《现代交换原理》实验报告

实验名称　　时间表调度实验和摘挂机检测实验

班　　级　　2020211314

学　　号　　2020211502

姓　　名　　王小龙

指导教师　　赵　学　达

实验一　　时间表调度实验

一、实验目的

通过驱动交换网络实验用来考查学生对时间表调度原理的掌握情况。

二、实验内容和实验步骤（简写）

实验内容：

在程控数字交换的体系结构中，周期级程序（例如摘挂机检测程序、脉冲识别程序、位间隔识别程序）是由时间表调度实现的。所谓时间表调度，是指每经过交换系统的最短有效时间（这通常是指各周期性程序周期的最大公约数），都会检查调度表的调度要求，如果某个程序在这时需要执行，则调度程序开始执行它。

在我们设计的时间表调度实验中，这个调度表的调度是静态的。所谓静态，是指我们的调度表是在系统初始化的时候就建立起来的，在系统运行的情况下不再改动。 实验要求的就是这个调度表的初始化。

我们这个交换系统提供了三个周期性调度程度（摘挂机检测程序、脉冲识别程序和位间隔识别程序），它们的调用周期分别为 200ms、10ms 和 100ms，所以我们系统的最小调度时间为 10ms。如图所示，每隔 10ms,我们就会检查这个表的一行，如果该行上某一列为 1，我们就执列所对应的任务，如果为 0，就什么都不做。每当执行到这个表的最后一行，调度任务会返回第一行循环执行。而你所要做的就是按照你的理解来填写这个调度表。

实验步骤如下：

1、理解实验要求和所给数据结构定义

通过上述实验内容可以知道，我们要初始化一个二维数组用作调度表，在二维数组中每一个值代表是否在当前时刻执行对应任务，如果值为 1，则代表执行，如果值为 0，则代表不执行，表中每一行相当于 10ms 的时间段，每一列代表一种类型的任务，一共有三种任务，调用周期分别为 200ms、10ms 和 100ms，并且如果执行到这个表的最后一行，调度任务会返回第一行循环执行。

由所给数据结构可知，SchTabLen 代表这个调度表为 20 行，相邻行间隔 10ms;

SchTabWdh 代表这个调度表为 3 列，共三个任务。

实验主要数据结构：

函数功能：完成调度表的初始化；

函数原型：initSchTable(int ScheduleTable[SchTabLen][SchTabWdh])；

其中SchTalLen和SchTabWdh为在bconstant.h中的宏定义：

```
#define SchTabLen 20    //代表这个调度表为20行（相邻行之间的时间间隔为10ms)；
#define SchTabWdh 3     //代表三个周期性调度任务——0：摘挂机检测任务；1：脉冲检测任务；2：位间隔检测任务；
```

2、编写代码

函数主体就是一个 for 循环，通过判断当前循环变量的值是否满足一定要求来决定是否要修改二维调度表的值。由于每隔 10ms 扫描一次，而表中相邻行间隔也是 10ms，所以每 20 行就要对第一列的位置设置 1，每 1 行就要对第二列的位置设置 1，每 10 行就对第三列的位置设置 1。

三、源代码（注明代码含义）

函数代码如下：

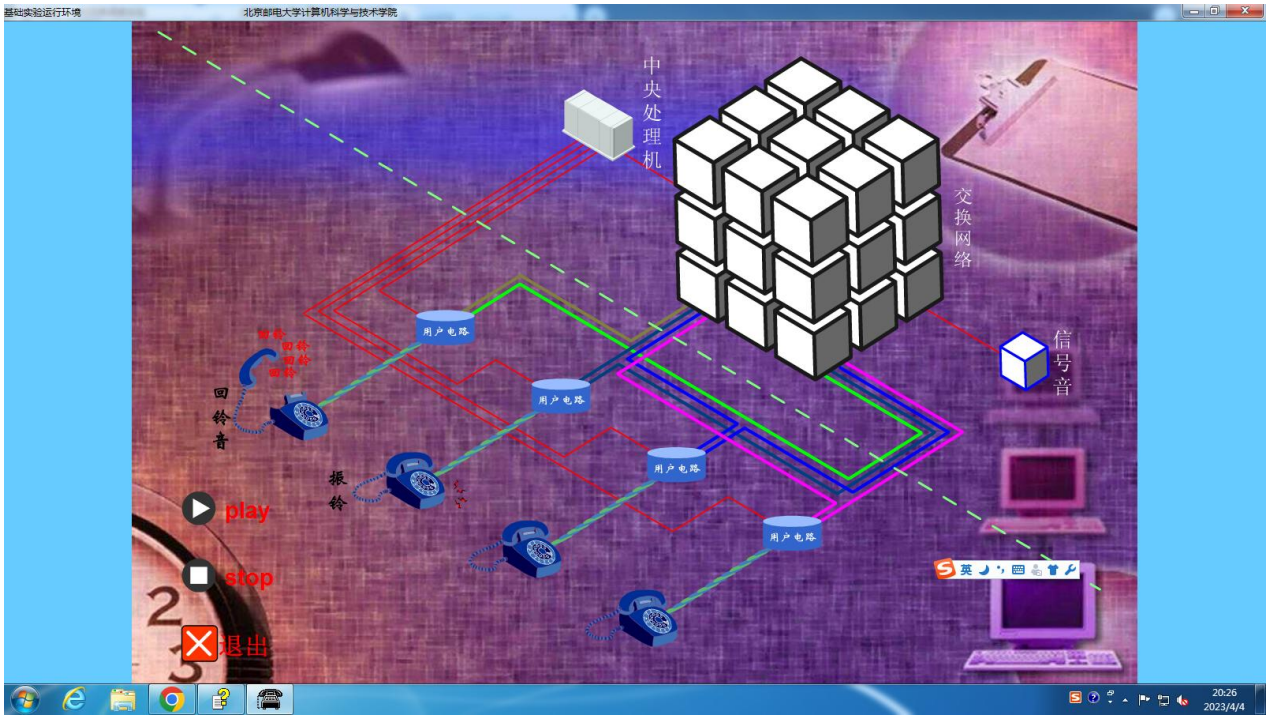
基础实验一：时间表调度实验　　北京邮电大学计算机科学与技术学院

选择实验　源文件编辑　编译运行　帮助

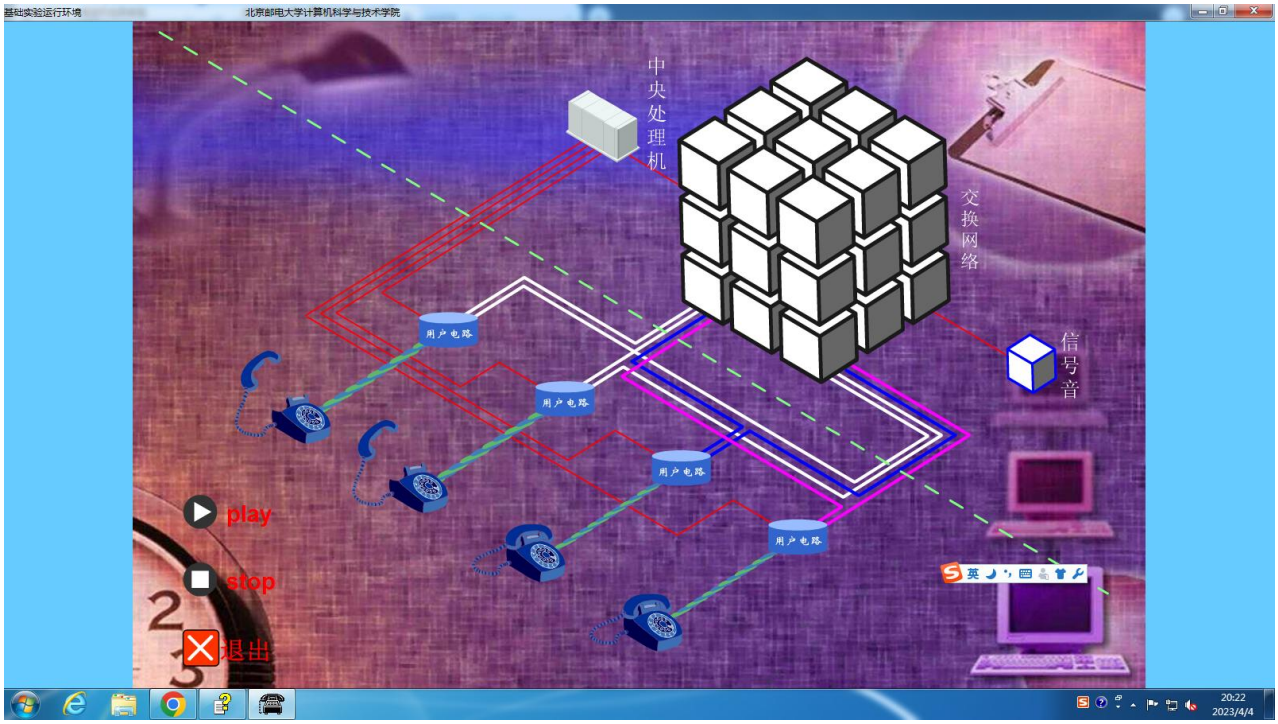
```
#include "bconstant.h"

extern "C" _declspec(dllexport) void initSchTable(int ScheduleTable[SchTabLen][SchTabWdh]) {
   for (int i = 0; i < SchTabLen; i++) {
      if (i % 20 == 0) {   //程序一
         ScheduleTable[i][0] = 1;
      }
      else {
         ScheduleTable[i][0] = 0;
      }

      ScheduleTable[i][1] = 1;  //程序2

      if (i % 10 == 0) {          //程序3
         ScheduleTable[i][2] = 1;
      } else {
         ScheduleTable[i][2] = 0;
      }
   }
   return;
}
```

四、实验结果

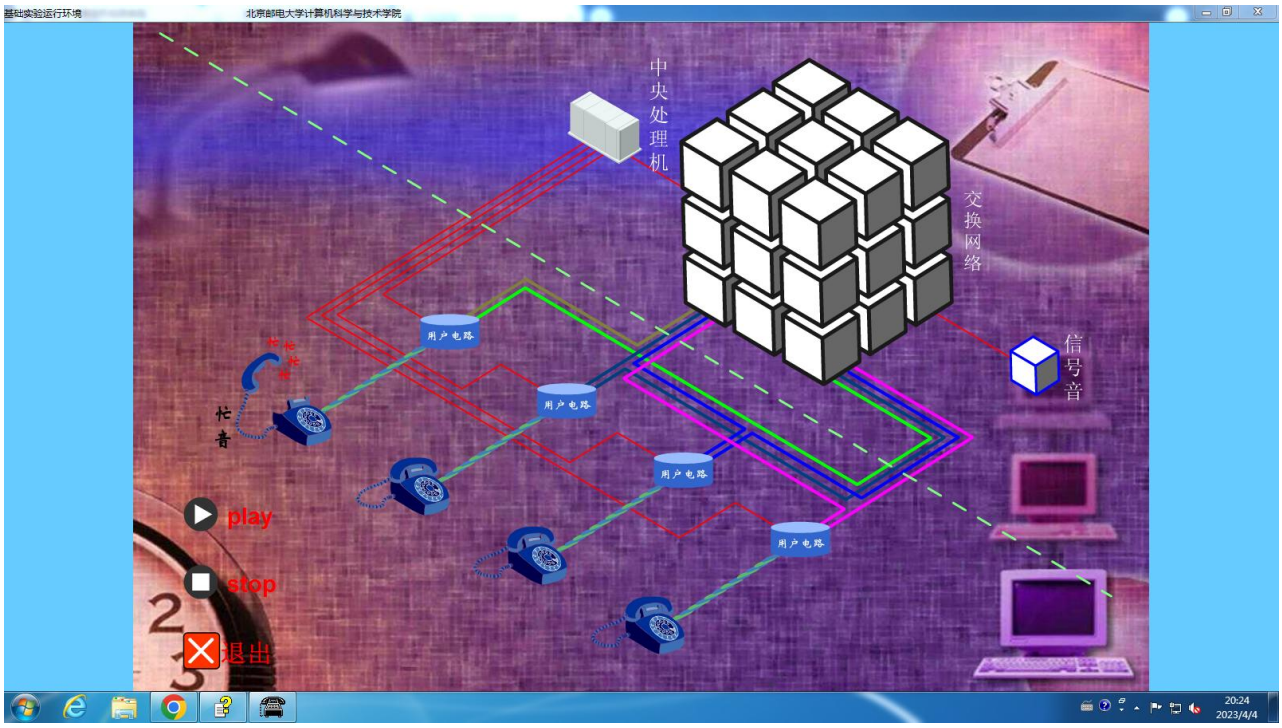
在下述演示中，电话 1 拨打电话 2 的号码，可以看到电话 2 拿起话筒时，显示振铃，即拨号成功：



电话 2 拿起话筒，线路接通：



电话 2 挂断：



五、实验心得

通过本次实验，增强了我对时间表调度原理的掌握，并且了解到了周期级程序（例如摘挂机检测程序、脉冲识别程序、位间隔识别程序）是由时间表调度实现的。在这次实验中，我不仅锻炼了编写代码的能力，同时也增长了很多关于时间调度表的知识，收获很多。

实验二　摘挂机检测实验

一、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。

二、实验内容和实验步骤（简写）

实验内容：

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。

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。

实验步骤：

1、理解实验要求和所给数据结构定义

通过上述实验内容可以知道，我们需要编写一个函数来将前200ms的线路状态读取出来，同时读出这次的线路状态，进行一定的操作，来判断摘挂机消息，并将该消息通过节点的形式放入队列中。

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。我们所要做的就是把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。

数据结构说明：

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；
   };
```

主要的数据结构如上，我们可以看到，linestate200这个数组存储了200ms前线路状态，linestate存储了当前的线路状态，UpOnnode结构为节点结构，有了这些数据结构，我们就可以开始编码了。

2、编写代码

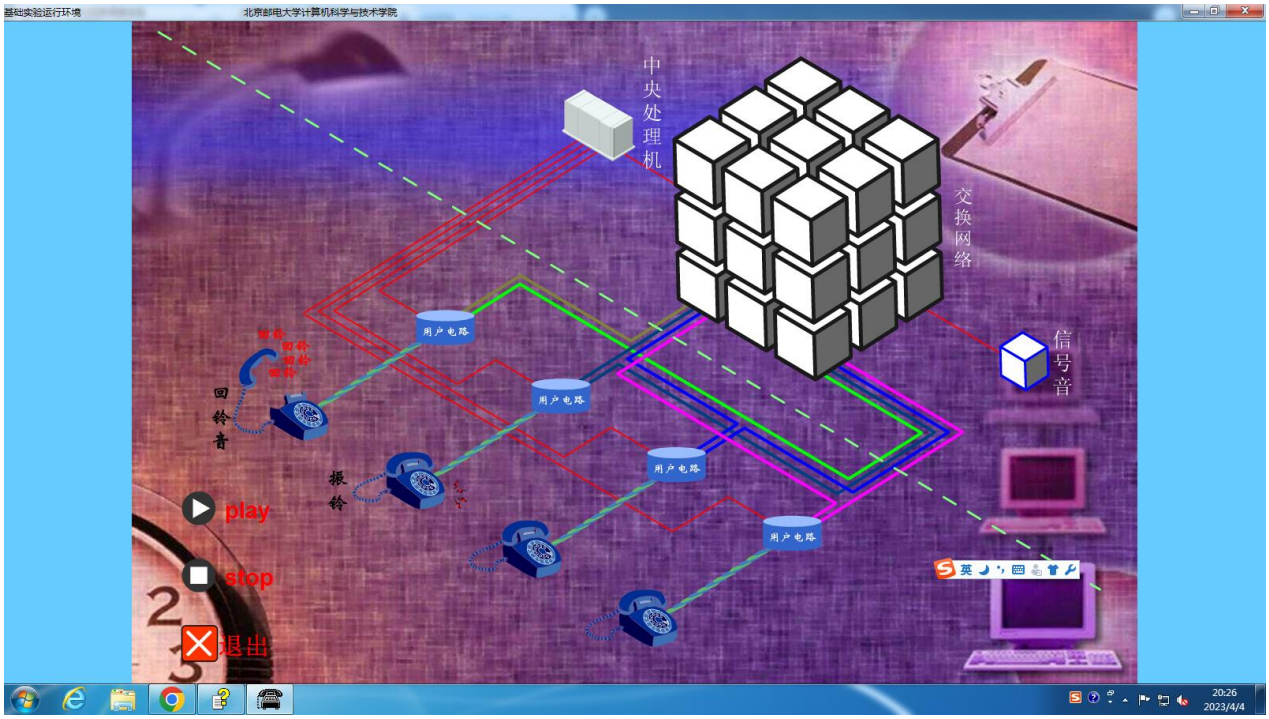
200ms前为0，现在是1，则表示摘机，所以把前200ms的线路状态取反与这次的线路状态相与，如果为1，就说明检测到摘机消息了；200ms前为1，现在是0，则表示挂机，把这次的线路状态取反再与前200ms的线路状态相与，如果为1就说明检测到挂机消息了。然后把对应信息，以节点的方式插入到链表中。

三、源代码（注明代码含义）

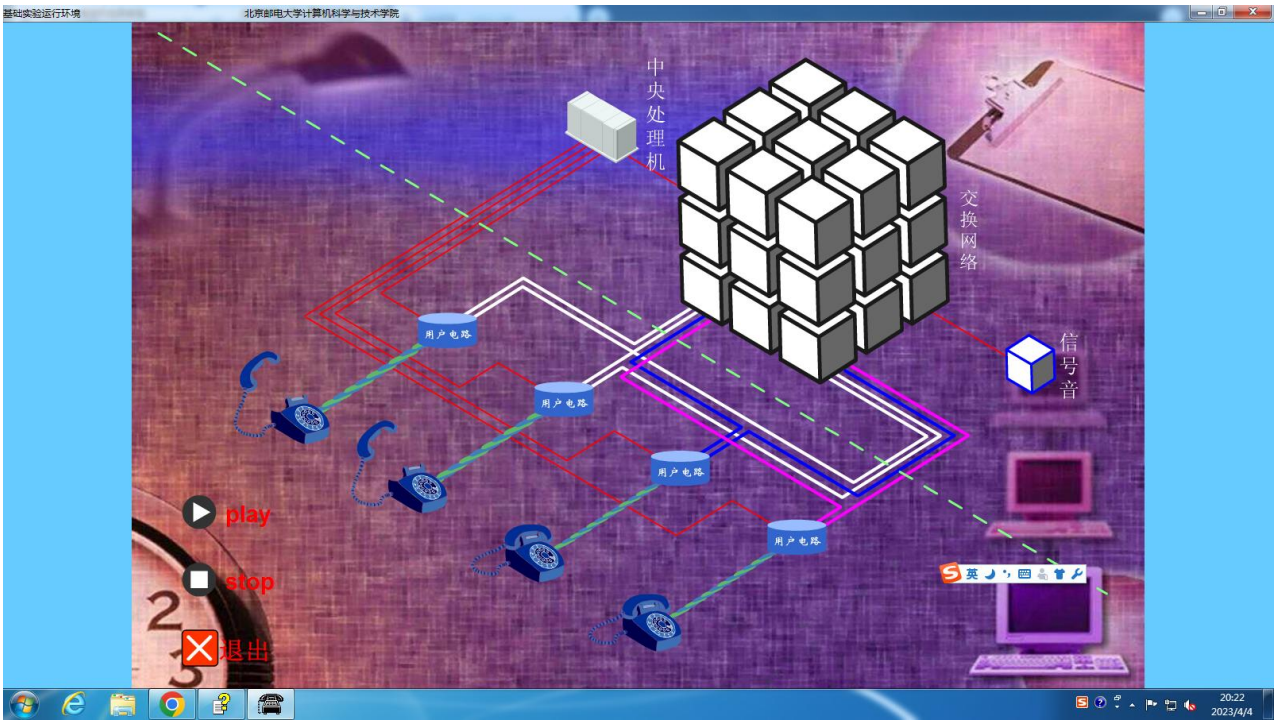
函数代码如下：

四、实验结果

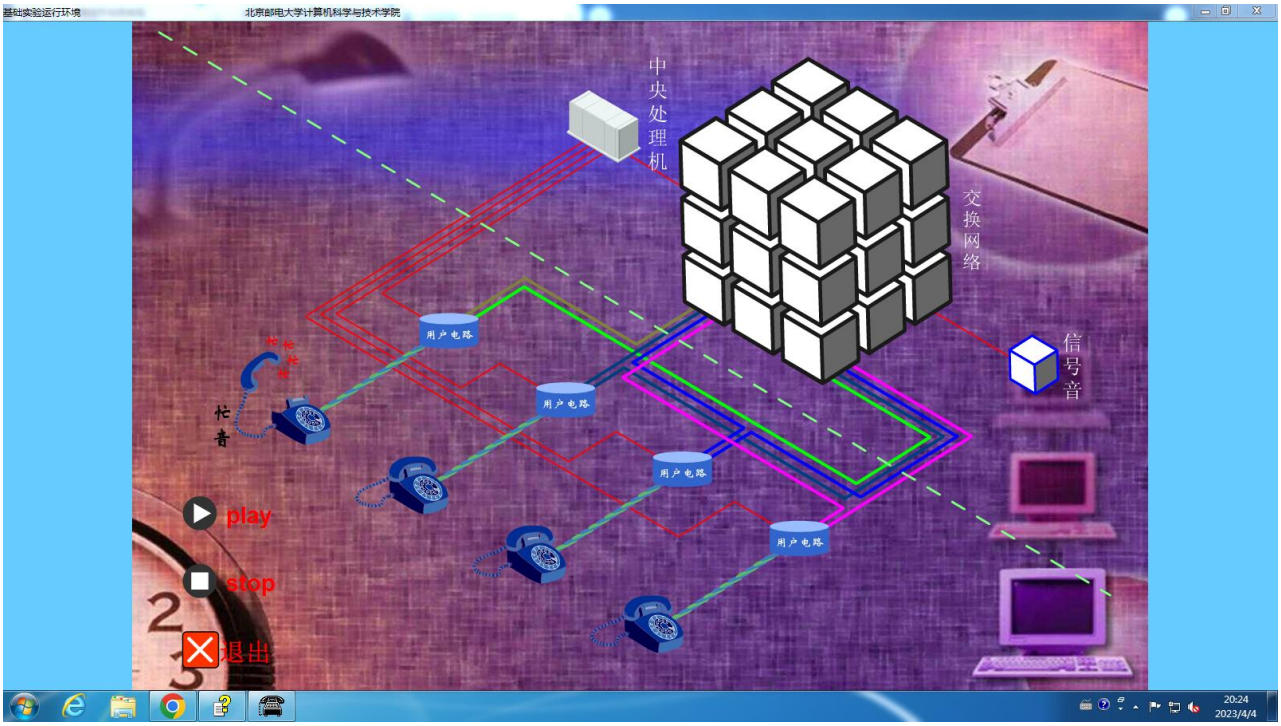
在下述演示中，电话 1 拨打电话 2 的号码，可以看到电话 2 拿起话筒时，显示振铃，即拨号成功：



电话 2 拿起话筒，线路接通：



电话 2 挂断：



五、实验心得

经过本次实验，增强了我对摘挂机检测原理的掌握，了解到了摘挂机信息产生的机制，同时在编码的过程中，认识到了一些实现摘挂机信息存储的注意点。在这次实验中，我不仅锻炼了编码的能力，同时也获得了有关摘挂机检测相关的知识，收获很多。