## 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。

## 实验内容

本实验的基本内容是修改 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。

## 报告

|  |
| --- |
| * 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？ * 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？ |
| 最多三个参数，也就是通过寄存器ebx，ecx，edx，可以通过寄存器传递参数位置，也就是寄存器间接寻址来传递多个参数 |
| 修改 include/linux/sys.h  在sys\_call\_table最后加入sys\_foo，  再加上extern rettype sys\_foo();  修改include/unistd.h  添加 #define \_\_NR\_foo 系统调用号  修改 kernel/system\_call.s  nr\_system\_calls =系统调用总数目加1  在kernel中添加 foo.c  最后修改kernel的Makefile，以链接foo.c  调用处添加有  #define \_\_LIBRARY\_\_  #include <unistd.h>  \_syscallN宏展开系统调用 |

## 实验步骤

1. 修改 bootsect.s 更改字符串及其长度，运行并显示

|  |
| --- |
| 修改文件 |
|  |
| 运行效果 |
|  |

1. 修改bootsect.s和setup.s显示Now we are in SETUP

|  |
| --- |
| bootsect.s文件 |
|  |
| setup.s文件 |
|  |
| 运行效果 |
|  |

1. 修改 build.c 方便使用

|  |
| --- |
|  |

1. 读取内存大小、光标位置并显示

|  |
| --- |
| 关键代码 |
|  |
| 运行效果 |
|  |