

密码小键盘的驱动

中国银行陕西省分行信息科技处 胡旋/文 (西安 710001)

在银行、证券部门大量使用密码小键盘,客户毋须柜员介入就可使用密码小键盘输入自己的密码。既方便了顾客,也提高了系统的安全性。本文以实达终端为例介绍密码小键盘的驱动。

实达终端的辅口2接口信号为TTL,专门用于配接密码小键盘。主机通过以下命令选择辅口2处于特定工作状态:

ESC[P1;P2;P3;P4Z

其中:

P1 = 0 波特率为9600 P1 = 1 波特率为4800

P1 = 2 波特率为2400 P1 = 3 波特率为1200

P2 = 0 为无校验 P2 = 1 为奇校验 P2 = 2 为偶校

验

P3 = 0 为8位数据位 P3 = 1 为7位数据位

P4 = 0 为1位停止位 P4 = 1 为2位停止位

例如:

ESC[3;0;0;0Z

它的含义是设定辅口2的波特率为1200,8位数据位,无校验,1位停止位。辅助口仿真实达命令代码集如下:

ESC[/50h 允许辅口操作

ESC[/50l 禁止辅口操作

ESC[/53h 辅口内容直接送主机

ESC[/53l 辅口内容前加80H,后加81H送主机

ESC[2h 键盘锁定

ESC[2l 键盘正常

建议采用小键盘内容送到主机时锁定终端键盘。

ESC[/54h 辅口送出的字符暂存在终端内,只有当接收到回车键(0dH)后才一起送到主机。当接收到更正键(2eH)时,取消以前的内容,重新开始输入。

ESC[/54l 终端收到一个字符即向主机送一个字符。

ESC[/51h 执行该命令后,主机发送到终端的数据将全部送给辅口,主要用于控制小键盘的指示灯。

ESC[/51l 主机结束向辅口送数据

例如:ESC[/51h\x81ESC[/51l 密码键盘红灯亮绿灯灭

ESC[/51h\x82ESC[/51l 密码键盘绿灯亮红灯灭

ESC[/51h\x81ESC[/51l 二灯全亮

使用以上命令集时,必须进入终端 setup 程序将仿真命令设为仿真达。

在小键盘输入过程中,ENTER 为确认键,按更正键取消输入的字符,其余各键均为发送相应 ASCII 码。

以下密码键盘驱动程序在 STARNT-560 终端上调试通过,读者可将其移植到其他型号的终端上。

```
#include<stdio. h>
```

```
#include<termio. h>
```

```
#include<fcntl. h>
```

```
#include<time. h>
```

```
#define ERR 1
```

```
#define OK 2
```

```
#define RED "\x81" /* 红灯亮 */
```

```
#define GREEN "\x82" /* 绿灯亮 */
```

```
#define DARK "\x83" /* 二灯灭 */
```

```
#define QUITKEY 0x7f
```

```
/* 终端辅助口命令 */
```

```
#define SELAUX2 "\x1b[3;0;0;0Z"
```

```
/* 选择辅口2 */
```

```
#define INITAUX "\x1b[ /50h\x1b[ /54h\x1b[53h"
```

```
/* 辅口允许回车才送主机字符 辅口内容无前导符 */
```

```
#define CLOSEAUX "\x1b[ /50l\x1b[2l"
```

```
/* 辅口禁止键盘正常工作 */
```

```
#define AUXLEAD "\x1b[ /51h"
```

```
/* 主机往辅助口送的内容以51h打头 */
```

```
#define AUXEND "\x1b[ /51l"
```

```
/* 主机往辅助口送的内容以51l结束 */
```

```
#define KEYLOCK "\x1b[2h"
```

```
/* 键盘锁定 */
```

```
static struct termio oterm, term;
```

```
static int term_fd;
```

```
main()
```

```
{
```

```
int i;
```

```
char ch, passwd_buf[20];
```

```
setterm(0); /* 设置终端状态 */
```

```
selaux2(); /* 初始化端口2 */
```

```
sendaux(GREEN); /* 密码键盘绿灯亮 */
```

```
i = 0;
```

```
while(1){
```

```
read(term_fd, &ch, 1);
```

```
passwd_buf[i++] = ch;
```

```
if(ch == 0x0a) /* 确认键退出 */
```

```
break;
```

```
}
```

```
printf("The input password is %s\n", passwd_buf);
```

```
sendaux(DARK); /* 关密码键盘灯 */
```

```
closeaux(); /* 关闭辅口 */
```

```
resetterm(); /* 恢复终端状态 */
```

```
return;
```

```
}
```



/* 设置终端状态

参数: fstat

返回: -1, 出错

*/

setterm(int fstat)

{

term_fd = open((char *)ttyname(1), O_RDWR | fstat);

if(term_fd < 0) return -1;

if(ioctl(term_fd, TCGETA, &term) < 0) return ERR;

oterm = term;

term.c_iflag &= ~IXON;

term.c_lflag &= ~ECHO;

term.c_lflag &= ~ICANON;

term.c_cc[VMIN] = 1;

term.c_cc[VTIME] = 1;

term.c_cc[VQUIT] = QUITKEY;

if(ioctl(term_fd, TCSETA, &term) < 0) return ERR;

return term_fd;

}

/* 恢复终端状态

参数: term_fd: 终端文件句柄

返回: ERR: 出错

OK: 正确

*/

resetterm()

{

if(term_fd < 0) return -1;

if(ioctl(term_fd, TCSETA, &oterm) < 0) return ERR;

close(term_fd);

return OK;

}

/* 设置终端辅助口2 */

selaux2()

{

write(term_fd, SELAUX2, strlen(SELAUX2));

write(term_fd, INITAUX, strlen(INITAUX));

write(term_fd, KEYLOCK, strlen(KEYLOCK));

return OK;

}

/* 向辅口发送字符串 */

sendaux(s)

char * s;

{

write(term_fd, AUXLEAD, strlen(AUXLEAD));

write(term_fd, s, strlen(s));

write(term_fd, AUXEND, strlen(AUXEND));

}

/* 关闭辅口 */

closeaux()

{

write(term_fd, CLOSEAUX, strlen(CLOSEAUX));

} □

