

**实验报告**



**题目： 键盘驱动程序的分析与修改**

**班 级： 2020211306**

**学 号： 2020211376**

**姓 名： 马天成**

**学 院： 计算机学院**

**2021年 12 月 10 日**

目录

[一、实验目的 2](#_Toc90114324)

[二、实验环境 2](#_Toc90114325)

[三、实验内容 2](#_Toc90114326)

[四、实验步骤及实验分析 3](#_Toc90114327)

[准备工作 3](#_Toc90114328)

[phase1 5](#_Toc90114329)

[初步理解 5](#_Toc90114330)

[修改代码 5](#_Toc90114331)

[运行结果 6](#_Toc90114332)

[phase2 7](#_Toc90114333)

[初步理解 7](#_Toc90114334)

[修改代码 7](#_Toc90114335)

[运行结果 9](#_Toc90114336)

[五、总结体会 10](#_Toc90114337)

[六、诚信声明 10](#_Toc90114338)

**版本修正：**

**有限状态自动机的state转移修改了一处：见P8，P9**

# 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。**

**报告邮寄（最迟时间：2021年12月22日晚23：59）：**

**大二班（5-8班）：**yangyyj98@bupt.edu.cn

# 二、实验环境

**1、硬件：学生个人电脑（x86-64）**

**2、软件：Windows 10，VMware Workstation 15 Player，32 位 Linux-Ubuntu 16.04.1**

**3、gcc-3.4 编译环境**

**4、GDB 调试工具**

# 三、实验内容

**解压lab4.tar.gz 文件，解压后进入 lab4 目录得到如下文件和目录：**

**\*\*\*\***

**安装gcc编译器：**

**\*\*\*\***

**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，键入学生本人的姓名拼音，首尾字母等显示\***

**比如：zhangsan，显示为：\*ha\*gsa\***

** Phase 2**

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

**比如：zhangsan，显示为：\*ha\*gsa\*，**

**键入“学生本人的学号-” ：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**

**其他说明见 实验四.ppt 。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。**

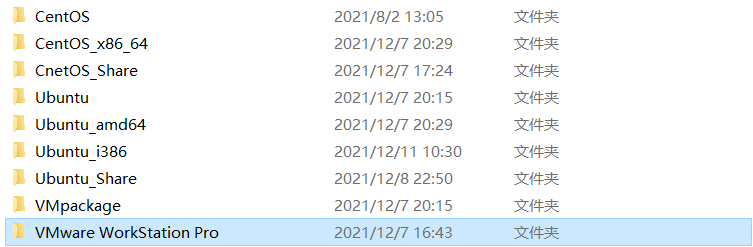
# 四、实验步骤及实验分析

## 准备工作

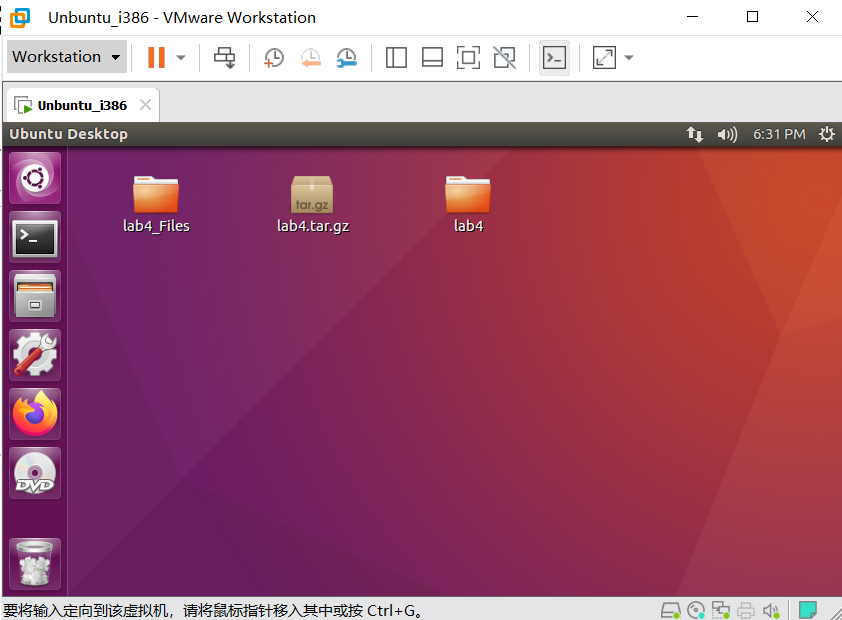
**①. 安装虚拟机**



选择第一个**32位**光盘映像文件

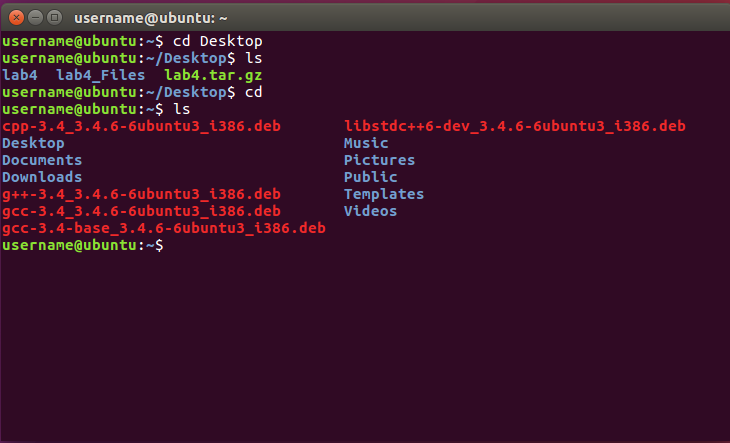


我的**Visual Machine Folder**



**Ubuntu\_i386**

**②. 安装C（C++）环境**



**C(C++)**语言运行环境的下载和安装

此外，还有**两个东西需要自行解决**：

* **Vim版本不够导致上下左右键输出ABCD**
* **不能进行make指令**

这个需要在网上搜指令解决。

## phase1

### 初步理解

第一题主要是跟着PPT走思路，将io的思路了解一下。

**功能是：键入 F12，激活\*功能，键入学生本人的姓名拼音，首尾字母等显示\***

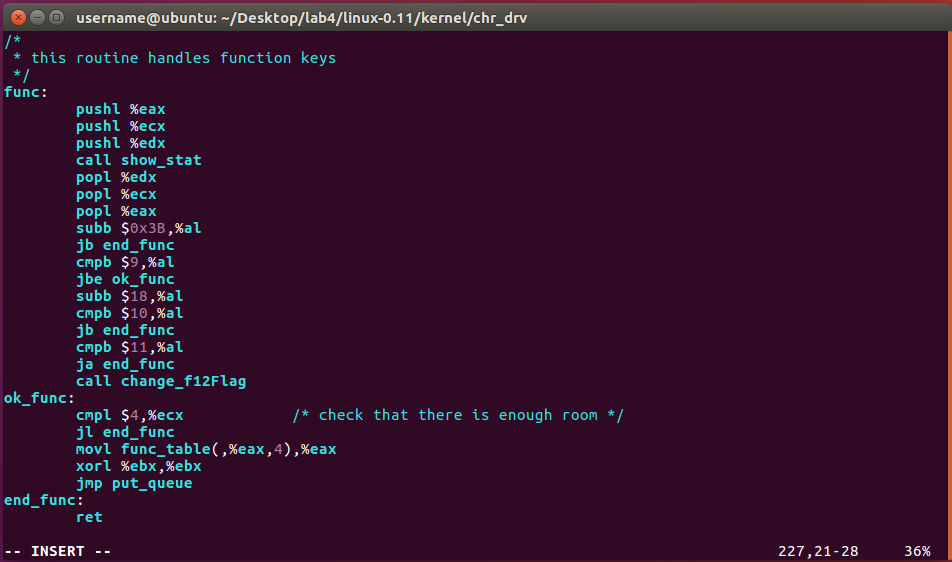
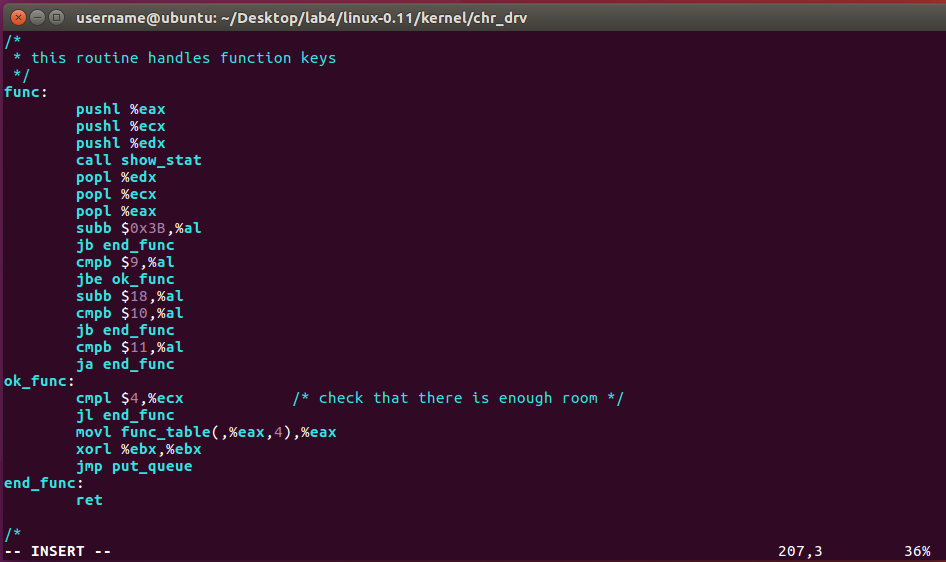
* **MaTianCheng -> \*aTianChen\***
* **matiancheng -> \*atianchen\***
* **MATIANCHENG -> \*ATIANCHEN\***

主要是进行两个文件的修改：

* **keyboard.S**
* **console.c**

### 修改代码

①. 修改**keyboard.S**

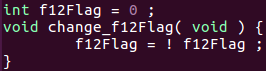
 

原来的**keyboard.S** 现在的**keyboard.S**

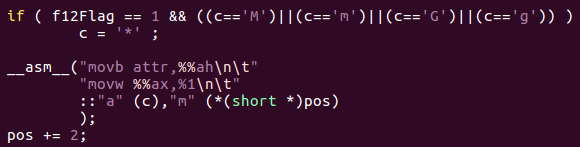
并且需要加上全局变量



②. 修改**console.c**

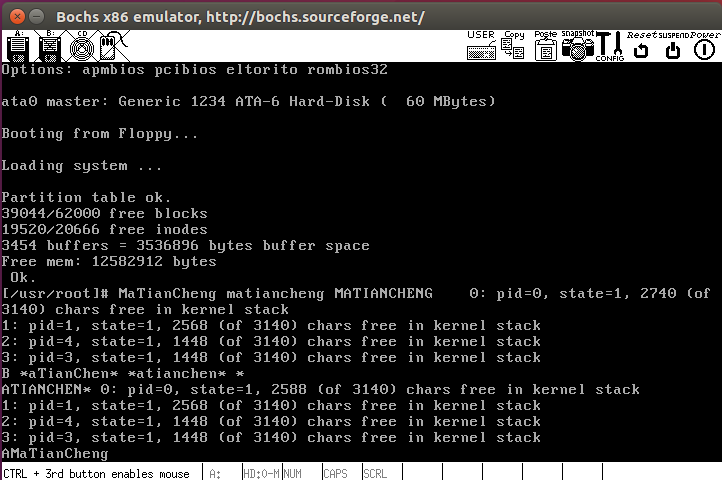


**void change\_f12Flag( void )**



**修改条件**

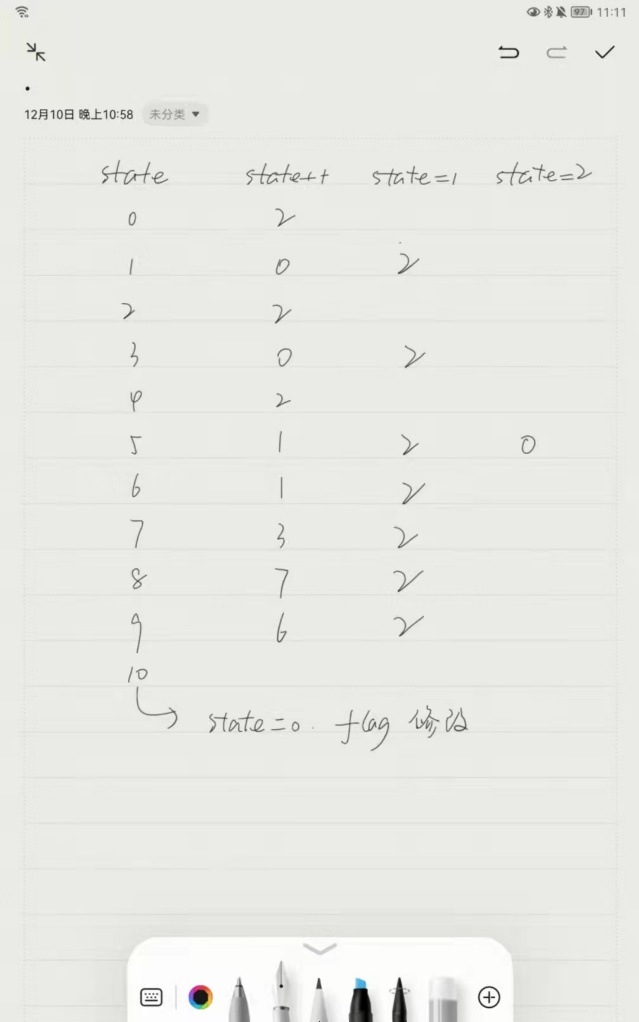
### 运行结果



## phase2

### 初步理解

显然，这是一个**有限状态自动机**的问题。那么只需要把转移条件写出来就行。



**state : 0 -> 10**

表示输入后已经正确了几个数字。

其中，**状态5**极为特殊：

在数字为**0**时是返回**state = 4**

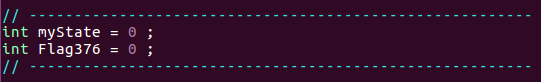
**Flag : 0 1**

1表示屏蔽，0表示正常。

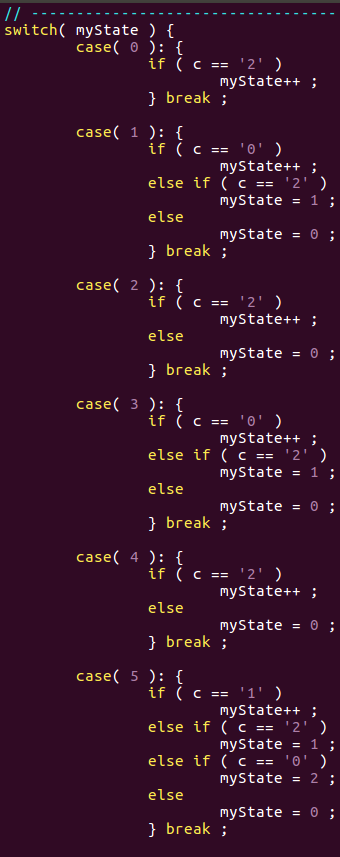
### 修改代码

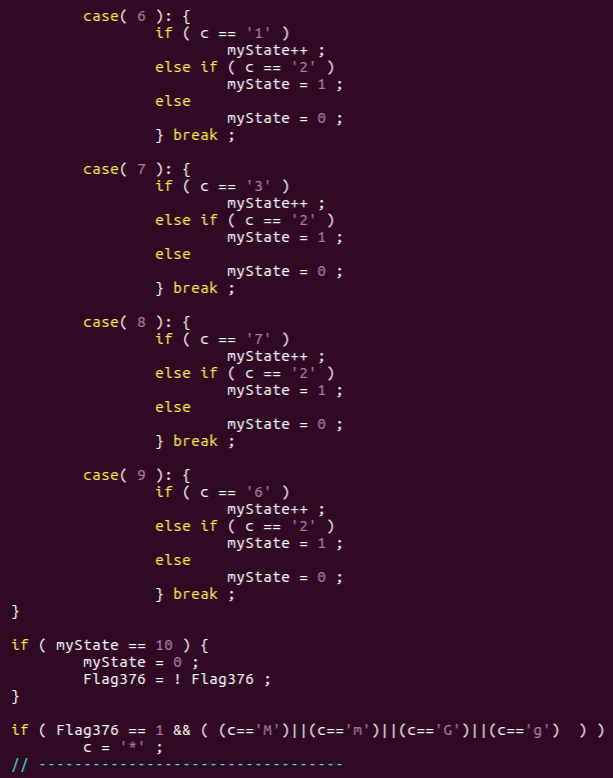
① ./run init 清空第一题

②. 修改**console.c**



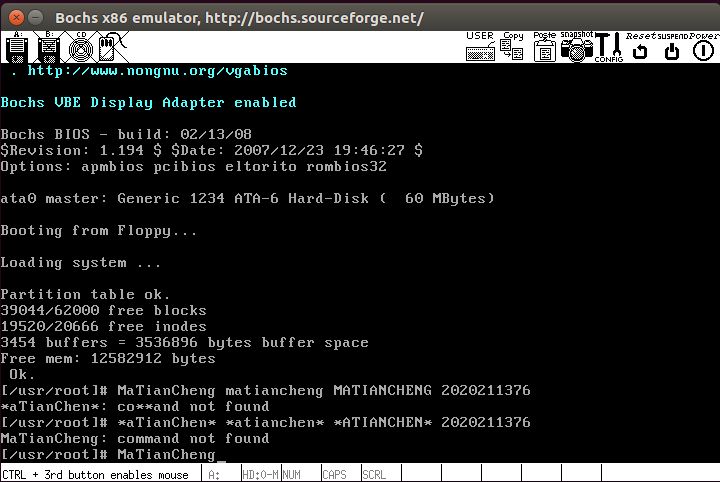
变量



code\_1 

code\_2

### 运行结果



# 五、总结体会

本次实验较之前比较简单，难点主要在ubuntu环境下软件的配置问题以及理解各个层面的相互调用上，在动手完成ppt上的实例后已经对整个层面有了初步的了解，再加上阅读一定的文献，对linux的io内核有了更深层的理解，所以实验做起来也算得心应手。因为之前大一上oj上练过好几次有限状态自动机的题，所以看到第二阶段的意思也没有很慌张，而是很快画出了状态图并转换成了c代码，完成了阶段2。

在下载安装as86的过程中，出现了很多问题，我也在不断寻找解决问题的途径和手段，在这个过程中对于sudo等终端内的指令更加熟练的操作，并通过在ubuntu系统下更加明显的感受到linux系统相比于windows在开源程度等方面的差异和区别，在linux环境下，有效帮助我更加了解了计算机io等深层计算机功能的认识。

本次实验很令人愉悦，增强了对linux系统的操作熟练度，让我真正体会到了命令行控制电脑的魅力。

# 六、诚信声明

需要填写如下声明，并在底部给出手写签名的电子版。

此外，我还参考了以下资料：

linux指令：

* 重装新版本 Vim sudo apt-get install vim
* 重装 yum -y install make

在我提交的程序中，还在对应的位置以注释形式记录了具体的参考内容。

我独立完成了本次实验除以上方面之外的所有工作，包括分析、设计、编码、调试与测试。

我清楚地知道，从以上方面获得的信息在一定程度上降低了实验的难度，可能影响起评分。

我从未使用他人代码，不管是原封不动地复制，还是经过某些等价转换。

我未曾也不会向同一课程（包括此后各届）的同学复制或公开我这份程序的代码，我有义务妥善保管好它们。

我编写这个程序无意于破坏或妨碍任何计算机系统的正常运行。

我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按照0分计。

（签名）