Document Name		Confidentiality Level
Theater Ticketing System		Only for Recipients' Reference
Template Code	Version	Document Code
BP	1.0	BP

Theater Ticketing System

——Thread Synchronization and Mutex

Copyright Declaration

Contents

1 Teaching Tips

(1) Learn about the definition of the thread.

(2) Understand the status of the thread.

(3) Understand the synchronization and mutex of the thread.

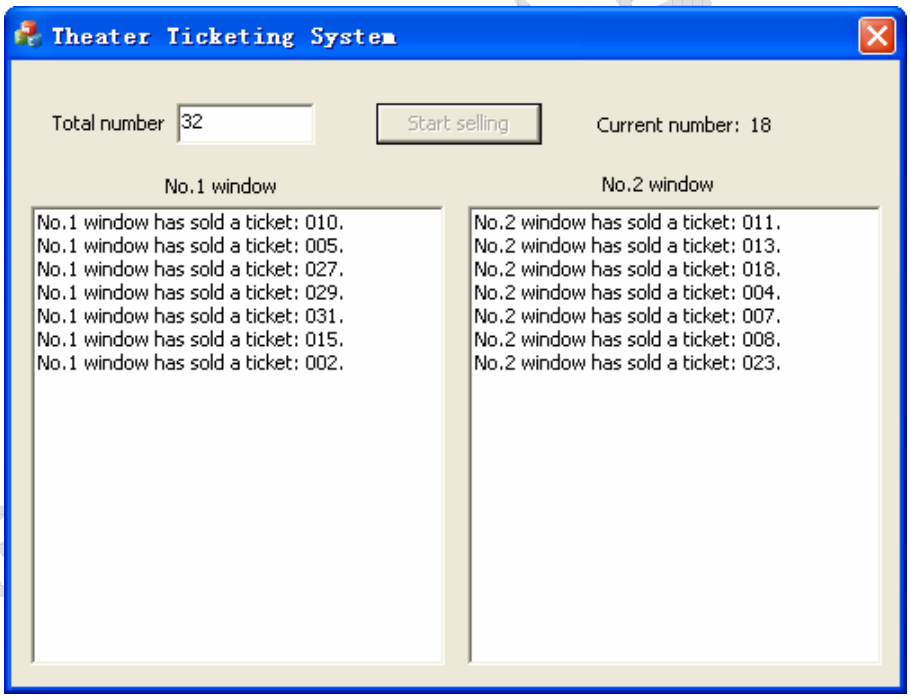
(4) Master how to create the thread.

(5) Master the common thread functions.

(6) Create MFC Dialog program. Develop “Theater Ticketing System” program with the thread synchronization and mutex technologies.

2 Function Requirements

A theater has some movie tickets. The movie tickets will be sold in two windows simultaneously until they have been sold out. We will develop a MFC Dialog program to simulate the ticket sale process of the theatre. The program interface is as follows:



Requirements:

1. Click “Start selling”. Two windows will start selling tick simultaneously without disturbing each other.

2. Display the ticket sale processes of these two windows.

3. Save the ticket sale records of these two windows into a log file sale.txt.

The format of the log file sale.txt is as follows:

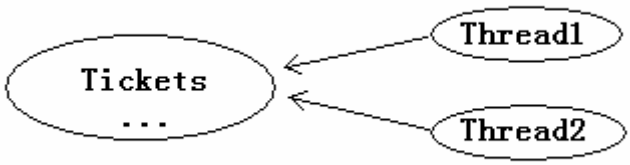
```
No.2 window has sold a ticket: 011.
```



```
No.1 window has sold a ticket: 010.
No.1 window has sold a ticket: 005.
No.2 window has sold a ticket: 013.
No.2 window has sold a ticket: 018.
No.1 window has sold a ticket: 027.
...
```

3 Design Ideas

With Mircosoft Visual Studio 2010 development tool, create MFC Dialog project. With thread synchronization and mutex technologies, develop “Theatre Ticketing System”. The project name is SellTickets.

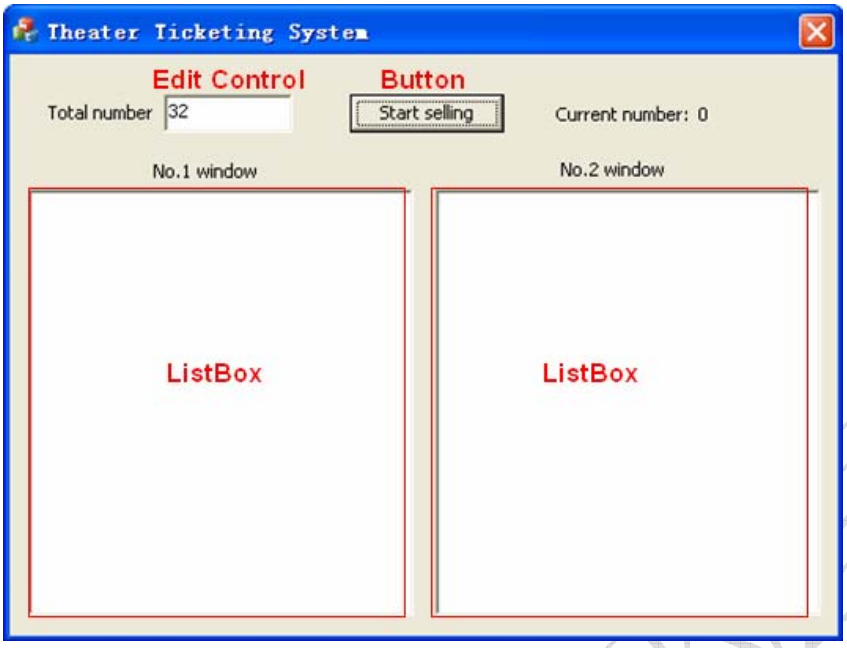


Create a CTicketBiz class that is used to manage the movie tickets. With CreateThread() function, create two ticket sale threads SellThread. Access the same CTicketBiz object. This object is the sharing resource of these two threads.

Define critical section object CRITICAL_SECTION g_csTicket. And manage the synchron ization and mutex of the thread with thread function InitializeCriticalSection().EnterCritical Section().LeaveCriticalSection().

3.1 Interface Design

1. Interface



2. Interface Description

(1) Create MFC Dialog as the main window of the program.

(2) With Edit control, receive the ticket number input by the user.

(3) With Button control, create OnBnClickedButtonSellTicket() function, and response "Start Ticket Sale" operation.

(4) With ListBox control, simulate two ticket windows to display the ticket sales of the windows.

(5) With Timer control, create OnTimer() function to update the interface regularly, and display the current ticket number of the system.

3.2 Data Structure

1. Movie Ticket Information

In CTicketBiz class, define dynamic array int *m_pTickets to represent all the movie ticket information. When "Start selling", according to the ticket number input by the user, initialize this array, save the number of each ticket.

2. Ticket Selling Thread Structure

Save the following information into the ticket sale thread. So that, threads can share the movie ticket data through CTicketBiz object, and update the ticket windows on the interface.

```
// Ticket sale thread parameter structure
struct SellThreadParam{
    CTicketBiz* biz;
    CListBox* wndShow;  // Point to the ticket window
    int win_no;         // Ticket window number
```

```
    bool service;       // Ticket sale status
};
```

3. Log File

The ticket sale thread will output the ticket sale records into the log file. According to the requirements, the format of sale.txt is as follows:

```
No.%d window has sold a ticket: %03d.\n
```

3.3 CTicketBiz Class

1. Introduction

Ticket management class is used to save the movie ticket information.

2. Data Member

Type	Name	Description
int*	m_pTicket	Point to the array that saves the ticket information
int	m_nTotalNum	Total ticket number
int	m_nSoldNum	Sold ticket number
int	m_nBalanceNum	Remaining ticket number

3. Member Function

Function	Description
void GenerateTicket(int)	Generate the ticket. Initialize the movie ticket array.
int GetRandTicket()	Get a ticket randomly
int GetBalanceNum()	Get the remaining ticket number

4 Implementation Idea

4.1 Create Project

1. Create Project
2. Interface Layout
3. Compile and Run

4.2 Define CTicketBiz Class

1. Define CTicketBiz class

2. Define GenerateTicket() function

3. Define GetRandTicket() function

4. Define GetBalanceNum() function

4.3 Create Ticket Sale Thread

1. Define thread parameter SellThreadParam

2. Define ticket sale thread function SellTicketThreadProc()

3. Create the thread.

4. Compile and run

4.4 Display Remaining Ticket Number

1. Create the timer.

2. Update the current ticket number.

3. Compile and run