**作業系統期末專案報告  
第14組**

103703013 資科三 黃育萱

103703015 資科三 蔡雨芝

103702012 心理三 陳冠聞

目錄

1. 報告實作主題、內容簡介
2. Thread類型與功能介紹
3. Multi-Thread之間的同步與合作
4. 關鍵區域程式碼
5. GUI呈現
6. Github與版本管理、分工
7. 組員心得
8. 參考資料

報告實作主題、內容簡介

主題：The Sleeping Barber (SB) Problem

語言：C ++ 11

內容：The barber shop has m barbers with m barber chairs, and n chairs (m < n) for waiting customers, if any, to sit in. If there are no customers present, a barber sits down in a barber chair and falls asleep. When a customer arrives, he has to wake up a sleeping

barber. If additional customers arrive while all barbers are cutting customers’ hair, they either sit down (if there are empty chairs) or leave the shop (if all chairs are full). The thread synchronization problem is to program the barbers and the customers without

getting into race conditions.

Thread類型與功能介紹

1. barberThread：  
   每個barber各自擁有的、獨立的thread，用來控管barber的狀態(sleeping or being busy)。
2. customerThread：  
   用來產生客人的thread

Multi-Thread之間的同步與合作

關鍵區域程式碼

void \*barberThread(void\* arg) {

int \*pID = (int\*)arg;

while(1) {

barMutex.lock();

cusMutex.lock();

if(totalServedCustomers < realNum\_customer){

cusMutex.unlock();

customers.wait(); // Try to acquire a customer.

//Go to sleep if no customers

Mutex.lock(); // Acquire access to waiting

//When a barber is waken -> wants to modify # of available chairs

barbers.signal(); // The barber is now ready to cut hair

int nowCut = nextCut;

nextCut = (nextCut+1) % NUM\_CHAIRS;

availableChairs++;

Mutex.unlock(); // Release waiting

/\* GUI change barber's mode \*/

isBusy[\*pID-1] = true;

cutting[\*pID-1] = waitingChairs[nowCut].data->cusID;

isSit[waitingChairs[nowCut].seqNumber] = false;

glutPostRedisplay(); //////////////GUI

cutHair(\*pID, waitingChairs[nowCut]); //pick the customer which counter point

isBusy[\*pID-1] = false;

glutPostRedisplay(); //////////////GUI

}

else{

barMutex.unlock();

cusMutex.unlock();

break;

}

}

}

關鍵區域程式碼(cont.)

void \*customerThread(void\* arg) {

struct customerData \*data = (struct customerData\*)arg;

Mutex.lock(); // Acquire access to waiting

if( availableChairs == 0 ) {

comeCus[data->cusID-1] = false;

cusMutex.lock();

--realNum\_customer;

cusMutex.unlock();

Mutex.unlock();

pthread\_exit(0);

}

comeCus[data->cusID-1] = false;

seat[nextSit] = data->cusID;

waitingChairs[nextSit].data = data;

nextSit = (nextSit+1) % NUM\_CHAIRS;

availableChairs--;

showWhoSitOnChair();

glutPostRedisplay(); //////////////GUI

customers.signal(); // Wake up a barber (if needed)

Mutex.unlock(); // Release waiting

barbers.wait(); // Go to sleep if number of available barbers is 0

waitForHairCut(data);

}

Github與版本管理、分工

# Version 1：<pthread.h>(適用Windows/ Ubuntu(Mac OS有編譯問題))

[SB01~SB10.cpp]  
虛擬碼撰寫：陳冠聞  
c++程式碼撰寫、優化：陳冠聞、蔡雨芝

# Version 2：Windows -> Mac OS(解決Mac OS編譯問題)

[SB11.cpp] 程式碼轉換：黃育萱

# Version 3：程式碼優化、轉換為適用於Windows版本

[SB12、SB13win.cpp] 程式碼優化：陳冠聞、蔡雨芝

# Version 4：GUI (using Qt)

[SB13.cpp] GUI程式碼：黃育萱

# Version 5：<thread> & <mutex>(適用於Windows/Mac OS/Ubuntu)

[SB15.cpp] 程式碼轉換：蔡雨芝

# Version 6 (Final)：GUI (using OpenGL)

[SB16.cpp] GUI程式碼：黃育萱

組員心得

# 103703013 資科三 黃育萱：

# 103703015 資科三 蔡雨芝：

# 103702012 心理三 陳冠聞：