OSHw3 Fish cutter!

A homework of making a parallel production line

Due date: 6/16 (FRI.) 23:59 (好孩子請勿遲交、抄襲，期末專題不接受補交)

*目的:

在這個作業中，我們將從上一個處理程序(Process)間的訊息溝通的問題，變成一個多執行緒(Thread)資料交換與同步問題。同時，我們將進一步擴展為多步驟的 Consumer-Producer 產線問題，老師預期你會需要第 6-8 章行程同步/死結等概念來實做整個作業。
在這個作業中你會學會以下幾件事：

1. PThread 的基本使用、執行緒間資料共享
2. Mutex 以及(Counting) Semaphore 的使用與設計
3. 如何管理各種佇列、排隊的問題 (甚至避免死結)
4. 思考如何好好運用多執行緒來模擬真實系統

***開始撰寫時請注意，每個人產出的結果不會相同是正常的，過程中我們會檢查你每一隻魚的生命週期跟每個工廠的行為是否正確，藉此判斷你的程式是否正確。**

故事從一個魚料加工廠開始，我們將請你撰寫一個魚料加工廠模擬器，透過平行的方式將魚料經過切割(Cutting)和製作罐頭(Canning)兩個步驟來完成。每條魚會交替經過排隊跟加工的過程。過程中平均每 5~10ms 會送上一條從魚貨船拿出來的鮮魚到切割工廠，為確保魚料新鮮，切割工廠備有N個備料格（slot, 預設為 5 個），擺放在備料格上的魚料將以先到先處理(FCFS)的排隊方式送進切割工廠，切割過程中魚需要在工廠中待上 10~30ms 才能切割完畢。並在完成之後送進罐頭工廠進行加工。

切割完成後，魚料會先放回備料台等待進入工廠（會標示為已切割），排隊等待工廠有空再送入，這部分依然是先切完的會先送罐頭工廠(FCFS)排隊。進入罐頭工廠後將會進行 50~100ms 的加工。魚料經過兩階段的工作就會完成加工並且離開工廠。

這中間如果遇到都沒有魚料需要處理的狀況，切割工廠和罐頭工廠就會一起進入檢討模式(under reviewing)，若是個別沒有工作的情況下則會進入維護模式(under maintenance)，這部分建議超過 10ms 以上再檢查，以免因為頻繁進入檢討模式而產出大量訊息。

此外，若是 N 個備料格都滿了，而且切割工廠也正在運作（第 N+1 個魚料），那麼新來的魚料就必須放進冷凍庫(Freezer)花費 30~50ms 排隊，若是時間結束仍沒有空位，那麼就必須再回到冷凍庫等候。每次魚料被處理完之後就會放回備料格，萬一 N+1 個魚料都處理完畢，那麼最後一個完成的魚料直接放在切割場中等待製作罐頭（此時切割場會進入維護模式）。系統預設輸入 M 條魚，程式執行時將允許使用者輸入 M 與 N。M 條魚都處理完則程式結束。

本作業的要求就是請你寫出一個魚料工廠模擬器，模擬時，要輸出魚料以及各個工廠的狀態（於狀態改變同時輸出時間、所要執行的工作以及工作時間長度），例如有新的魚料進來以及變成罐頭離開、工廠處理魚料、工廠進入維護模式、工廠們一起進行檢討等等。整題時間是由 0ms 開始，平均每 5~10ms 會送上一條鮮魚，起始時工廠們預設在維護模式。在維護模式或檢討模式中，有任何魚料進來則會盡快動工。

***以上的 ms 只是一個相對的時間單位，你可以自己使用一個 clock thread 來負責計時**

***該如何設計有哪些 Thread 去負責那些任務，是這個系統設計的關鍵工作。請務必在報告中詳細陳述你的設計以及為何可以達到工作。**

程式執行參數範例:

./hw3.out 15 5

以上表示有 15 條魚，切割工廠共有 5 個備料台

(每個切割工廠還可以多存放一個正在切割的)

*注意事項：

1. 本作業僅限 C/C++，並使用 Pthread API。測試平台則跟前面的作業都相同。（無法編譯執行均不給分）實作必須以 pthread API 來建構 multithreaded 環境，並使用其中的 mutex 或 semaphore 機制形成 critical section 來設計同步與資源分享。沒有考慮同步跟共享機制的，作業 0 分。Thread 的設計基本上是完全自由的，你可以是每個狀態(工廠/佇列)都有一個 Thread 控管，來抓取魚來做事。或是每個工廠都有佇列(共享資源)，魚本身作為 thread 來去每一個站報到處理。。。這部分的設計有許多可能，請務必在報告中說明清楚。
2. 請務必仔細閱讀我的輸出，並且跟著印出對應項目（像是工廠正在處理哪一條魚、處理多少時間等務必印出，否則無法批改）**每個人產出的結果不會相同是正常的，過程中我們會檢查你每一隻魚的生命週期跟每個工廠的行為是否正確，藉此判斷你的程式是否正確。**

3. 在程式中請使用亂數來產生每一個工作所需的時間，請在程式開始時運用亂數函式 srand(0) 來設定亂數序列起始點，並用 rand() 來取得亂數。範例：如果是取 10~20 的亂數，請用 **(rand()%11)+10** 取得。
4. 在程式中，ms(milliseconds)為時間控制基礎單位，可以考慮使用 usleep()函式來完成。不過本作業中 ms 只是一個相對的時間單位，你可以自己使用一個 clock thread 來負責計時。只要程式可以用正確的順序跑完所有工作即可。
5. 由於使用 PThread，本作業將會需要在編譯時連結 pthread 函式庫，參考範例如下：
   g++ XXX.cpp -lpthread -o XXX.out
   (沒意外的話，你原裝的 ubuntu 應該已經內含 pthread，有需要使用 c++11 的請自行加上參數並在文件註明)

*BONUS：

1. 可以設置一個以上的切割場以及罐頭製作工廠 (+10pts)
   （以上可以透過 cout/cin 互動的方式提示使用者輸入數值，請提示使用者該如何輸入。比方說兩個工廠的數量就可由使用者一次輸入，預設為 1 1 ，兩種工廠各一個）
2. 在冷凍庫的總時間長度可以設置上限，超過上限則自動離開冷凍庫，並在備料台旁邊排隊。也就是說，在備料台有空位後可以優先使用。同時，需設置在超過一定時間後必須拋棄離開，請新增一個魚料壞掉事件。(+10pts)
3. 說明在你的設計中死結發生的可能性，若有機會請採用對應的策略避免，如設計 Monitor (+10pts)

*作業繳交:

1. （90%）程式碼（單一檔案）：請繳交你可編譯的程式碼，不需要附上任何編譯好的檔案。程式碼請用以下形式命名：s1234567_OShw3.cpp (結尾可是.c 或.cpp)
1. （10%）報告：s1234567_OShw3.pdf，報告需要說明如何編譯你的程式、你完成的功能（列表），Thread 的設計，整個程式是如何運作（設計技巧以及如何運用 Pthread/ mutex/ semaphore 機制）以及如果有 BONUS 功能請詳細說明使用方法跟提供一些測試範例（過程與截圖），推薦以投影片或是報告方式呈現，有錄音更好。

*參考資料

[1] POSIX Threads Programming | LLNL HPC Tutorials
[2] Linux Tutorial: POSIX Threads (yolinux.com) （多個範例程式）
你也可以在系統中直接查詢完整文件，比方說鍵入：man pthread_mutex_lock
中文的參考資料，也算是相當詳盡，不過資訊比較舊一點
POSIX 線程(pthread)入門文章分享 @ 真實旅程 :: 痞客邦 :: (pixnet.net)

測試的輸出範例在最後！請務必閱讀

***請注意，每個人產出的結果不會相同是正常的，過程中我們會檢查你每一隻魚的生命週期跟每個工廠的行為是否正確，藉此判斷你的程式是否正確。以下僅是模擬內容，在不同次執行結果不同。**

Output 範例 (./hw3.out 10 5)

00ms -- CANNER: ms -- CANNER: under reviewing together...under maintenance.
(上為錯誤示範，輸出打架了，請盡量減少但此項不扣分，打架時通常會多空一行)
0ms -- CUTTER: under maintenance.
顯示切割工廠維護中
0ms -- CUTTER: under reviewing together...
兩個工廠都在休息，所以就一起進入檢討模式
1ms -- CANNER: under reviewing together...
1ms -- CUTTER: under reviewing together...
2ms -- CANNER: under reviewing together...
2ms -- CUTTER: under reviewing together...
工廠在維護或是一同檢討順序不定，上述就是沒有頻繁檢查是否忙碌的範例，會輸出一堆訊息，建議超過 10ms 以上閒置再檢查/輸出
...
5ms -- CUTTER: under reviewing together...
5ms -- Fish#1: waiting in the slot
顯示第一號魚進入備料區

5ms -- CANNER: under reviewing together...
6ms -- Fish#1: enters the CUTTER
顯示第一號魚進入切割場（座位釋放）
6ms -- CANNER: under maintenance.
6ms -- CUTTER: cutting... cutting... Fish#1 -- 15ms
第一號魚切割中
7ms -- CANNER: under maintenance.
...
13ms -- CANNER: under maintenance.
13ms -- Fish#2: waiting in the slot
14ms -- CANNER: under maintenance.

...
21ms -- CANNER: under maintenance.
21ms -- Fish#1: leaves CUTTER (complete 1st stage)
21ms -- Fish#2: enters the CUTTER
21ms -- CUTTER: cutting... cutting... Fish#2 -- 20ms
22ms -- CANNER: under maintenance.
22ms -- Fish#1: waiting in the slot (cutted)
顯示第一號魚切割完成回到備料區，標示為切割好了
第一條魚離開後，第二條魚接著處理
23ms -- Fish#3: waiting in the slot
23ms -- Fish1: enters to the factory (CANNER)
第一條魚進入罐頭工廠（會空出座位）
23ms -- CANNER: processing & canning the Fish#1 --99ms
31ms -- Fish#4: waiting in the slot
38ms -- Fish#5: waiting in the slot
41ms -- Fish#2: leaves CUTTER (complete 1st stage)
41ms -- Fish#3: enters the CUTTER
41ms -- CUTTER: cutting... cutting... Fish#3 -- 14ms
41ms -- Fish#2: waiting in the slot (cutted)
45ms -- Fish#6: waiting in the slot
54ms -- Fish#7: waiting in the slot
第三號魚還在切，2, 4~7 號在備料台等(2 號是等著送進裝罐頭)
55ms -- Fish#3: leaves CUTTER (complete 1st stage)
55ms -- Fish#4: enters the CUTTER
55ms -- CUTTER: cutting... cutting... Fish#4 -- 12ms
56ms -- Fish#3: waiting in the slot (cutted)
61ms -- Fish#8 has been sent to the Freezer - 48ms
66ms -- Fish#9 has been sent to the Freezer - 31ms
第四號魚還在切，2, 3, 5~7 號在備料台等(2,3 號是等著送進裝罐頭)
因此沒座位了，8 與 9 送進冷凍庫
68ms -- Fish#4: leaves CUTTER (complete 1st stage)
68ms -- Fish#5: enters the CUTTER
68ms -- CUTTER: cutting... cutting... Fish#5 -- 29ms

69ms -- Fish#4: waiting in the slot (cutted)
77ms -- Fish#10 has been sent to the Freezer - 36ms
97ms -- Fish#5: leaves CUTTER (complete 1st stage)
97ms -- Fish#6: enters the CUTTER
97ms -- CUTTER: cutting... cutting... Fish#6 -- 28ms
97ms -- Fish#5: waiting in the slot (cutted)
98ms -- Fish#9 has been sent to the Freezer - 39ms
109ms -- Fish#8 has been sent to the Freezer - 34ms
113ms -- Fish#10 has been sent to the Freezer - 40ms
123ms -- Fish1: leaves CANNER (Complete)
123ms -- Fish2: enters to the factory (CANNER)
第二號魚進入罐頭工廠
123ms -- CANNER: processing & canning the Fish#2 --57ms
125ms -- Fish#6: leaves CUTTER (complete 1st stage)
125ms -- Fish#7: enters the CUTTER
125ms -- CUTTER: cutting... cutting... Fish#7 -- 16ms
126ms -- Fish#6: waiting in the slot (cutted)
137ms -- Fish#9: waiting in the slot
因為第二號魚進入罐頭工廠，所以空出一個位置，九號可以進來等了
141ms -- Fish#7: leaves CUTTER (complete 1st stage)
141ms -- Fish#9: enters the CUTTER
141ms -- CUTTER: cutting... cutting... Fish#9 -- 29ms
142ms -- Fish#7: waiting in the slot (cutted)
143ms -- Fish#8 has been sent to the Freezer - 48ms
153ms -- Fish#10 has been sent to the Freezer - 43ms
171ms -- Fish#9: leaves CUTTER (complete 1st stage)
171ms -- CUTTER: under maintenance
171ms -- Fish#9: waiting in the slot (cutted)
172ms -- CUTTER: under maintenance
...
179ms -- CUTTER: under maintenance
180ms -- Fish2: leaves CANNER (Complete)
180ms -- Fish3: enters to the factory (CANNER)

180ms -- CANNER: processing & canning the Fish#3 --86ms
180ms -- CUTTER: under maintenance
...
191ms -- CUTTER: under maintenance
191ms -- Fish#8 has been sent to the Freezer - 45ms
192ms -- CUTTER: under maintenance
193ms -- CUTTER: under maintenance
195ms -- CUTTER: under maintenance
196ms -- CUTTER: under maintenance
197ms -- Fish#10 has been sent to the Freezer - 46ms
197ms -- CUTTER: under maintenance
...
236ms -- CUTTER: under maintenance
236ms -- Fish#8 has been sent to the Freezer - 32ms
237ms -- CUTTER: under maintenance
...
243ms -- CUTTER: under maintenance
243ms -- Fish#10 has been sent to the Freezer - 49ms
244ms -- CUTTER: under maintenance
...
266ms -- CUTTER: under maintenance
266ms -- Fish3: leaves CANNER (Complete)
266ms -- Fish4: enters to the factory (CANNER)
266ms -- CANNER: processing & canning the Fish#4 --57ms
267ms -- CUTTER: under maintenance
268ms -- CUTTER: under maintenance
268ms -- Fish#8: waiting in the slot
270ms -- Fish#8: enters the CUTTER
270ms -- CUTTER: cutting... cutting... Fish#8 -- 23ms
292ms -- Fish#10: waiting in the slot
293ms -- Fish#8: leaves CUTTER (complete 1st stage)
293ms -- Fish#10: enters the CUTTER
293ms -- CUTTER: cutting... cutting... Fish#10 -- 29ms

293ms -- Fish#8: waiting in the slot (cutted)
322ms -- Fish#10: leaves CUTTER (complete 1st stage)
323ms -- Fish#10: waiting in the slot (cutted)
第十號魚完成切割，切割工廠離線（之後不會再有維護訊息）
剩下就是魚料等著依序送進罐頭工廠
324ms -- Fish4: leaves CANNER (Complete)
324ms -- Fish5: enters to the factory (CANNER)
324ms -- CANNER: processing & canning the Fish#5 --57ms
381ms -- Fish5: leaves CANNER (Complete)
381ms -- Fish6: enters to the factory (CANNER)
381ms -- CANNER: processing & canning the Fish#6 --55ms
436ms -- Fish6: leaves CANNER (Complete)
436ms -- Fish7: enters to the factory (CANNER)
436ms -- CANNER: processing & canning the Fish#7 --84ms
521ms -- Fish7: leaves CANNER (Complete)
521ms -- Fish9: enters to the factory (CANNER)
521ms -- CANNER: processing & canning the Fish#9 --93ms
614ms -- Fish9: leaves CANNER (Complete)
614ms -- Fish8: enters to the factory (CANNER)
根據前述順序，本次 8 號會比 9 號晚一步送進罐頭工廠，
只要魚有送去冰，順序就可能改變
614ms -- CANNER: processing & canning the Fish#8 --52ms
666ms -- Fish8: leaves CANNER (Complete)
666ms -- Fish10: enters to the factory (CANNER)
666ms -- CANNER: processing & canning the Fish#10 --66ms
732ms -- Fish10: leaves CANNER (Complete)
732ms -- CANNER: under reviewing together...
工作完成！程式正常結束