

北京邮电大学

现代交换原理实验报告



实验名称：摘挂机检测实验

班 级：2016211301

学 号：2016211134

姓 名：李智盛

指导教师：丁玉荣

实验日期：2019 年 5 月 8 日

0. 目录

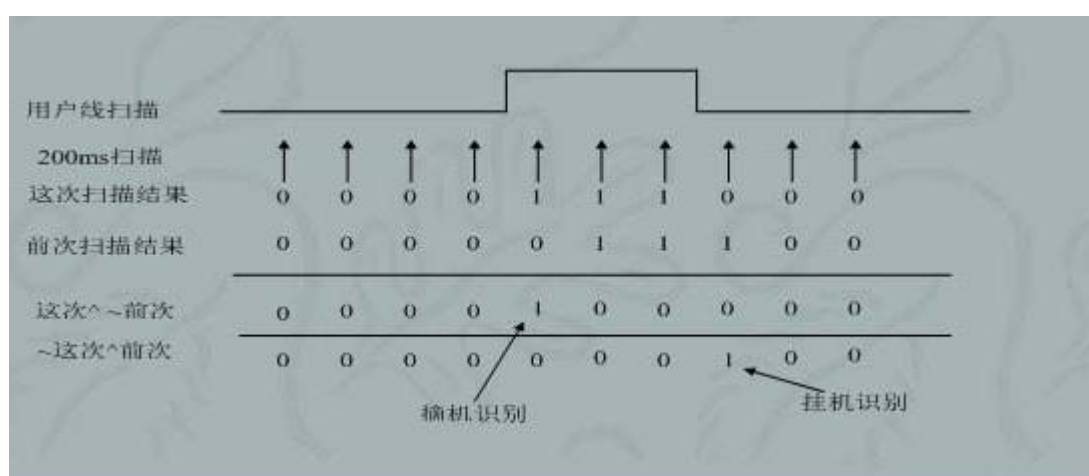
0. 目录	1
1. 实验目的	2
2. 实验内容和实验步骤	2
3. 源代码	4
4. 实验结果	5
5. 实验小结	5

1. 实验目的

摘挂机检测实验用来考查学生对摘挂机检测原理的掌握情况

2. 实验内容和实验步骤

设用户在挂机状态时扫描输出为“0”，用户在摘机状态时扫描输出为“1”，摘挂机扫描程序的执行周期为200ms，那么摘机识别，就是在200ms的周期性扫描中找到从“0”到“1”的变化点，挂机识别就是在200ms的周期性扫描中找到从“1”到“0”的变化点，该原理的示意图如下所示：



在我们的实验中，我们把前200ms的线路状态保存以备这次可以读取，同时读出这次的线路状态，把前200ms的线路状态取反与这次的线路状态相与，如果为1，就说明检测到摘机消息了。同理，我们把这次的线路状态取反再与前200ms的线路状态相与，如果为1就说明检测到挂机消息了，然后把摘挂机信号作为事件放入摘挂机队列中。

实验主要数据结构如下：

函数功能为：检测到摘、挂机事件，并把该事件放入到摘挂机事件队列中。

函数原型：void scanfor200(int linestate200[LINEMAX],int linestate[LINEMAX],UpOnnode * head1, UpOnnode* end1); 其中LINEMAX为线路总数，是定义在"bconstant.h"中的一个宏,linestate200[LINEMAX]为已保存的200ms前线路状态，linestate[LINEMAX]为当前的线路状态，head1,end1为摘挂机队列的首尾指针，该队列已经在主程序中进行了初始化。我们所要做的就是检测到的摘挂机事件以摘挂机队列节点的形式插入到摘挂机事件队列中。

头文件: "bconstant.h";(以下的数据结构都已在该文件中定义)

LINEMAX : 最大线路数;

int linestate200[LINEMAX],linestate[LINEMAX]: 线路从 0 开始编号; 状态: 1: 有电流, 0 无电流;

enum UporOn {ehandup,ehandon}:为摘挂机区别符: ehandup 表示摘机, ehandon 表示挂机;

```

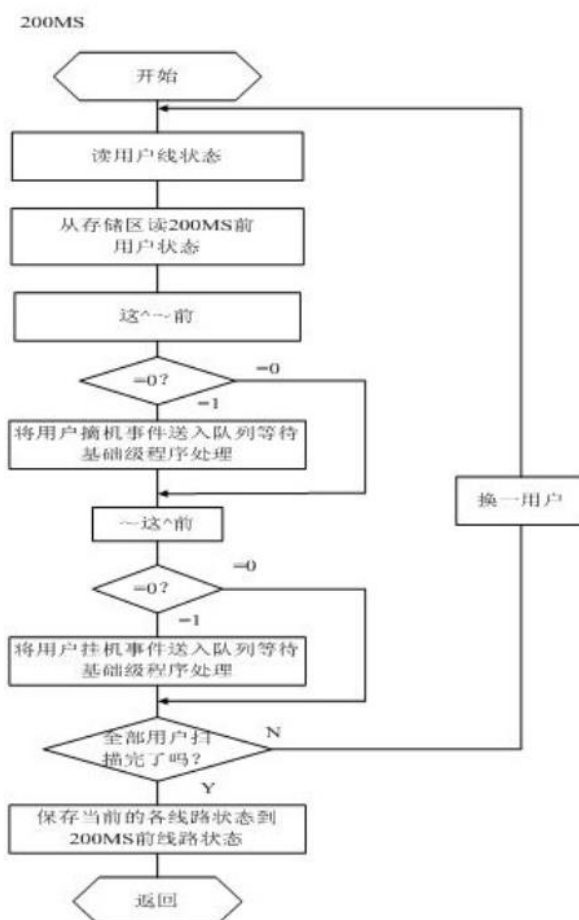
struct UpOnnode           //摘挂机队列节点结构
{
    UporOn phonestate;      //摘挂机区别符;

    int linenum;            //线路号 (从 0 开始);

    struct UpOnnode* next;  //指向下一节点的指针;
};

```

实验主体流程图



3. 源代码

```

#include "bconstant.h"
extern "C" _declspec(dllexport)
void scanfor200(int linestate200[LINEMAX], int linestate[LINEMAX], UpOnnode *
head1, UpOnnode * end1)
{
    int i;
    for(i=0;i!=LINEMAX;i++)
    {
        if(linestate[i]==1&&linestate200[i]==0)
        {
            struct UpOnnode *newNode = new struct UpOnnode;
            newNode->phonestate =ehandup;
            newNode->linenum = i;
            newNode->next=end1->next;
            end1->next = newNode;
            end1 = newNode;
        }

        if(linestate[i] == 0 &&linestate200[i]==1)
        {
            struct UpOnnode*newNode = new struct UpOnnode;
            newNode->phonestate =ehandup;
            newNode->linenum = i;
            newNode->next=end1->next;
            end1->next = newNode;
            end1 = newNode;
        }
        linestate200[i]=linestate[i];
    }
    return;
}

extern "C" _declspec(dllexport) void freenode(UpOnnode * node)
{
    delete node;
}

```

4. 实验结果

摘挂机检测正确，能够进行正常的通话。

5. 实验小结

本次实验是交换原理课程的第二次实验，用来考察我们对摘挂机检测的原理的掌握情况。在实验中，我们需要以 200ms 为周期依次对 4 根线路进行检测，通过判断当前线路状态和 200ms 前线路状态来检测是否有摘挂机事件发生；若检测到有摘挂机事件发生，则修改该事件的状态，并将该事件添加到事件队列的末尾；最后再将 200ms 前线路状态更新为当前线路状态。

本次实验的代码很简单，类似于对链表的操作。通过编码和实验，我们在仿真环境下模拟了摘挂机的检测以及完整的通话过程，加深了对这部分原理的理解，相信对日后的学习也会有帮助。