

实验 1 线程的创建与撤销

一、实验目的

通过本实验熟悉 Windows 系统提供的线程创建与撤销等系统调用，掌握 Windows 系统环境下线程的创建与撤销方法。

二、实验内容

1. 熟悉开发环境 Visual C++ 6.0;
2. Windows 系统环境下线程的创建与撤销方法;
3. 利用 2 个线程模拟火车票销售系统。

三、实验准备知识

1. 线程创建函数

```
HANDLE CreateThread(  
    LPSECURITY_ATTRIBUTES lpThreadAttributes,  
    SIZE_T dwStackSize,  
    LPTHREAD_START_ROUTINE lpStartAddress,  
    LPVOID lpParameter,  
    DWORD dwCreationFlags,  
    LPDWORD lpThreadId  
);
```

2. 线程撤销函数

```
VOID ExitThread(  
    DWORD dwExitCode  
);
```

3. 线程终止函数

```
BOOL TerminateThread(  
    HANDLE hThread,  
    DWORD dwExitCode  
);
```

4. 线程挂起函数

```
VOID Sleep(  
    DWORD dwMilliseconds  
);
```

5. 关闭句柄函数

```
BOOL CloseHandle(  
    HANDLE hObject  
);
```

这些函数的详细说明请查阅 MSDN。

四、实验步骤

1. Windows 系统环境下线程的创建与撤销方法

该程序段在主线程中调用 `CreateThread()` 创建 2 个子线程，并在子线程中显示类似 “Thread is running!” 等字样。其主要代码如下：

```
#include <windows.h>
#include <iostream.h>

int index = 0;

DWORD WINAPI Thread1Proc(
    LPVOID lpParameter
);
DWORD WINAPI Thread2Proc(
    LPVOID lpParameter
);

int main()                                // 主线程
{
    HANDLE hThread1;
    HANDLE hThread2;
    hThread1 = CreateThread(NULL, // 创建线程 1
        0,
        Thread1Proc,
        NULL,
        0,
        NULL);
    hThread2 = CreateThread(NULL, // 创建线程 2
        0,
        Thread2Proc,
        NULL,
        0,
        NULL);
    CloseHandle(hThread1);                // 关闭句柄
    CloseHandle(hThread2);
    while(index++<1000)
    {
        cout<<"main thread is running!"<<endl;
    }
    ExitThread(0);    // 撤销线程，参数 0 表示要撤销进程中的所有线程；缺省亦可
    return 0;
}
```

```

}

DWORD WINAPI Thread1Proc(
    LPVOID lpParameter
)
{
    while(index++<1000)
    {
        cout<<"Thread 1 is running!"<<endl;
    }
    return 0;
}

DWORD WINAPI Thread2Proc(
    LPVOID lpParameter
)
{
    while(index++<1000)
    {
        cout<<"Thread 2 is running!"<<endl;
    }
    return 0;
}

```

2. 利用 2 个线程模拟火车票销售系统 1

该程序段利用上例代码实现利用 2 个线程模拟火车票销售系统,其主要代码如下所示。需要注意的是该段程序是含有隐患的,注意考虑其隐患所在。

```

#include <windows.h>
#include <iostream.h>

```

```

    Thread1Proc,
    NULL,
    0,
    NULL);
hThread2 = CreateThread(NULL, // 创建线程 2
    0,

```

```

        Thread2Proc,
        NULL,
        0,
        NULL);
CloseHandle(hThread1);          // 关闭句柄
CloseHandle(hThread2);
Sleep(5000);
ExitThread(0);    // 撤销线程，参数 0 表示要撤销进程中的所有线程；缺省亦可
return 0;
}

DWORD WINAPI Thread1Proc(
    LPVOID lpParameter
)
{
    while(TRUE)
    {
        if(tickets>0)
            cout<<"Thread 1 sell ticket : "<< tickets-- <<endl;
        else
            break;
    }
    return 0;
}

DWORD WINAPI Thread2Proc(
    LPVOID lpParameter
)
{
    while(TRUE)
    {
        if(tickets>0)
            cout<<"Thread 2 sell ticket : "<< tickets-- <<endl;
        else
            break;
    }
    return 0;
}

```

3. 利用 2 个线程模拟火车票销售系统 2

该段程序亦是实现利用 2 个线程模拟火车票销售系统，其主要代码如下所示。

其主作用在于将上例中可能出现的隐患展现出来，注意观察两段程序的不同之

其主要作用在于将上例中可能出现的隐患展现出来。注意观察两段程序的不同之处，思考隐患出现的原因，并考虑避免隐患的方法。

```
#include <windows.h>
#include <iostream.h>

int tickets = 100;

DWORD WINAPI Thread1Proc(
    LPVOID lpParameter
);
DWORD WINAPI Thread2Proc(
    LPVOID lpParameter
);

int main()                                // 主线程
{
    HANDLE hThread1;
    HANDLE hThread2;
    hThread1 = CreateThread(NULL, // 创建线程 1
        0,
        Thread1Proc,
        NULL,
        0,
        NULL);
    hThread2 = CreateThread(NULL, // 创建线程 2
        0,
        Thread2Proc,
        NULL,
        0,
        NULL);
    CloseHandle(hThread1);                // 关闭句柄
    CloseHandle(hThread2);
    Sleep(5000);
    ExitThread(0);    // 撤销线程，参数 0 表示要撤销进程中的所有线程；缺省亦可
    return 0;
}

DWORD WINAPI Thread1Proc(
    LPVOID lpParameter
)
```

```
{
```

```

while(TRUE)
{
    if(tickets>0)
    {
        Sleep(1);
        cout<<"Thread 1 sell ticket : "<< tickets-- <<endl;
    }
    else
        break;
}
return 0;
}

DWORD WINAPI Thread2Proc(
    LPVOID lpParameter
)
{
    while(TRUE)
    {
        if(tickets>0)
        {
            Sleep(1);
            cout<<"Thread 2 sell ticket : "<< tickets-- <<endl;
        }
        else
            break;
    }
    return 0;
}

```

五、作业

1. 熟悉 Windows 系统环境下线程的创建与撤销方法。
2. 实现利用 2 个线程模拟火车票销售系统。
3. 请指出火车票销售模拟系统的隐患所在，及解决方法。