**实验二 多线程编程实验**

**一、实验目的**

1. 掌握线程的概念，明确线程和进程的区别。

2. 学习Linux下线程创建方法及编程。

3. 了解线程的应用特点。

4． 掌握用锁机制访问临界区。

**二、实验内容**

1. 通过多线程模拟多窗口售票,在主线程下创建4个子线程，模拟4个售票窗口，假设有20张票待售，运行该程序看会有什么样的结果，分析原因。**（2分）**

参考程序

#include<pthread.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

int ticket\_sum=20;

void \*sell\_ticket(void \*arg)

{

int i;

for(i=0;i<20;i++)

{

if(ticket\_sum>0)

{

sleep(1);

printf("sell the %dth\n",50-ticket\_sum+1);

ticket\_sum--;

}

}

return 0;

}

int main()

{

int flag,i;

pthread\_t tids[4];

for(i=0;i<4;i++)

{

flag=pthread\_create(&tids[i],NULL,&sell\_ticket,NULL);

if(flag)

{

printf("pthread create error ,flag=%d",flag);

return flag;

}

}

sleep(20);

void \*ans;

for(i=0;i<4;i++)

{

flag=pthread\_join(tids[i],&ans);

if(flag)

{

printf("tid=%lu,join erro flag=%d",tids[i],flag);

return flag;

}

printf("ans=%d\n",(int)ans);

}

return 0;

}

运行结果：（部分截屏）

分析原因：

1. 修改第1题，显示哪个窗口卖出的票（显示线程ID）。**（2分）**
2. 修改上题，用锁机制实现线程互斥进入临界区。 **（3分）**