**CS305作業系統概論 Prog. #3 Process Coordination 說明報告.docx**

**作者　：1043335 賴詩雨**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修改日期** | **說明** | **修改者** |
| **1.0.0.0** | **2017-05-29** | **初稿** | **賴詩雨** |
| **1.0.0.1** | **2017-05-31** | **修改** | **賴詩雨** |
| **1.0.0.2** | **2017-06-01** | **修改** | **賴詩雨** |
|  |  |  |  |

**目錄**

[第1章 程式完成部分: 2](#_Toc484119947)

[第2章 設計理念 2](#_Toc484119948)

[2.1 程式中特殊介紹 2](#_Toc484119949)

[2.2 結構圖 3](#_Toc484119950)

[2.3 Thread產生示意圖 4](#_Toc484119951)

[第3章 程式流程簡介 5](#_Toc484119952)

[第4章 程式輸出簡介 6](#_Toc484119953)

[第5章 程式如何編譯 7](#_Toc484119954)

[第6章 如何操作 7](#_Toc484119955)

1. 程式完成部分:
2. 基本功能
3. 進階功能
4. 設計理念

利用pthread的mutex機制形成critical section，使dispatcher 和 producer 能正確利用共用變數溝通。

* 1. 程式中特殊介紹

1. 有兩個結構，分別為 \_S\_Table 和 \_S\_Producer
2. **typedef** struct \_S\_Table

{

int quantity; // 空拍機組裝完成的數量

int count; // 桌上擁有零件的數量

int generate[3][2]; // 各個producer產生的數量

bool dispatch; // 為 1，代表 dispatcher產生完成

bool A\_or\_B; // 輪到 dispatcherA ，為 0

bool component[3]; // 放零件的地方

int DisCount[3][3]; // 計算 dispatcher 產生的個別零件

pthread\_mutex\_t lock;

} S\_Table, \*P\_Table;

1. **typedef** struct \_S\_Producer

{

int number; // 此 producer 擁有的零件號碼，對應 enum 定義

P\_Table ST;

} S\_Producer, \*P\_Producer;

1. 整個程式中有用到的結構宣告為

P\_Table pT; // 用於 dispatcher

P\_Producer pPRO**[**3**]**; // 用於三個 producer

1. 程式中 enum 定義

typedef enum {false, true} bool;

enum {battery, aircraft, propeller};

1. Function 說明

void producer( P\_Producer ptr ); // producer

void dispatcher( P\_Table ptr ); // 基本功能中的 dispatcher

void dispatcherA( P\_Table ptr ); // 進階功能中的 dispatcherA

void dispatcherB( P\_Table ptr ); // 進階功能中的 dispatcherB

* 1. 結構圖

1. \_S\_Table = dispatcher ( pT )

quantitydispatchcount A\_or\_B

generate[3][2]component[3]lock DisCount[3][3]

number

ST

number

ST

number

ST

\_S\_Producer pPRO**[**0**]** pPRO**[**1**]** pPRO**[**2**]**

1. **基本功能 進階功能**

**A\_or\_B**

**(bool)**

Dispatcher

Component[3] 零件放置區

**0**

**1**

**battery、aircraft**

DispatcherA

DispatcherB

0

1

Dispatch

(bool)

**搶自己缺少的，並組裝**

Producer

(battery)

Producer

(aircraft)

Producer

(propeller)

**battery、aircraft、**

**propeller**

**Lock**

**aircraft、propeller**

搶到lock才可進入紅色區域

* 1. Thread產生示意圖

結構

**EX :**

**pthread\_create (&proThread1, NULL, (void \*) &producer, pPRO[0]);**

主程式

**producer()**

**dispatcherB ()**

**dispatcherA ()**

**NOTE : pT 是指向共用結構區的指標**

**( p.3 的淺藍色區域 )**

**dispatcher ()**

1. 程式流程簡介
2. 在main設定好結構的基本需求以及初始lock後，會開始create三個producer ( pPRO[0]~pPRO[2] ) 。
3. 再來，根據命令列的數字( 1 or 2 )，決定要create一個( dispatcher )還是兩個dispatcher( dispatcherA, dispatcherB )。
4. 都create好了以後，main會等待thread執行。

[ case 1 ] : 一個 dispatcher

1. 一開始producer還不可以搶lock，因為桌上還沒有任何零件，所以dispatcher搶到lock。
2. Dispatcher會開始產生零件放到共用變數區(component[3])，若產生兩個零件了，則producer可以開始搶lock。
3. 搶到lock的producer去判斷自己能否組成空拍機，若不行則把lock釋出給其他的producer。
4. 當共用變數區的零件被消耗掉(零件少於兩個)，producer就不可以搶lock了，dispatcher搶到lock以後，可以繼續產生零件。
5. 重複執行5~7步驟，直到產生了50台空拍機。

[ case 2 ] : 兩個 dispatcher

1. 一開始producer還不可以搶lock，因為桌上還沒有任何零件，所以dispatcherA和dispatcherB去搶lock 。

Note: 為了確保公平競爭，布林值A\_or\_B會隨機獲得 0 or 1， 0就讓dispatcherA搶到lock。

1. 搶到的Dispatcher會開始產生零件放到共用變數區(component[3])，產生完一個零件後，dispatcherA和dispatcherB再次去搶lock，若產生兩個零件了，則producer可以開始搶lock。
2. 搶到lock的producer去判斷自己能否組成空拍機，若不行則把lock釋出給其他的producer。
3. 當共用變數區的零件被消耗掉(零件少於兩個)，producer就不可以搶lock了，dispatcher搶到lock以後，可以繼續產生零件。
4. 重複執行9~12步驟，直到產生了50台空拍機。
5. 擁有50台空拍機以後，main會

* 列印出dispatcher產生各種零件的數量
* 根據每個producer組成空拍機的數量，由大到小印出。

1. 程式輸出簡介

**黑字 => 會印出的訊息**

**藍字 => 程式的動作**

**--**-------------------------------------------------------------------

**[ case 1 ]**

**Dispatcher搶到lock，開始生產零件**

Dispatcher: propeller

Dispatcher: battery

**Producer們可以去搶lock，搶到的開始嘗試組裝**

Producer (aircraft): OK, 1 drone(s)

Dispatcher: aircraft

Dispatcher: propeller

Producer (battery): OK, 2 drone(s)

…

…

Producer (propeller): OK, 50 drone(s)

**Main印出dispatcher生產的各種零件數量**

**Main 根據組裝數量由小排到大並印出**

**[ case 2 ]**

**Dispatcher們去搶lock，搶到的開始生產零件**

DispatcherA: propeller

DispatcherB: battery

**Producer們可以去搶lock，搶到的開始嘗試組裝**

Producer (aircraft): OK, 1 drone(s)

DispatcherB: aircraft

DispatcherB: propeller

Producer (battery): OK, 2 drone(s)

DispatcherB: aircraft

DispatcherA: propeller

Producer (battery): OK, 3 drone(s)

…

…

Producer (propeller): OK, 50 drone(s)

**Main印出dispatcher生產的各種零件數量**

**Main 根據組裝數量由小排到大並印出**

1. 程式如何編譯
2. 有附上Makefile檔提供操作

OR

1. 在命令列輸入 : gcc 1043335\_03.c -lpthread -o 1043335\_03
2. 編譯成功後，輸入 : ./1043335\_03 1

**NOTE : 1 可改成 2，分別代表基本功能和進階功能**

1. 如何操作
2. 將 Makefile 中 run 的1 改成 2，可從基本功能切換成進階功能
3. make main => 編譯
4. make run => 執行
5. make start => 編譯 & 執行
6. make clean => 清除編譯、執行後產生的檔案
7. make all => 清除檔案後編譯執行
8. 或是可以直接在命令列輸入 **第五章** 提供的指令