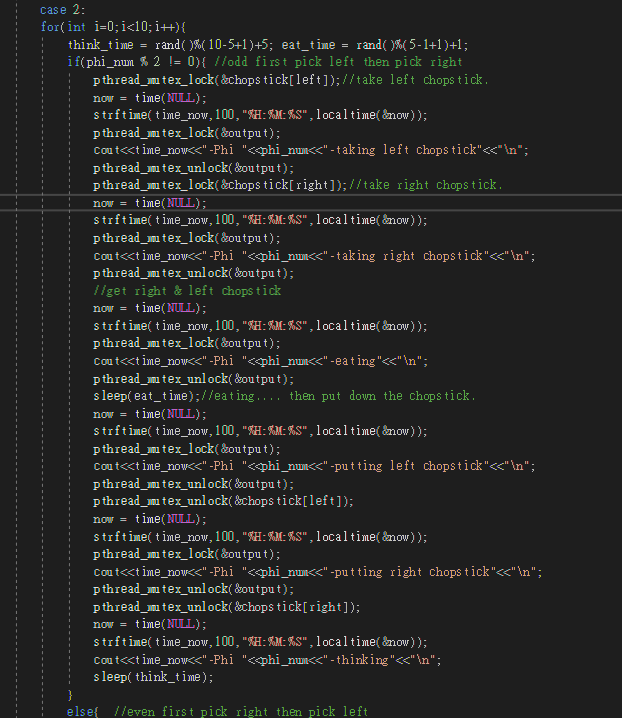
1. **各個演算法正確執行，可得以下分數：**
   1. **基本演算法：完成本演算法者，最多可得20分。**

****

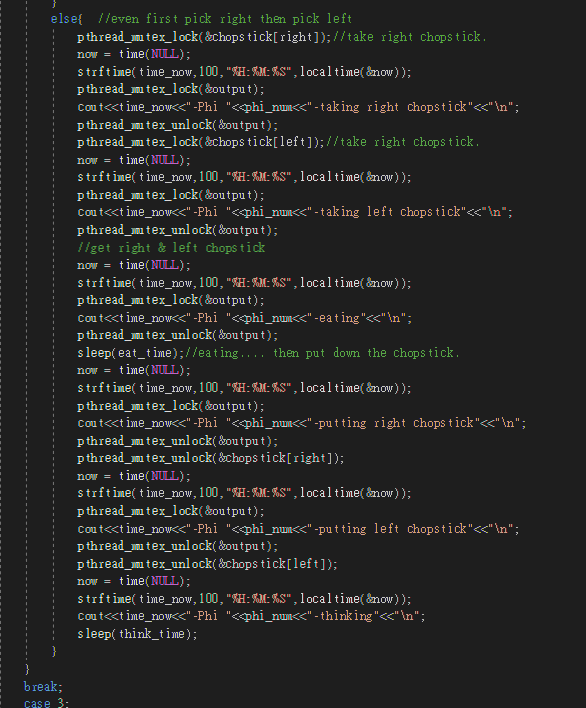
基本演算法，將哲學家的左右手邊的筷子的鎖拿到手後即可開始吃飯，吃1-5秒，之後放下筷子進行思考5-10秒，其中print出時用output上鎖，以免印出畫面時thread彼此衝突。 For迴圈設定為10次，吃完10次並思考後結束並印出thread leave的時間

* 1. **Asymmetric 演算法：完成本演算法者，最多可得20分。**

大致上與上一個差不多，但多加一個限制，若是奇數哲學家，先拿左手邊的筷子，再拿右手邊的筷子

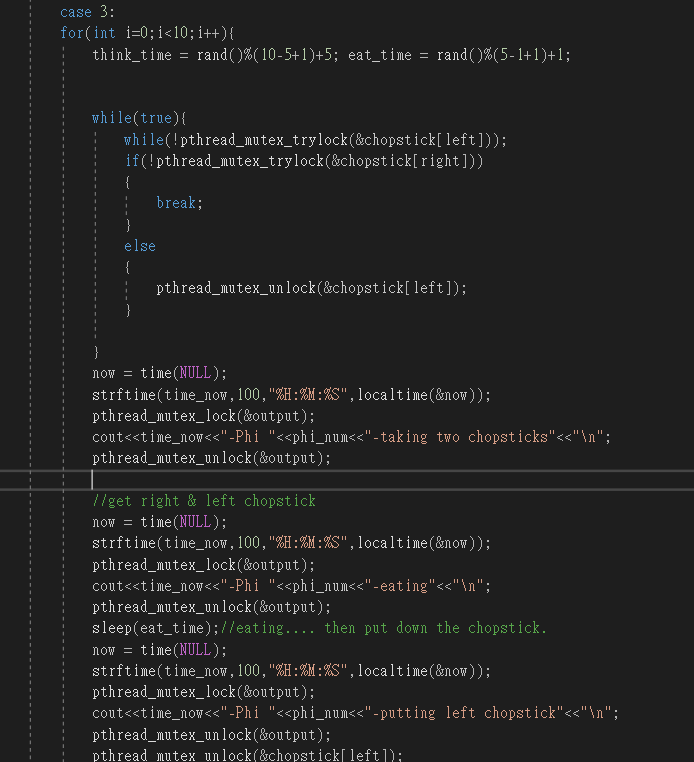
****

而偶數則相反，先拿右手邊的筷子再拿左手

****

* 1. **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。**

第三個演算法則是先拿左手的筷子，若右手的筷子無法拿，則將左手的筷子放下，若可以拿則印出拿到兩支筷子的message並且進行吃飯

****