哈尔滨工业大学计算机科学与技术学院
2016年秋季学期《操作系统》

Lab6:字符显示的控制

姓名	学号	联系方式
匡盟盟	1143220116	kuangmeng@msn.com
樊晨霄	15S008199	18513534698

目　录

一、实验目的

- 加深对操作系统设备管理基本原理的认识，实践键盘中断、扫描码等概念；
- 通过实践掌握Linux 0.11对键盘终端和显示器终端的处理过程。

二、实验内容

实验基本内容

修改Linux 0.11的终端设备处理代码，对键盘输入和字符显示进行非常规的控制。

在初始状态，一切如常。用户按一次F12后，把应用程序向终端输出所有字母都替换为“*”。用户再按一次F12，又恢复正常。第三次按F12，再进行输出替换。依此类推。

以ls命令为例：

正常情况：

ls

hello.c hello.o hello

第一次按F12，然后输入ls：

**

*****.* *****.* *****

第二次按F12，然后输入ls：

ls

hello.c hello.o hello

第三次按F12，然后输入ls：

**

*****.* *****.* *****

三、实验过程

1. 将linux-0.11/kernel/chr_drv/目录下的keyboard.S文件注释一行（因为我们不需要显示调用各任务的状态）：

```
func:
        pushl %eax
        pushl %ecx
        pushl %edx
/*      call show_stat   */
        popl %edx
        popl %ecx
        popl %eax
        subb $0x3B,%al
        jb end_func
        cmpb $9,%al
        jbe ok_func
        subb $18,%al
        cmpb $10,%al
        jb end_func
        cmpb $11,%al
        ja end_func
```

2. linux-0.11/kernel/chr_drv/目录下的tty_io.c文件，添加如下代码到copy_to_cooked函数（增加对F12按键的判断，接收的是扫描码）：

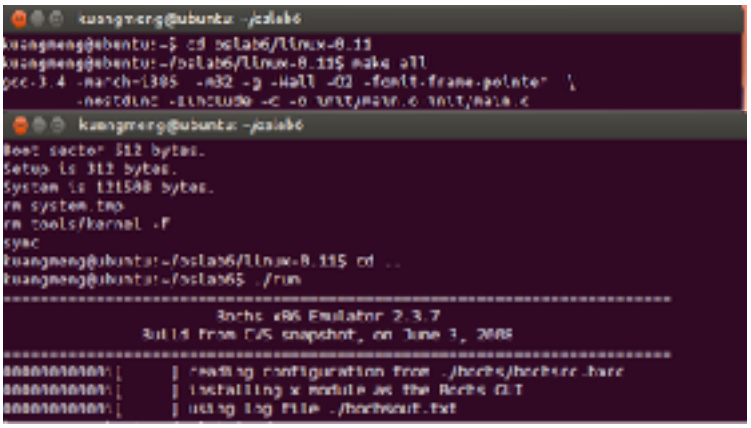
```
while (!EMPTY(tty->read_q) && !FULL(tty->secondary)) {
        GETCH(tty->read_q,c);
        if (c=='L'){
                if (judge) judge=0;
                else judge=1;
                break;
        }
```

3. 修改linux-0.11/kernel/chr_drv/目录下的console.c文件，添加如下代码到con_write函数（改变显示字符）：

4. 在linux-0.11/include/asm/目录下的system.h文件开始声明一个变量：

5. 使用“./run”运行linux-0.11，测试功能，得到如下截图：

使用的命令为：



至此，本实验结束！

四、回答问题

1. **在原始代码中，按下F12，中断响应后，中断服务程序会调用func？它实现的是什么功能？**

答：将F12转义成转义字符序列 [[L并进行之后的判断 , 对F1-F12处理类似 [[A -> [[L。（func函数的功能就是把功能键扫描码转换成转义字符并存放到读队列中，进而判断是否是F1——F12的扫描码，若是，则将查func_table中的四个字节的转义字符序列放入缓冲队列。）

2. **在你的实现中，是否把向文件输出的字符也过滤了？如果是，那么怎么能只过滤向终端输出的字符？如果不是，那么怎么能把向文件输出的字符也一并进行过滤？**

答：实现了文件输出的过滤，该过滤是通过修改fs/file_dev.c中file_write()函数，实现代码类似tty_write()函数。具体修改如下截图：

```
while (c-->0){
        tmp = get_fs_byte(buf++);
        if(f12_flag == 1){
                if((tmp>='A'&&tmp<='Z')||(tmp>='a'&&tmp<='z')||(tmp>='0'&&tmp<='9'))
                        tmp = '*';
        }
        *(p++) = tmp;
}
```

如果只过滤终端输出字符，则可以去掉file_write()修改即可；

实验心得

字符的显示是我们使用计算机时最频繁面对的问题，也是计算机与用户交互的最基本、最频繁的方式。当然，我们在日常使用计算机时并不会用到像本次实验这样的字符控制功能，但是通过自己动手运行了这次实验，我们能明白Linux是怎样实现从IO设备（键盘）中读取扫描码，并将其打印到屏幕上的过程，理解了这一过程，我们就可以很容易理解我们日常使用的打字程序的工作原理了，也可以理解屏幕显示的内在逻辑过程，为我们今后理解程序运行，起到了很好地铺垫作用。