

# 《现代交换原理》实验报告

实验名称    \_\_\_\_\_时间表调度实验\_\_\_\_\_

班    级    \_\_\_\_\_2017211305\_\_\_\_\_

姓    名    \_\_\_\_\_于海鑫\_\_\_\_\_

指导教师    \_\_\_\_\_丁玉荣\_\_\_\_\_

# 一、 实验目的

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、 实验原理及设计

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。这个调度表如下：

时间（10ms） \ 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
0	0/1	0/1	0/1
1	0/1	0/1	0/1
.....			
.....			
.....			
18	0/1	0/1	0/1
19	0/1	0/1	0/1

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，我们会检查这个表的一行，如果该行上某一列为 1，我们就执行所对应的任务，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。

### 三、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 SchTabLen 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；
```

### 四、实验代码

```
#include "bconstant.h"

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

```
ScheduleTable[i][1] = 1;

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

## 五、 实验效果检验

当调度表初始化正确时，能够进行正常的通话；如果初始化不正确，可能会造成周期性程序的不正常调用，例如位间隔调度的延迟会造成识别位间隔的延误甚至丢失。

注：由于为循环程序，所以调度表的初始化方案不唯一。

## 六、 实验结果

程序初始化后能正确检测摘挂机动作并进行通话，与预计结果相符。

## 七、 实验心得

本次实验比较简单，在理解了时间表调度原理之后，对调度表初始化程序的编程实现也非常容易，很快就可以编写完毕。通过本次实验，我对时间表调度的具体实现有了初步的理解。