

本科生实验报告

实验课程:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_操作系统\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验名称:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_编译内核\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

专业名称:\_\_\_\_\_\_\_\_\_\_\_计算机科学与技术\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生姓名:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_凌国明\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生学号:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21307077\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验地点:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_教室\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验成绩:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

报告时间:\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023.03.22\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **实验要求**

（1）学习x86汇编、计算机的启动过程、IA-32处理器架构和字符显存原理。

（2）根据所学的知识，能自己编写程序，然后让计算机在启动后加载运行，以此增进对计算机启动过程的理解，为后面编写操作系统加载程序奠定基础。

（3）将学习如何使用gdb来调试程序的基本方法。

**实验任务**

（1）（i）**复现**example 1。说说你是怎么做的，并将结果截图。

（ii）请修改example 1的代码，使得MBR被加载到0x7C00后在(12,12)(12,12)处开始**输出你的学号**。注意，你的学号显示的前景色和背景色必须和教程中不同。说说你是怎么做的，并将结果截图。

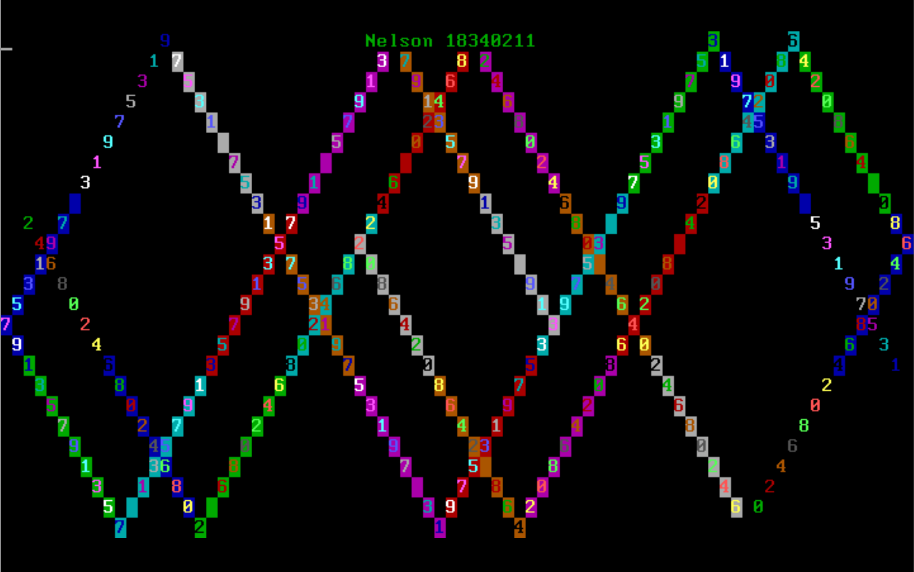
（2）（i）请探索实模式下的光标中断，**利用中断实现光标的位置获取和光标的移动**说说你是怎么做的，并将结果截图。

（ii）请修改1.2的代码，**使用实模式下的中断来输出你的学号**。说说你是怎么做的，并将结果截图

（iii）在2.1和2.2的知识的基础上，探索实模式的键盘中断，**利用键盘中断实现键盘输入并回显**

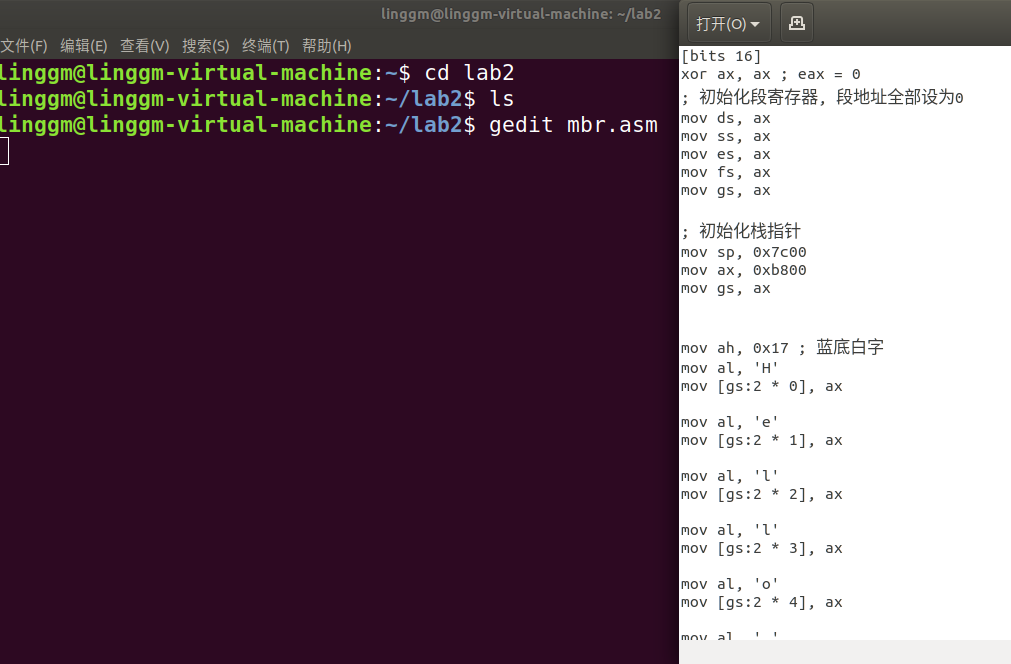
（3）用x86汇编语言实现**分支逻辑，循环逻辑和函数调用**，并通过测试

（4）用x86汇编语言编写**字符弹射小程序**。其从点(2,0)(2,0)处开始向右下角45度开始射出，遇到边界反弹，反弹后按45度角射出，方向视反弹位置而定。同时，你可以加入一些其他效果，如变色，双向射出等。注意，你的程序应该不超过510字节，否则无法放入MBR中被加载执行。



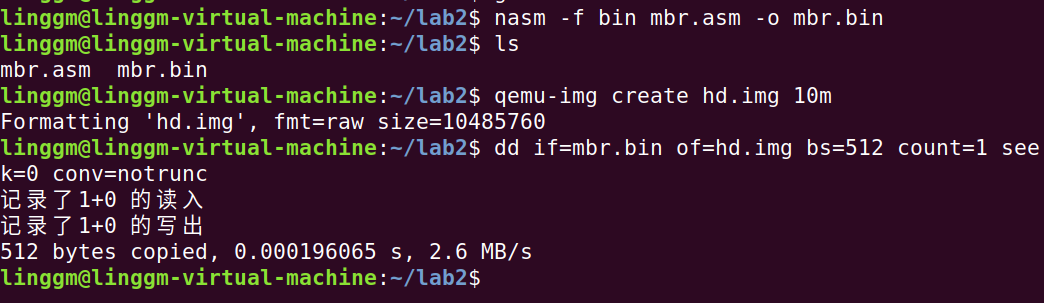
1. **实验过程**
   * 1. 复现example1：

* 第一步：编写mbr

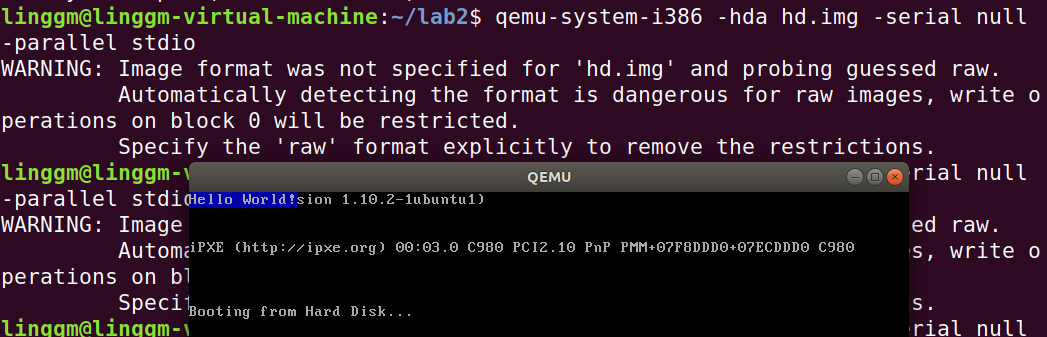
****

* 第二步：将mbr.asm汇编成bin格式；

创建磁盘；将mbr.bin写入磁盘首扇区

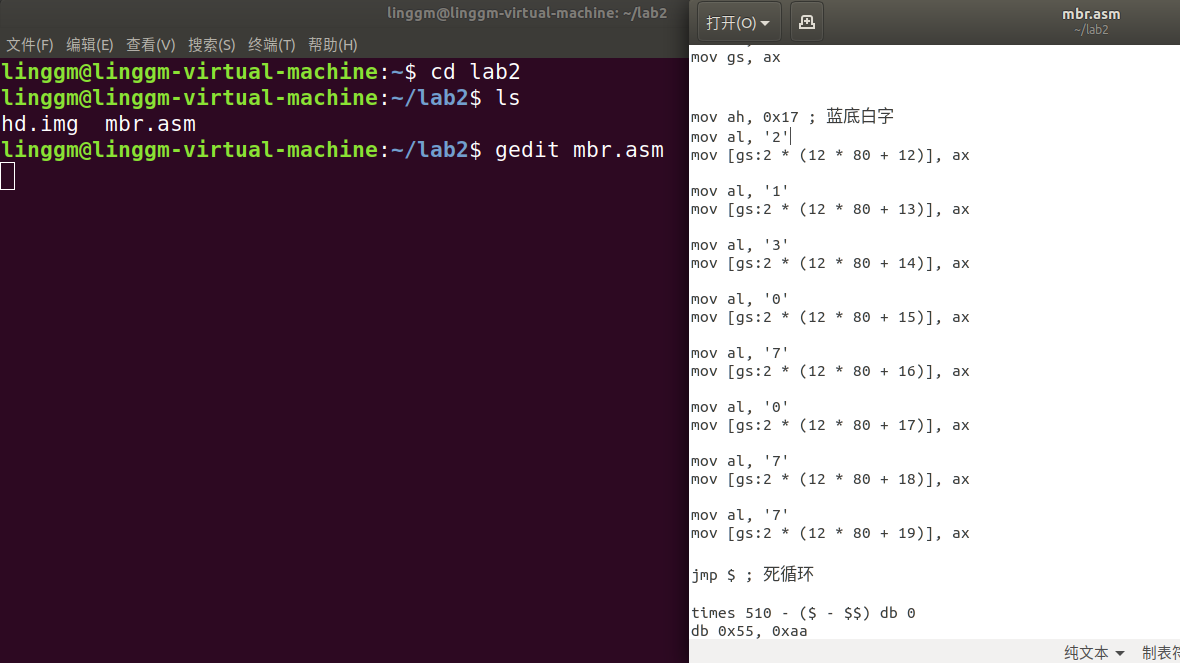


* 第三步：启动qemu进行测试

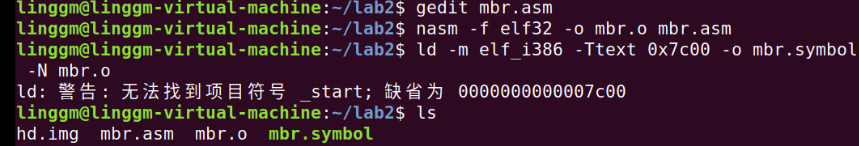


至此，example1复现完毕，启动gdb调试可参考任务1.2。

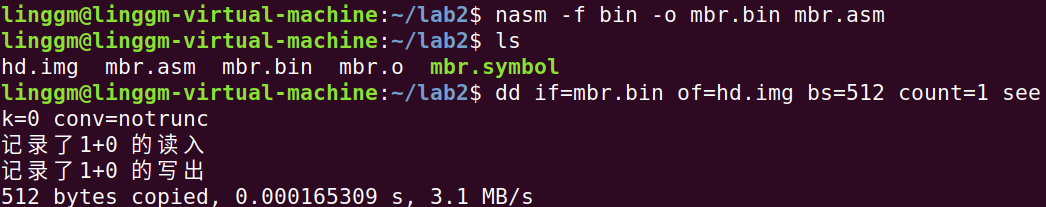
* + 1. 修改example 1的代码，使得MBR被加载到后**输出我的学号**
* 第一步：编写mbr



* 第二步：汇编为elf32，链接，生成符号表



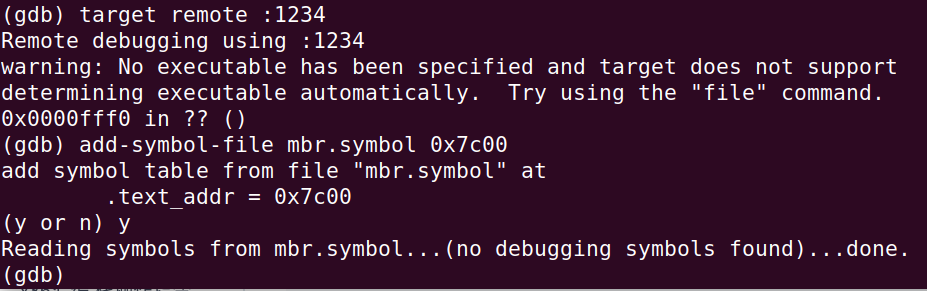
* 第三步：汇编为bin，写入磁盘首扇区



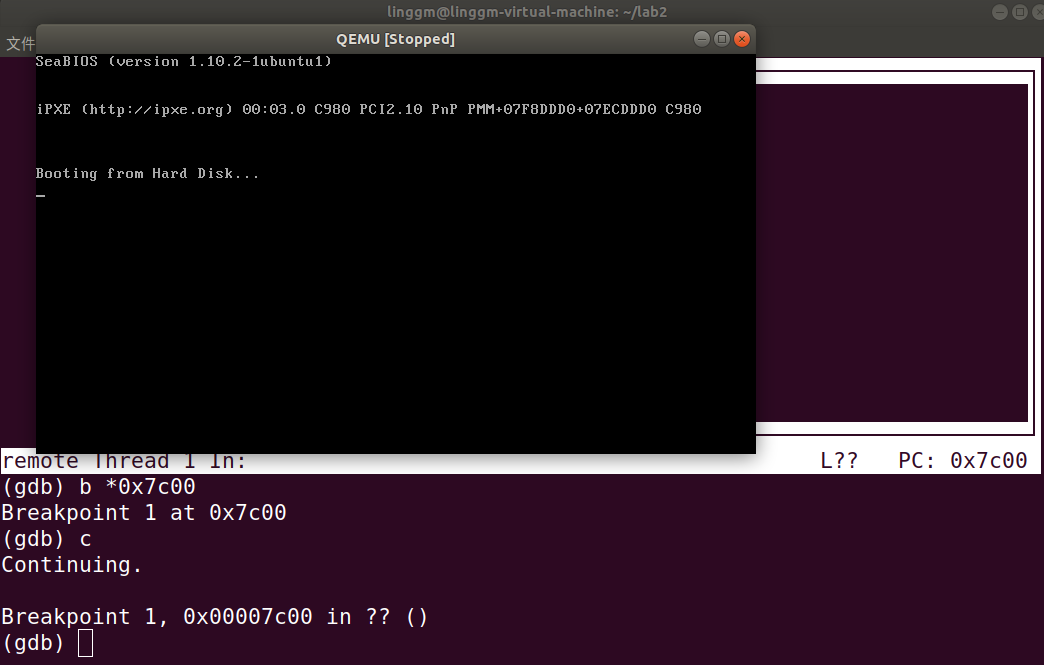
* 第四步：启动qemu进行测试



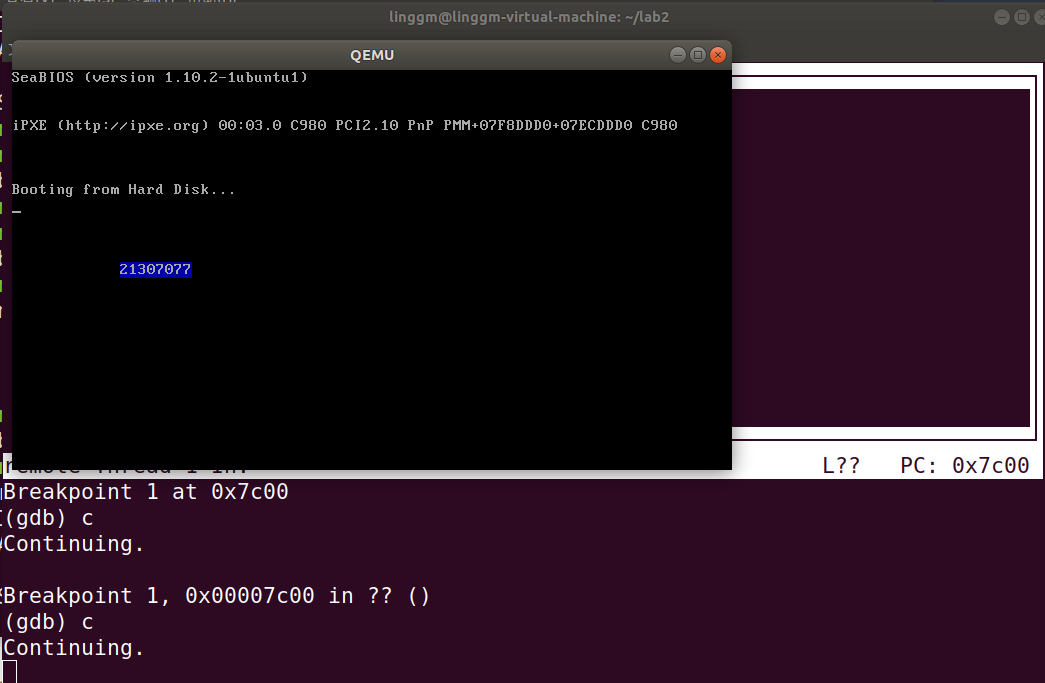
* 第五步：启动gdb，连接qemu，加载符号表



* 第六步，设置断点，c，调试



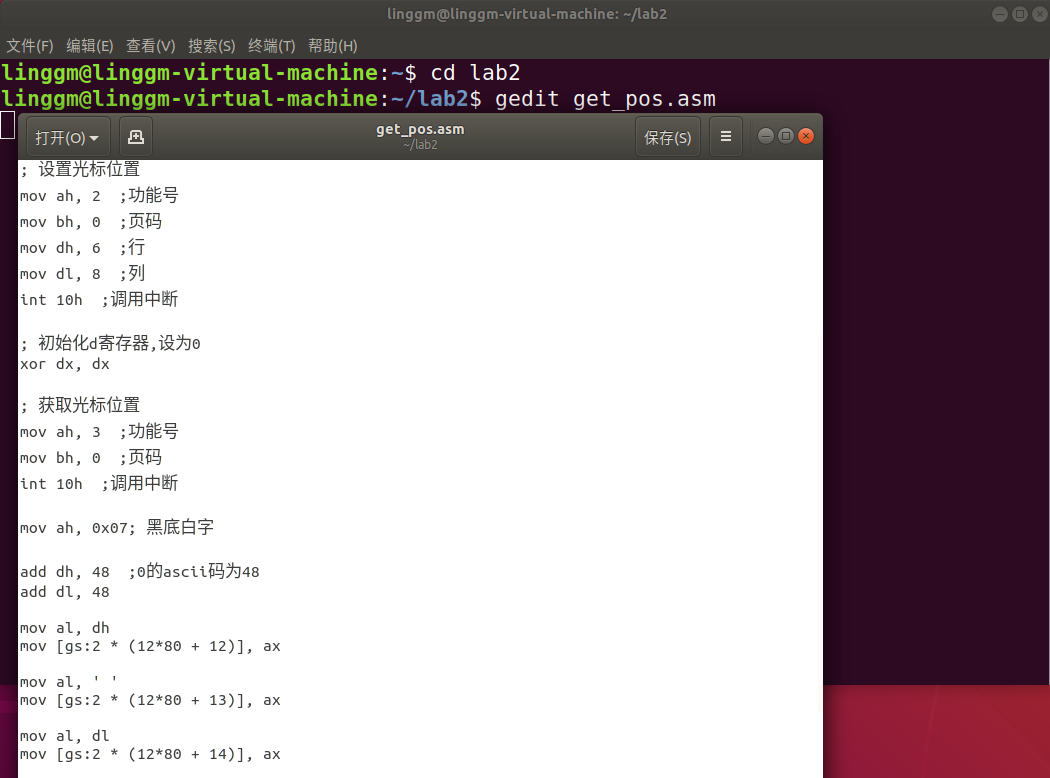
* 到达断点后按c继续运行，成功显示学号，进入死循环



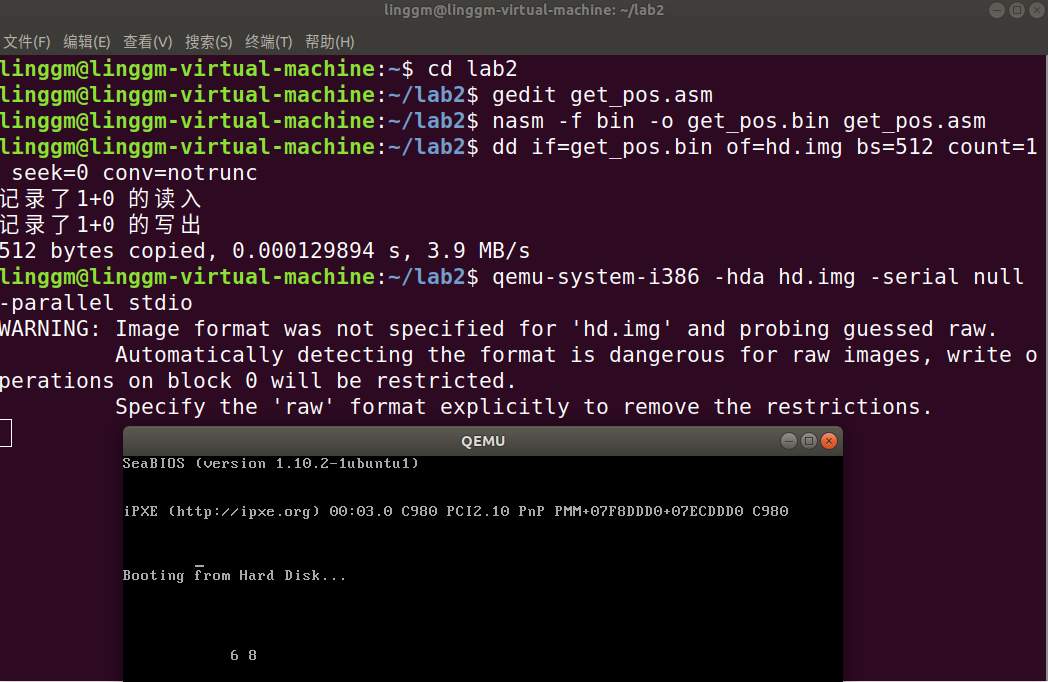
至此，1.2显示学号的任务完成

2.2.1 利用int10h中断，设置光标位置，然后获取光标位置

* 第一步：编写汇编程序



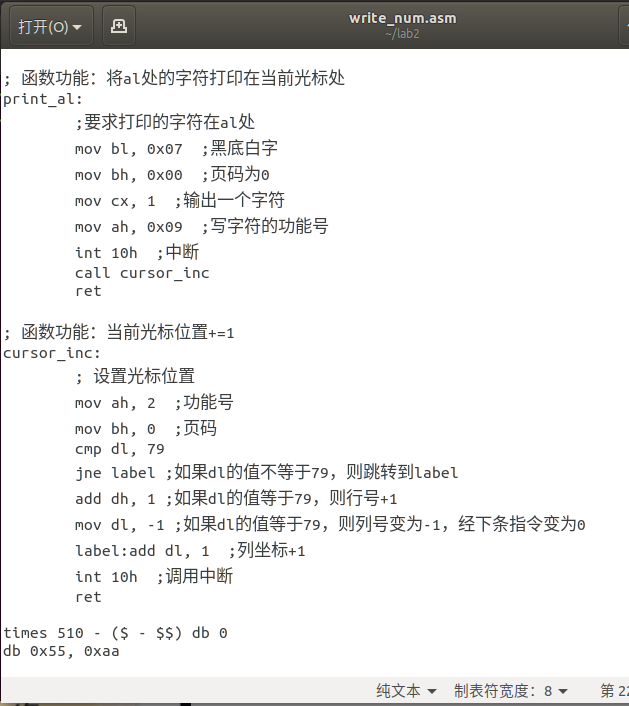
* 第二步，按照example1的过程，将程序载入磁盘，启动qemu测试



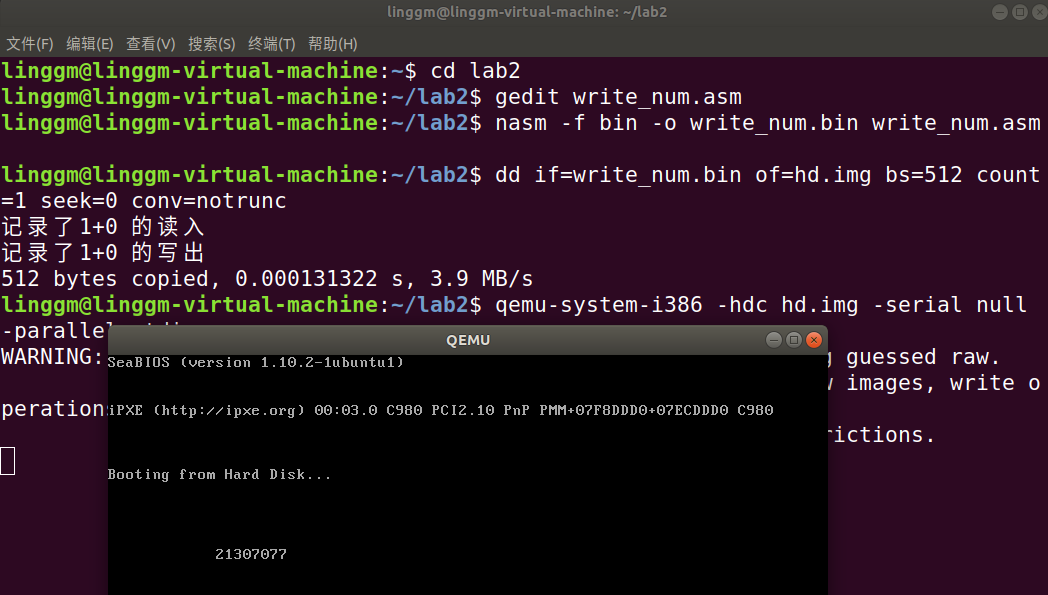
成功设置光标行号为6，列号为8。成功获取光标位置并输出成功。

2.2.2 利用int10h中断，打印我的学号

* 第一步：编写汇编程序（完整代码见3.2代码部分）



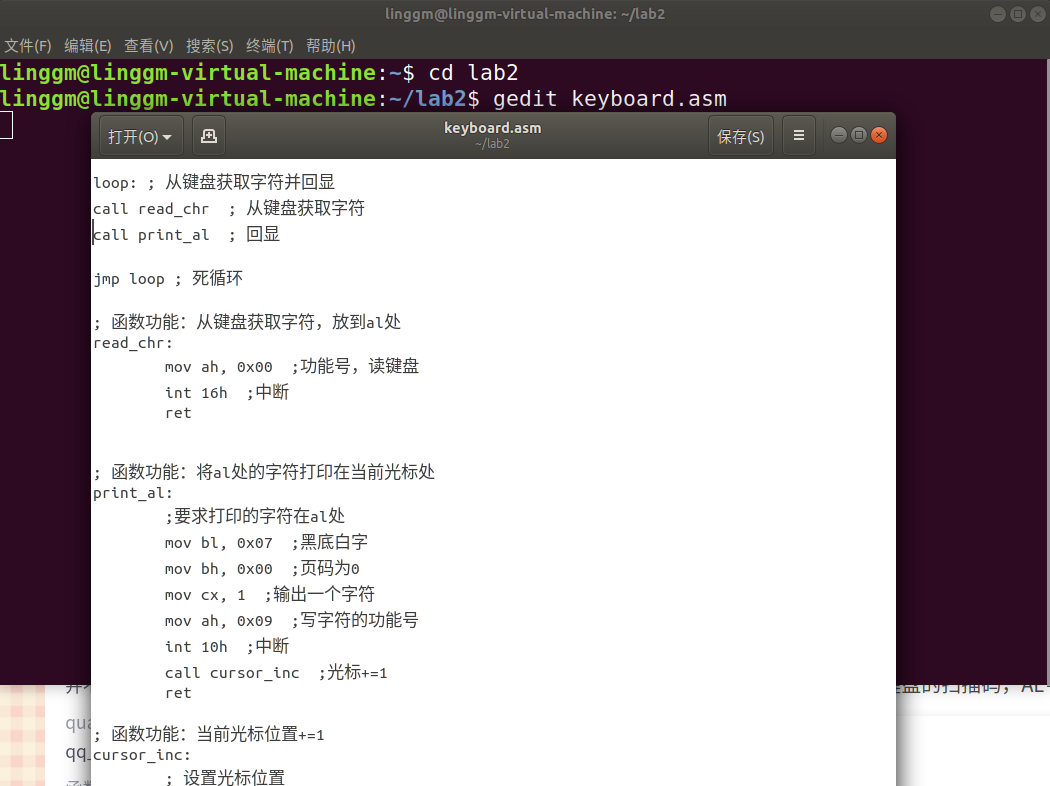
* 第二步，按照example1的过程，将程序载入磁盘，启动qemu测试



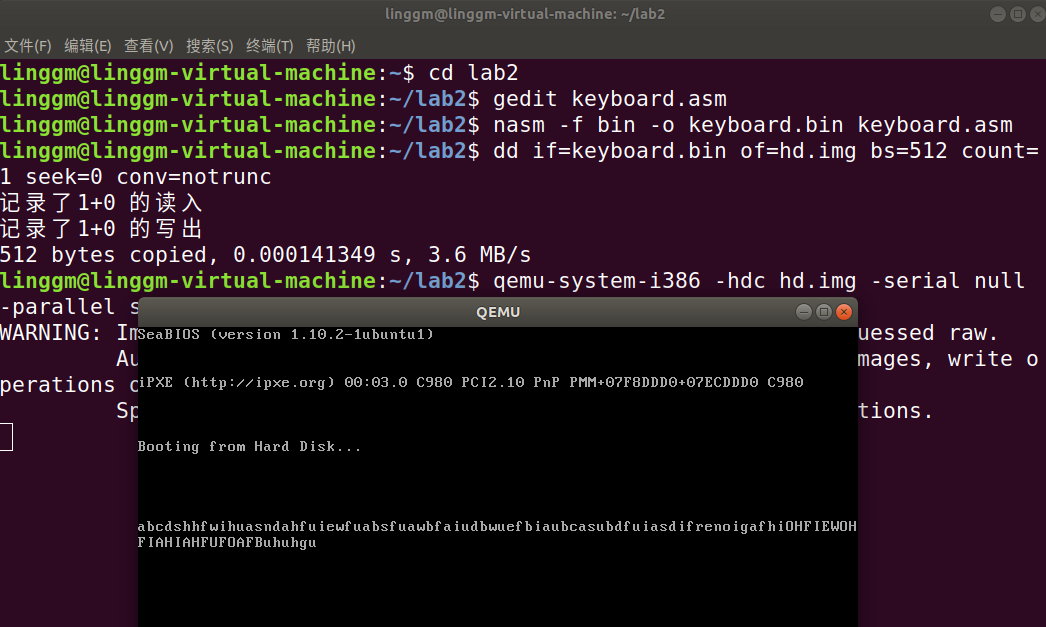
成功利用int10h中断，在(12,12)(12,12)处输出我的学号21307077

2.2.3 探索实模式的键盘中断，**利用键盘中断实现键盘输入并回显**

* 第一步：编写汇编程序，完整程序及解释见3.2代码部分

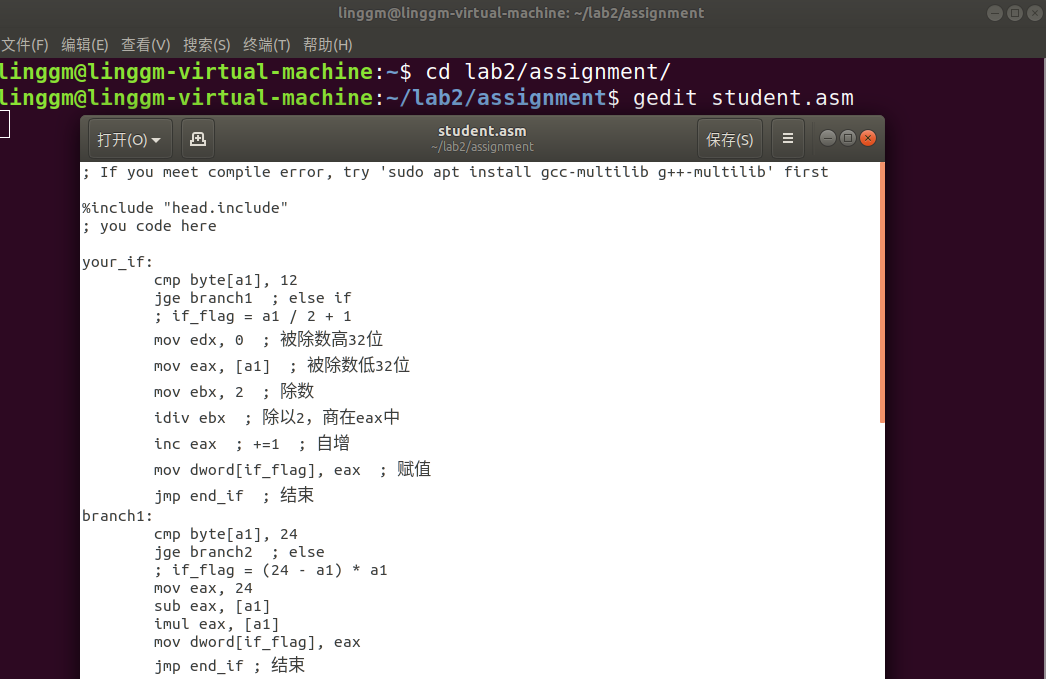


* 第二步，按照example1的过程，将程序载入磁盘，启动qemu测试

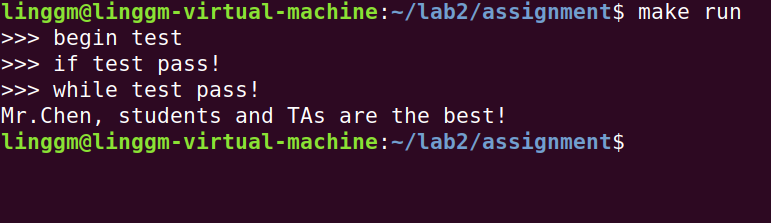


实现了从一直从键盘读字符，并输出到当前光标处，再将光标+1

* 1. 用x86汇编语言实现**分支逻辑，循环逻辑和函数调用**
* 第一步：



* 第二步



* 1. 字符弹射程序

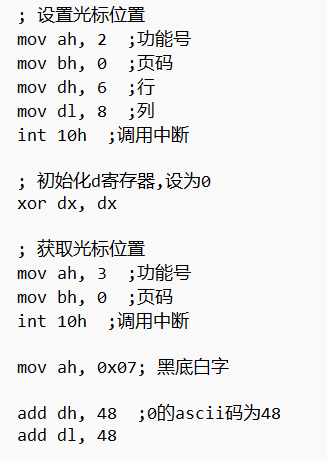
步骤同2.2，代码见3.4，结果见4.4

1. **关键代码**

3.1 任务1，输出学号的代码

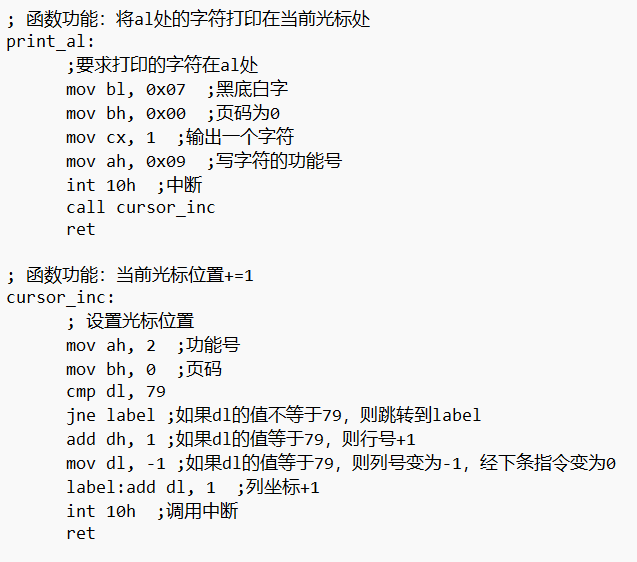
仅展示与example1不同的代码

3.2.1 任务2，设置光标、获取光标位置的代码

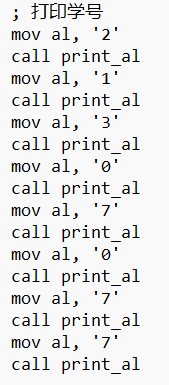


先设置光标位置到第6行第8列，再初始化dx寄存器，然后调用int10h中断，将光标位置读入dx中，传送到显存中，输出到屏幕上。接下来的部分与example1相同，不作展示

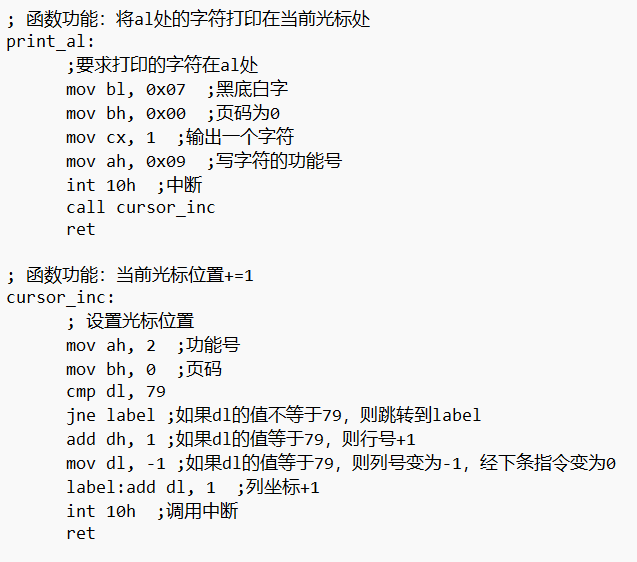
3.2.2 任务2，利用int10h中断打印学号的代码



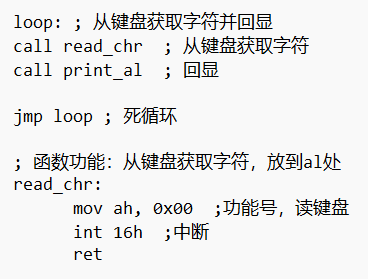
函数print\_al的功能是将al处的字符打印在当前光标处，cursor\_inc的功能是将当前光标位置+1（列坐标在79前，列坐标加一，列坐标在79时，列坐标置为0，行坐标加一），每次更改al中字符的值，然后调用print\_al即可



3.2.3 任务2，利用int16h中断从键盘获取字符并回显的代码



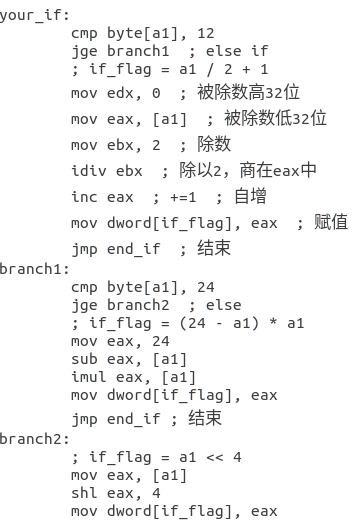
接着用上一个任务的两个函数



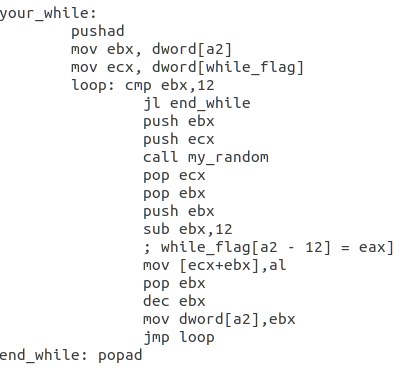
新增一个函数：利用int16中断读取键盘。

然后在一个死循环里不断读键盘，输出，读键盘，输出即可

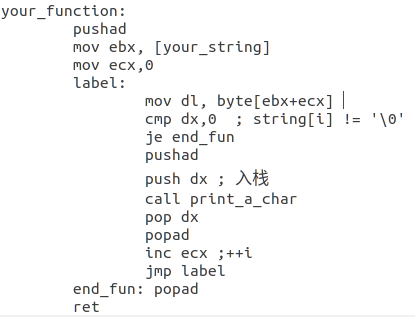
* + 1. 任务3，分支逻辑

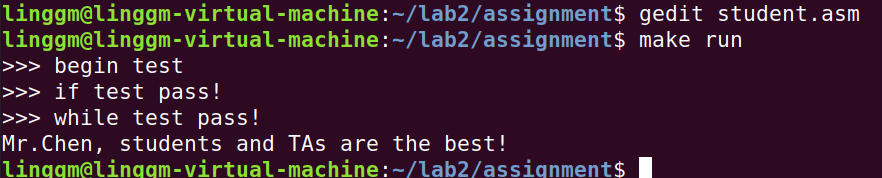


* + 1. 任务3，循环逻辑

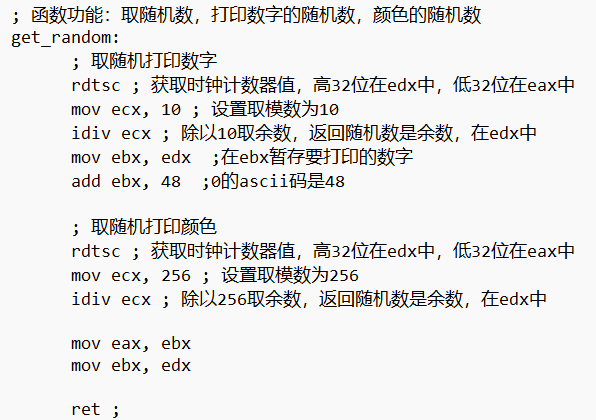


* + 1. 任务3，函数编写

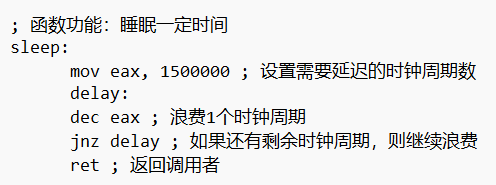




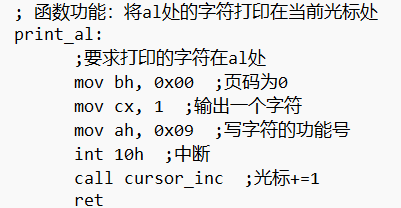
3.4 任务4，字符弹射小程序



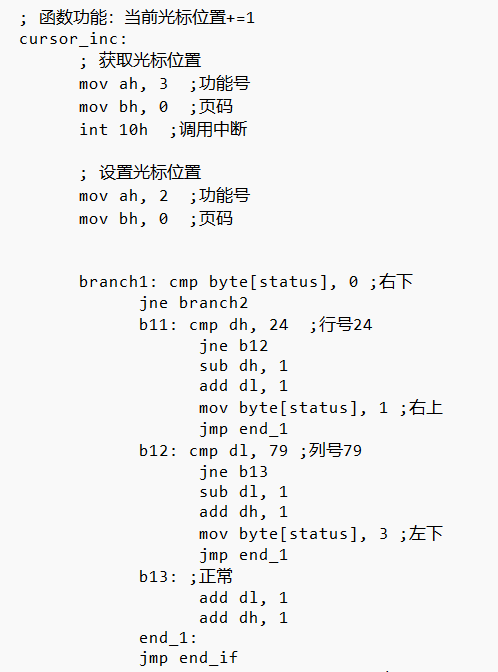
使用随机数，取得打印字符的颜色及数字

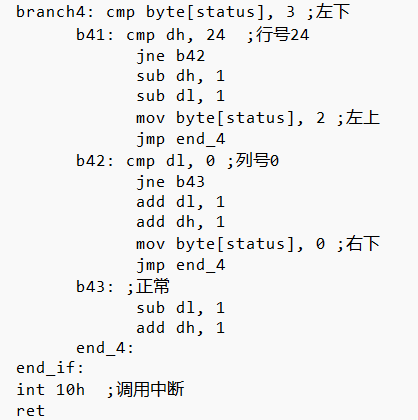


睡觉

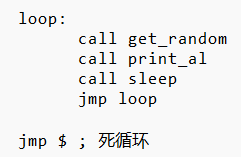


打印一个字符





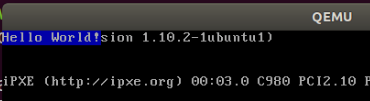
由于代码过长，仅展示最前面和最后面的部分。主要是嵌套两层的if-elif-else语句。外面的一层判断当前移动方向。里面一层判断是否碰到边界，碰到哪个边界，从而决定移动的方向。



主函数，死循环，每次循环随机决定打印的字符及其颜色，然后将其打印出来，然后光标移动，然后睡眠一段时间，然后继续循环

1. **实验结果**

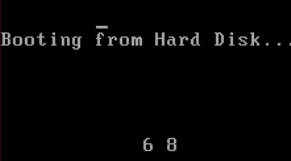
4.1.1 复现example1



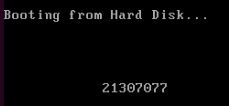
4.1.2 输出学号



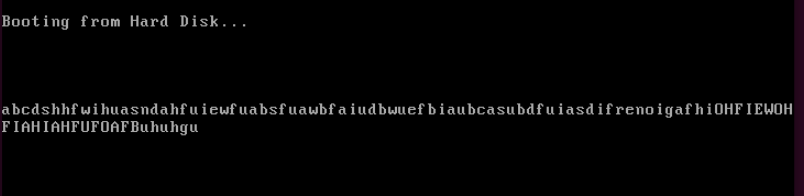
4.2.1 设置光标位置，获取光标位置



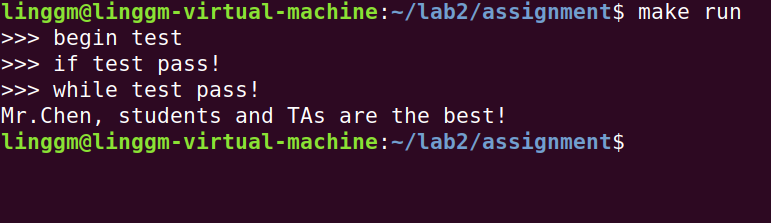
4.2.2 利用int10h中断输出学号



4.2.3 利用int16h从键盘获取字符并回显



4.3



4.4

