

**实验报告**



**题目： 键盘驱动程序的分析与修改**

**班 级：**

**学 号：**

**姓 名：**

**学 院：**

**2022年 11 月 日**

一、实验目的  
1.理解I/O系统调用函数和C标准I/O函数的概念和区别；

2.建立内核空间I/O软件层次结构概念，即与设备无关的操作系统软件、设备驱动程序和中断服务程序；

3.了解Linux-0.11字符设备驱动程序及功能，初步理解控制台终端程序的工作原理；

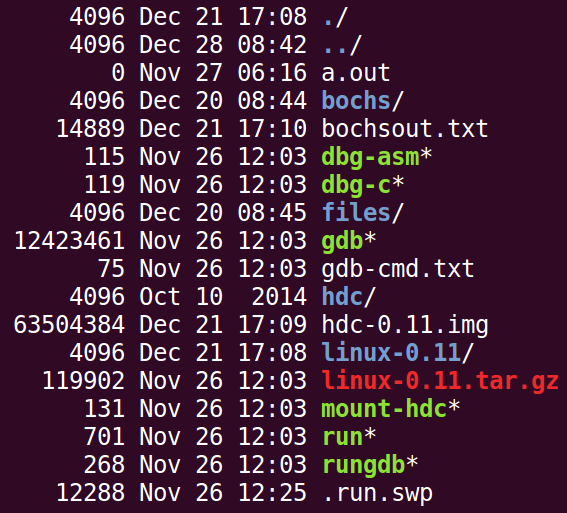
4.通过阅读源代码，进一步提高C语言和汇编程序的编程技巧以及源代码分析能力；

5.锻炼和提高对复杂工程问题进行分析的能力，并根据需求进行设计和实现的能力。

1. 实验环境
2. 硬件：学生个人电脑（x86-64）
3. 软件：Windows 10，VMware Workstation 15 Player，32位Linux-Ubuntu 16.04.1
4. gcc-3.4编译环境
5. GDB调试工具

三、实验内容

从网盘下载lab4.tar.gz文件，解压后进入lba4目录得到如下文件和目录：



实验常用执行命令如下：

* 执行./run ，可启动bochs模拟器，进而加载执行Linux-0.11目录下的Image文件启动linux-0.11操作系统
* 进入lab4/linux-0.11目录，执行make编译生成Image文件，每次重新编译（make）前需先执行make clean
* 如果对linux-0.11目录下的某些源文件进行了修改，执行./run init 可把修改文件回复初始状态

本实验包含2关，要求如下：

* Phase 1

键入F12，激活\*功能，键入学生本人姓名拼音，首尾字母等显示\*

比如：zhangsan，显示为：\*ha\*gsa\*

* Phase 2

键入“学生本人学号” ：激活\*功能，键入学生本人姓名拼音，首尾字母等显示\*

比如：zhangsan，显示为：\*ha\*gsa\*，

再次键入“学生本人学号-” ：取消显示\*功能

提示：完成本实验需要对lab4/linux-0.11/kernel/chr\_drv/目录下的keyboard.s、console.c和tty\_io.c源文件进行分析，理解按下按键到回显到显示频上程序的执行过程，然后对涉及到的数据结构进行分析，完成对前两个源程序的修改。修改方案有两种：

* 在C语言源程序层面进行修改
* 在汇编语言源程序层面进行修改

实验4的其他说明见lab4.pdf课件和爱课堂中虚拟机环境搭建相关内容。linux内核完全注释(高清版).pdf一书中对源代码有详细的说明和注释。

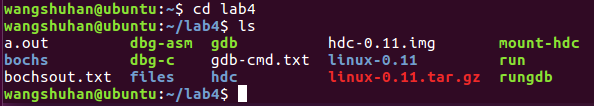
四、源代码的分析及修改

0、准备阶段

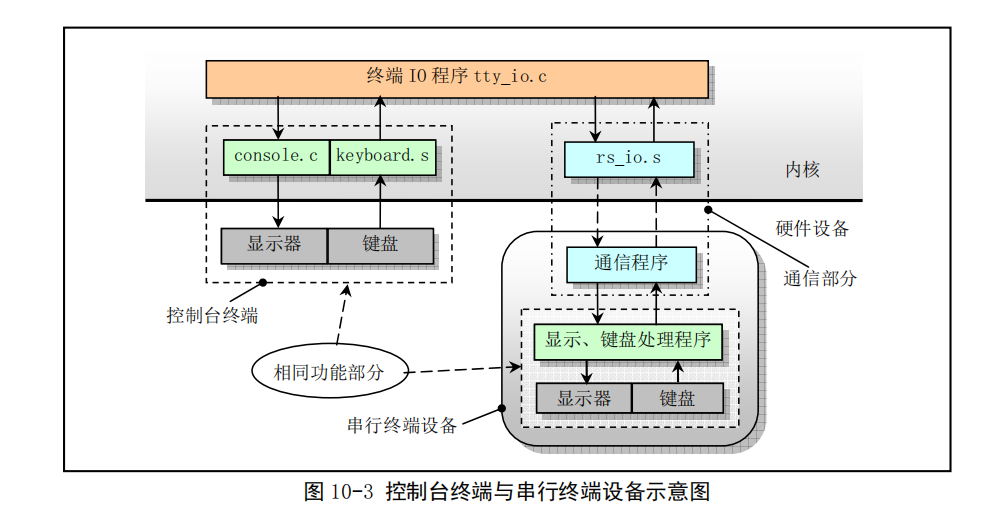
首先按照实验指导手册的指导配置好实验环境如图所示



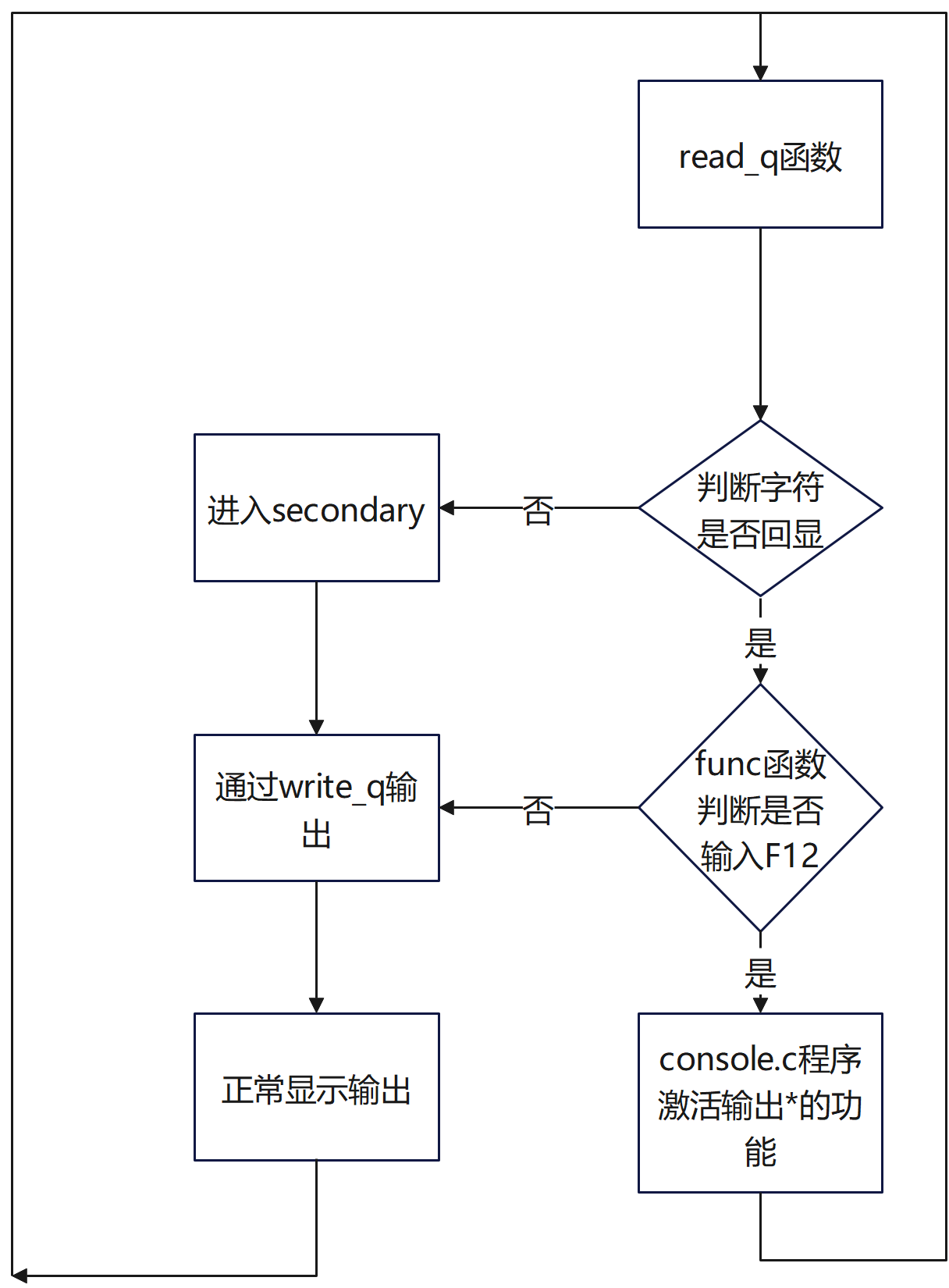
使用tar -svf lab4.tar指令解压lab4文件夹后使用cd lab4指令和ls指令重定向至lab4文件夹并打印文件中的内容



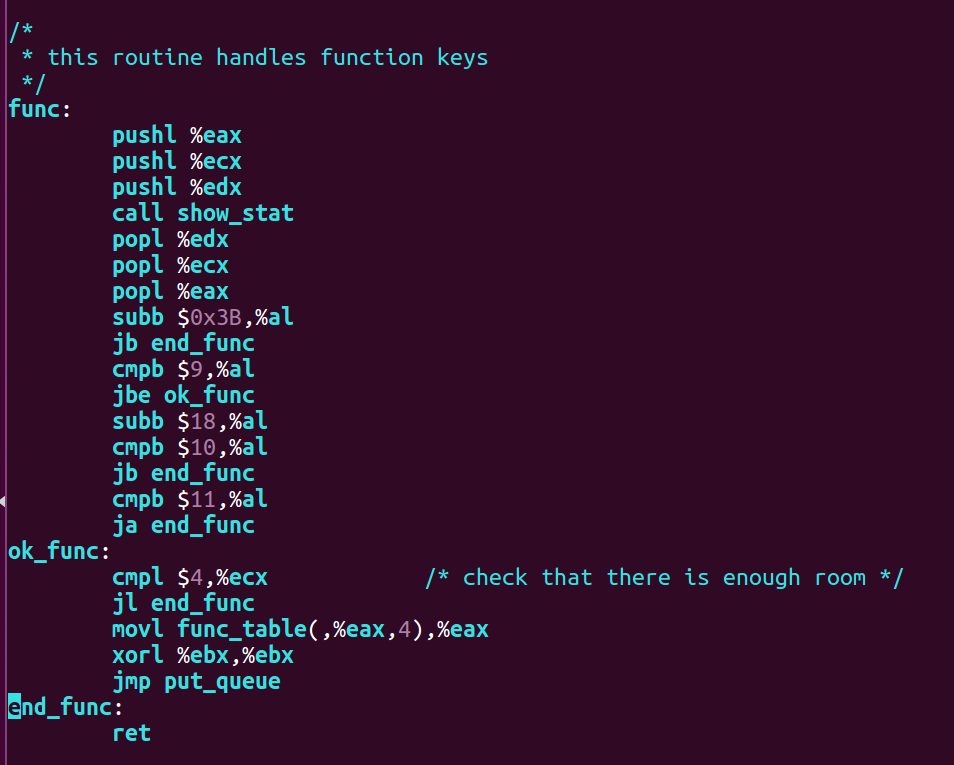
1、第一阶段

首先查阅linux内核完全资料，在第十章字符设备驱动程序中我们可以知道，在Linux-0.11系统中可以使用两类终端，一类是主机上的控制台终端，另一类是串行硬件终端设备。控制台终端由内核中的键盘输入终端处理程序keyboard.s和显示控制程序console.c进行管理。console.c接收上层tty\_io.c程序传递下来的显示字符或控制信息，并控制在显示器上显示的内容，同时主机把键盘按键触发产生的代码经由keyboard.s传送到tty\_io.c程序去处理。

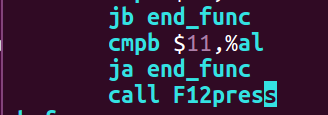
于是要实现输入F12隐藏字符的功能的思维导图为：

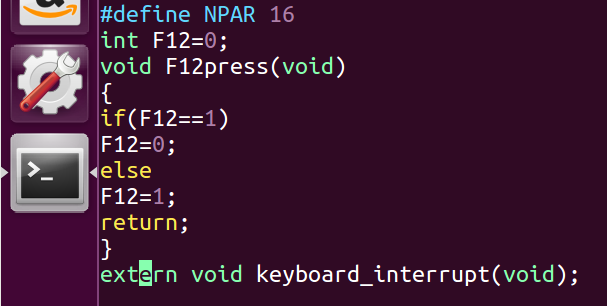


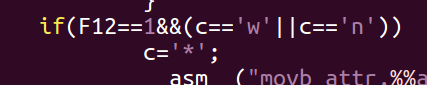
使用cd lab4/linux-0.11/kernel/chr\_drv/指令重定向至目标文件夹，使用vim keyboard.S 指令查看文件内容。根据参考书中的内容可知func函数处理输入的字符，及图中所示的部分



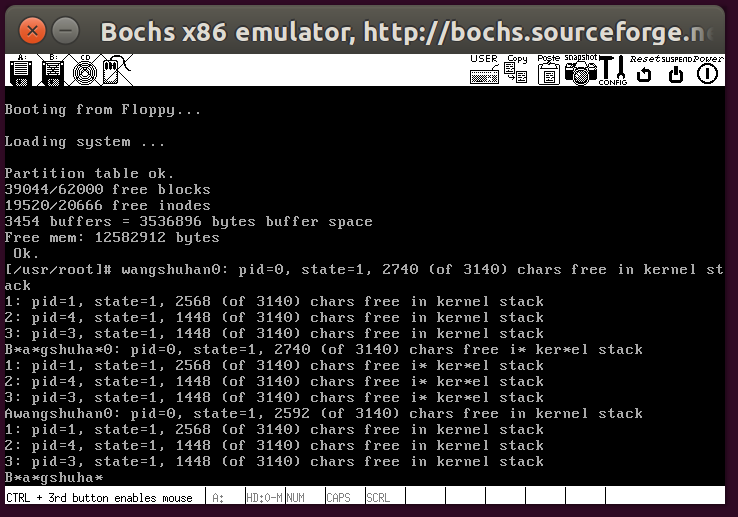
图中第15行检验输入字符是否为F12，如果不是不做处理，为了实现输入F12实现对应功能，考虑在第17行调用并构造函数F12press，使得当输入 字符为F12时调用该函数。



为了实现功能，我们还需更改汇编语言对应的C语言代码。函数F12press应存放在console.c文件中。使用vim console.c指令打开对应文件。之后写入以下代码片段：

之后要实现当F12为1时，将字符隐藏的功能。找到con\_write函数，本人的名字拼音为wangshuhan，故设置当F12为1时将输入的w和n设为\*。写入以下代码片段：

保存以上更改，将目录重定向回linux-0.11，执行make clean指令和make重新编译，最后执行./run启动模拟器。运行结果如下：

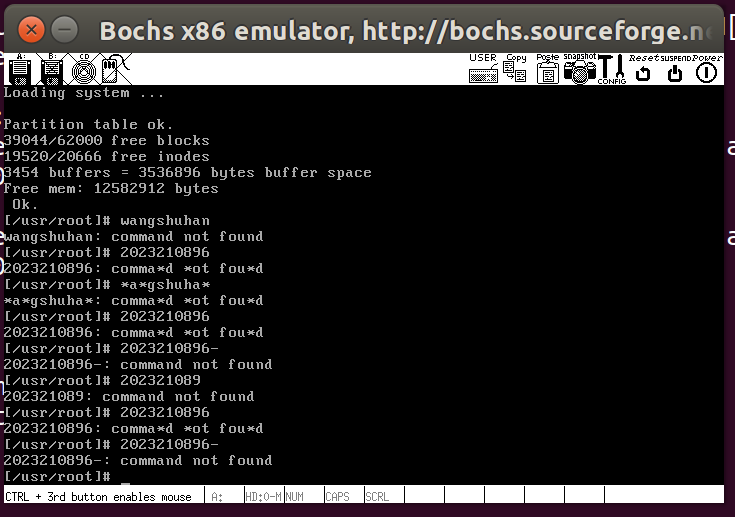


2、第二阶段

第二阶段与第一阶段类似，仅需将输入F12的标记变成连续输入“2023210896”字符串的标记。因为这个字符串的输出是连续的，故设置一个全局变量count来计算键盘输入到了第几位。为实现该功能，考虑以下代码



故在保留第一阶段的代码的前提下保存以上更改，将目录重定向回linux-0.11，执行make clean指令和make重新编译，最后执行./run启动模拟器。运行结果如下：



五、总结体会

这次的实验相比之前的实验更注重对linux系统输入输出的理解，同时实验内容页少于先前的实验，但这个实验的一大考验反而是在准备阶段配置环境的部分。一开始在下载按照VMware启动虚拟机后卡顿十分严重，在按照网络上的教程关闭Windows系统的内核隔离的功能后才没有那么卡顿。第二个问题出现在在下载了gcc编译器后执行sudo指令后要么显示失败，要么进度条保持不动，我在从头开始一行一行执行指令后才解决了这个问题。还有其他各种各样的小问题如虚拟机未配置vim文本编辑器需要自己配置，在修改完代码后使用./run指令生成模拟器失败等等，这些问题并非是知识上问题，但都是我在操作上的疏忽或是不完善，。所以我从这次实验中不仅学到了linux系统的操作内核，也让我积累了发现问题、解决问题的能力。这也是这门实践课的最后一次实验，这门课的四门实验都让我感到十分有趣，尤其是中间两个实验看到自己的名字不断向上攀升，同时利用所学知识解开一个又一个难题或是炸弹都让我喜爱上了这门课程。感谢周锋老师和助教杨同学和王同学的付出。