

本科生实验报告

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

# 实验名称:\_\_\_\_\_\_Lab2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

专业名称:\_\_\_\_\_\_信息与计算科学\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生姓名:\_\_\_\_\_\_张文沁\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生学号:\_\_\_\_\_\_20337268\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验地点:\_\_\_\_\_无固定地点\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

报告时间:\_\_\_\_\_\_2022.3.6\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **实验要求**

熟悉32位 Intel 汇编语言的基本语法；

熟悉汇编语言的编译、链接过程；

掌握简单编程；

掌握磁盘、显示I/O操作；

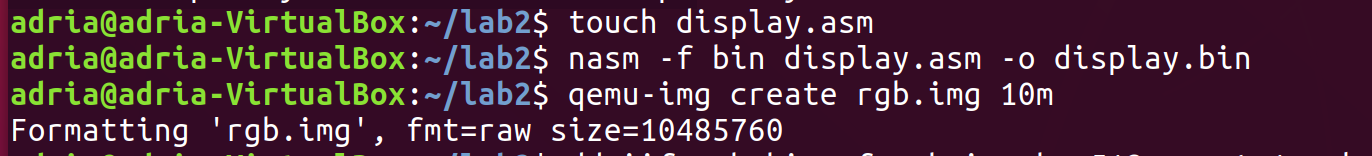
掌握汇编程序调用；

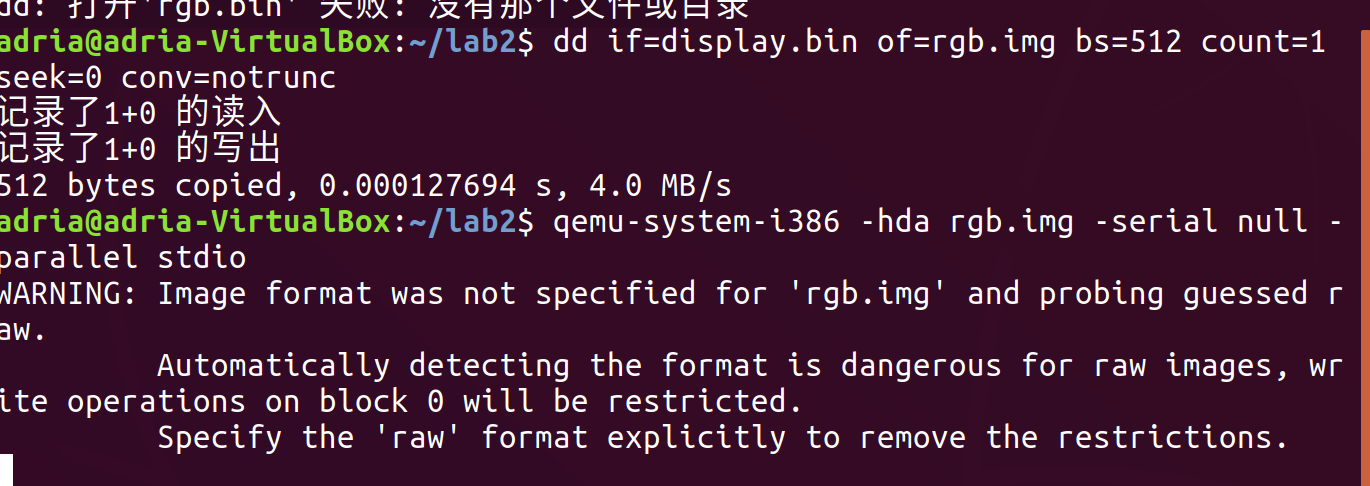
掌握C语言与汇编语言之间的调用；

1. 利用32位汇编程序从一个数组中找出最大的数（算法不限）；
2. 利用32位汇编计算前20个Fibonacci数列；
3. 利用汇编清理屏幕的显示内容并定位光标；
4. 利用C语言和汇编语言实现磁盘文件的拷贝以及显示体现C语言与汇编语言之间的调用关系
5. 利用不同颜色显式字符串（可选）；
6. **实验过程**
   1. 利用32位汇编程序从一个数组中找出最大的数
      1. 学习nasm汇编语言基础语法
      2. 编写C程序
      3. 翻译成汇编语言
      4. 调试
   2. 利用32位汇编计算前20个Fibonacci数列
      1. 同上
   3. 利用汇编清理屏幕的显示内容并定位光标
      1. 参考博客完成
      2. 需要的工具：qemu
      3. 原理：通过BIOS中断，来滚动屏幕，达到清屏的效果
      4. 光标位置：通过中断int 10h 可以在文本坐标下设置光标位置
   4. 利用C语言和汇编语言实现磁盘文件的拷贝以及显示体现C语言与汇编语言之间的调用关系
      1. 参考博客完成，但是只完成了磁盘文件的拷贝和显示
      2. 代码非本人完成，只是尝试了解了代码逻辑和实现方式
      3. 读取硬盘
      4. 逻辑：

1）sector count寄存器寄存器写入读取的扇区数  
2）LBA low寄存器，LBA mid寄存器，LBA high寄存器写入lba地址  
3）device寄存器写入lba地址和读取模式  
4）command寄存器写入写入命令  
5）读取两个字节数据，多次循环直到读取完扇区数据。

* 1. 利用不同颜色显式字符串:
     1. 了解原理
     2. 编写代码
     3. 编译
     4. 调动qemu







1. **关键代码**
   1. 利用32位汇编程序从一个数组中找出最大的数
2. extern printf
3. extern exit
4. section .data
5. **array** dd 1,2,3,4,5,6,7,8,9,10
6. format db "%d"
7. section .text
8. global \_start
9. \_start:
10. mov eax,0 ；记录最大值
11. mov ebx,**array** ；将第一个数字传入ebx
12. mov ecx,10 ；ecx记录数组长度，即循环次数
13. Loop:
14. cmp [ebx],eax
15. jl next ；大于就跳转
16. mov eax,[ebx] ；否则更新数字
17. next:
18. add ebx,4 ；到下一个数字，偏移寻址
19. loop Loop ；继续循环
20. push eax
21. push format
22. call printf
23. push 0
24. call exit
    1. 利用32位汇编计算前20个Fibonacci数列
25. %include 'functions.s'
27. section.text
28. global \_start
29. \_start:
31. mov ecx, 20
32. mov eax, 1
33. mov ebx, 1
35. fibonacci:
36. call iprintLF
37. add ebx, eax
38. mov edx, eax
39. mov eax, ebx
40. mov ebx, edx
41. dec ecx
42. jnz fibonacci
43. call quit
44. section.data
    1. 利用汇编清理屏幕的显示内容并定位光标
       1. 光标初始化
45. Init\_Cusor: ; 光标位置初始化
46. mov ah,0x02
47. mov bh,0
48. mov dx,0
49. int 0x10
    * 1. 清屏
50. Clear\_Screen: ;清除屏幕
51. mov ah,0x06
52. mov al,0
53. mov cx,0
54. mov df,0xffff
55. mov bh,0x17             ;属性为蓝底白字
56. int 0x10
    1. 利用C语言和汇编语言实现磁盘文件的拷贝以及显示体现C语言与汇编语言之间的调用关系

读取磁盘

1. .check\_status:;检查磁盘状态
2. nop
3. in al,dx
4. and al,0x88         ;第4位为1表示硬盘准备好数据传输，第7位为1表示硬盘忙
5. cmp al,0x08
6. jnz .check\_status   ;磁盘数据没准备好，继续循环检查


10. ;设置循环次数到cx
11. mov ax,cx           ;乘法ax存放目标操作数
12. mov dx,256
13. mul dx
14. mov cx,ax           ;循环次数 = 扇区数 x 512 / 2
15. mov bx,di
16. mov dx,0x1F0
18. .read\_data:
19. in ax,dx            ;读取数据
20. mov [bx],ax         ;复制数据到内存
21. add bx,2            ;读取完成，内存地址后移2个字节
23. loop .read\_data
24. ret

输出字符串：

1. ;程序核心内容
2. Entry:
3. ;---------------------------
4. ;输出字符串
5. mov si,HelloMsg         ;将HelloMsg的地址放入si
6. mov dh,0                ;设置显示行
7. mov dl,0                ;设置显示列
8. call Func\_Sprint            ;调用函数
9. jmp $           ;让CPU挂起，等待指令
   1. 利用不同颜色显式字符串:

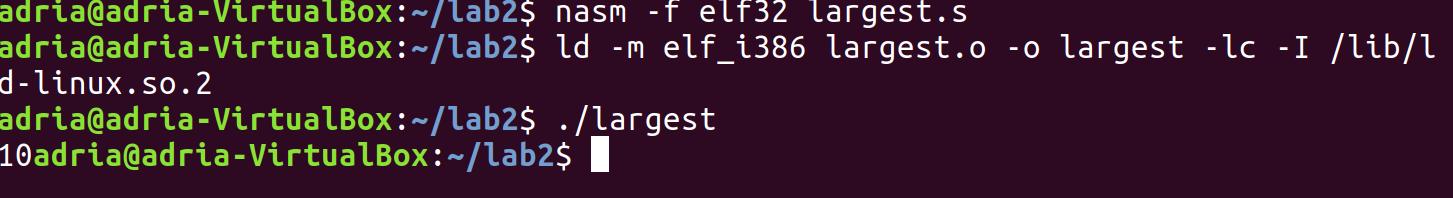
编译指令：

1. nasm -f bin mbr.asm -o mbr.bin //-f输出的文件格式，-o输出的文件名。mbr.bin保存机器指令
2. qemu-img create filename [size] //filename生成的硬盘文件名
3. qemu-img create hd.img 10m
4. dd **if**=mbr.bin of=hd.img bs=512 count=1 seek=0 conv=notrunc
5. //if输入文件。
6. //of输出文件。
7. //bs表示块大小，以字节表示。
8. //count表示写入的块数目。
9. //seek表示越过输出文件中多少块之后再写入。
10. //conv=notrunc表示不截断输出文件，如果不加上这个参数，那么硬盘在写入后多余部份会被截断。
11. qemu-system-i386 -hda hd.img -serial null -parallel stdio
12. //-hda hd.img表示将文件hd.img作为第0号磁盘映像。
13. //-serial dev表示重定向虚拟串口到空设备中。
14. //-parallel stdio 表示重定向虚拟并口到主机标准输入输出设备中。

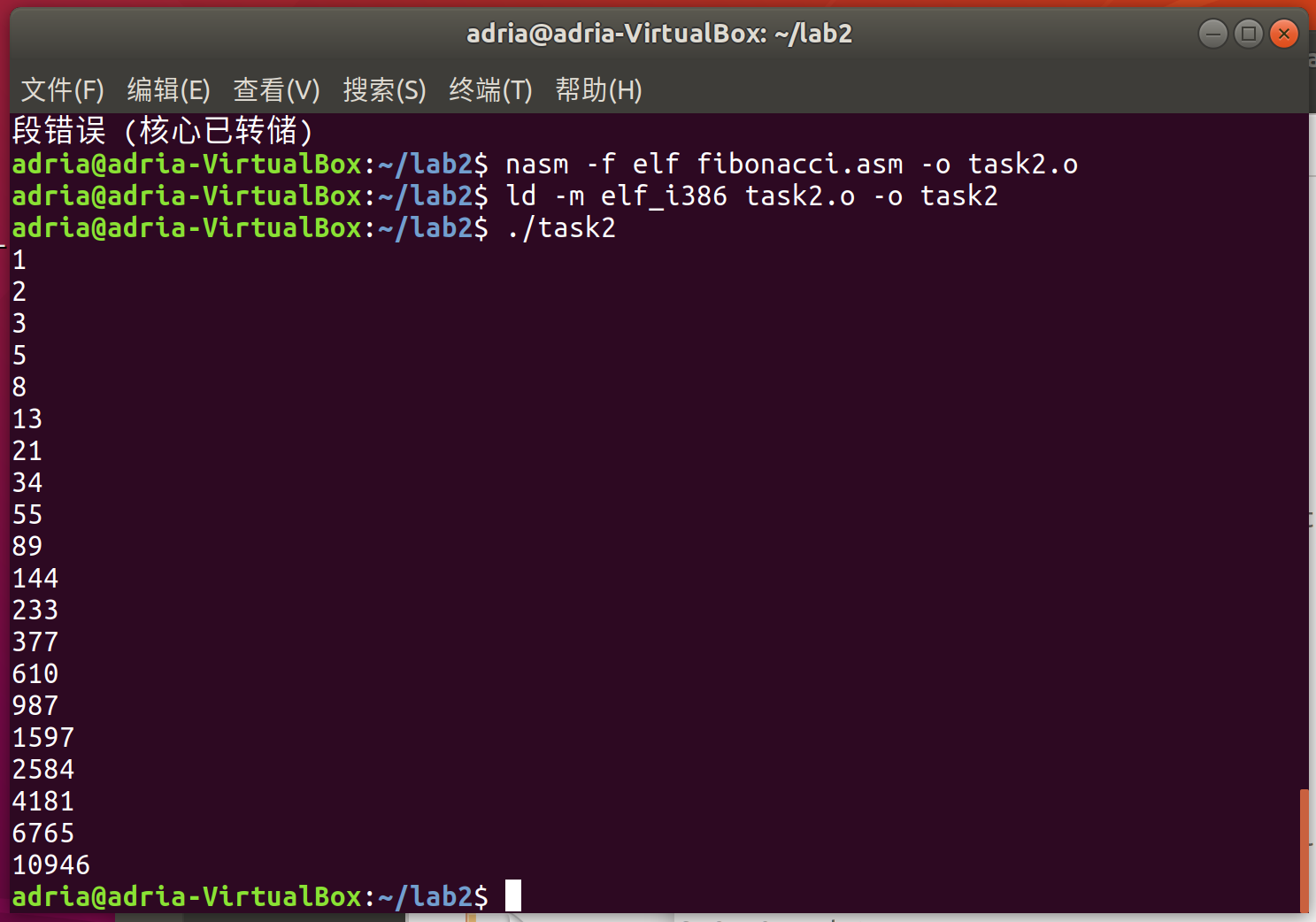
汇编代码：

1. org 0x7c00  ;代码中的代码标号和数据标号从0x7c00开始
2. [bits 16]  ;代码为16位格式
3. **xor** ax, ax ; ax初始化为0
4. ; 初始化段寄存器, 段地址全部设为0
5. mov ds, ax
6. ; 初始化栈指针
7. mov sp, 0x7c00 ;
8. mov ax, 0xb800
9. mov gs, ax
10. mov ah, 0x01 ;蓝色
11. mov al, 'H'
12. mov [gs:2 \* 0], ax
13. mov ah, 0x02  ;绿色
14. mov al, 'e'
15. mov [gs:2 \* 1], ax
16. mov ah, 0x03  ;浅蓝色
17. mov al, 'l'
18. mov [gs:2 \* 2], ax
19. mov ah, 0x04  ;红色
20. mov al, 'l'
21. mov [gs:2 \* 3], ax
22. mov ah, 0x05  ;粉色
23. mov al, 'o'
24. mov [gs:2 \* 4], ax
25. mov ah, 0x06
26. mov al, ' '
27. mov [gs:2 \* 5], ax
28. mov ah, 0x07  ;白色
29. mov al, 'W'
30. mov [gs:2 \* 6], ax
31. mov ah, 0x08  ;灰色
32. mov al, 'o'
33. mov [gs:2 \* 7], ax
34. mov ah, 0x00
35. mov al, 'r'  ;黑色
36. mov [gs:2 \* 8], ax
37. mov ah, 0x01
38. mov al, 'l'
39. mov [gs:2 \* 9], ax
40. mov al, 'd'
41. mov [gs:2 \* 10], ax
42. jmp $ ; 死循环
43. times 510 - ($ - $$) db 0
44. db 0x55, 0xaa
45. **实验结果**
    1. 利用32位汇编程序从一个数组中找出最大的数

输入的数组为1-10，输出为10

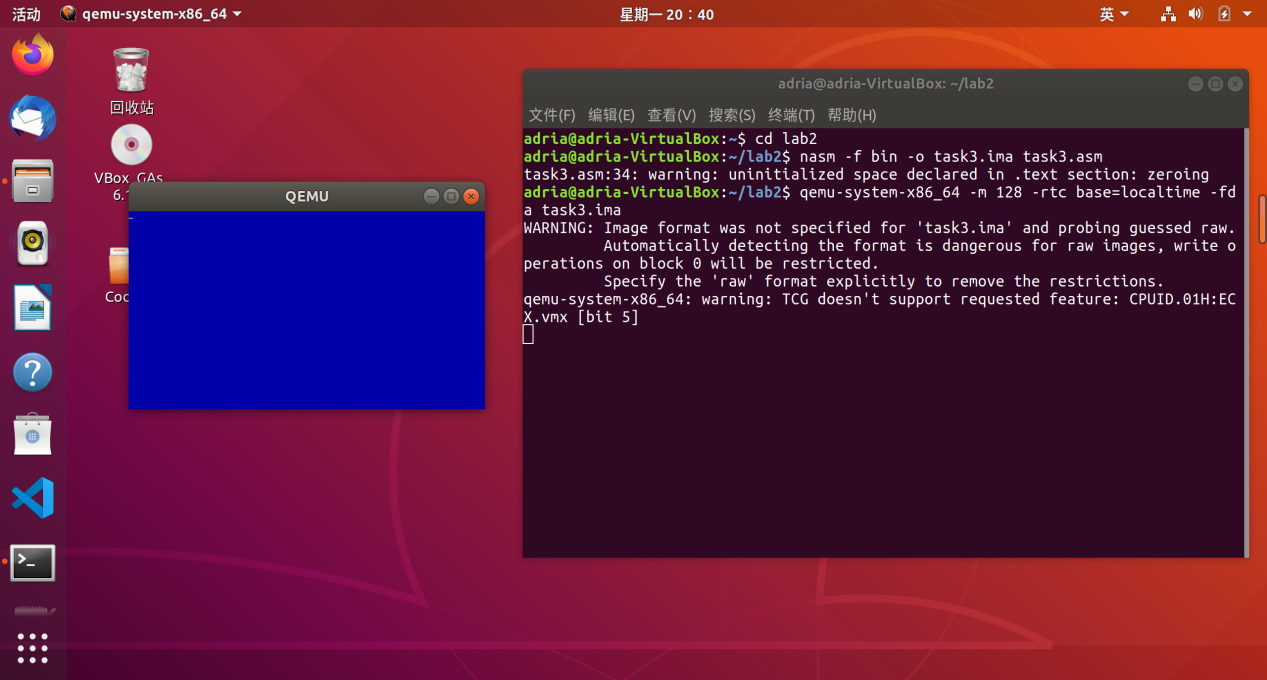


* 1. 利用32位汇编计算前20个Fibonacci数列



* 1. 利用汇编清理屏幕的显示内容并定位光标

如图所示，用qemu完成了清屏



* 1. 利用C语言和汇编语言实现磁盘文件的拷贝以及显示体现C语言与汇编语言之间的调用关系

输出蓝底白字“Hello world”

* 1. 利用不同颜色显式字符串:



1. **总结**
   1. 利用32位汇编程序从一个数组中找出最大的数
      1. 汇编语言分很多种，nasm和其他版本的语法相去甚远，查找资料的过程中要注意分辨
      2. 连接C库的时候所用的指令需要加上 -I /lib/ld-linux.so.2
      3. Mov ecx,10这段代码一定需要出现
   2. 利用32位汇编计算前20个Fibonacci数列
      1. 逻辑很简单，但是调用C函数总是无法开始循环，只输出三个数字便会结束循环。
      2. 查询之后发现可能是ecx和printf有牵连关系，导致无法完成循环，故使用了教程网站中的头文件中的输出函数
   3. 跟着博客和实验指导完成了任务，大致了解了一些中断类型及其用处，题目中用int 10h完成了中断。考虑并发性、用户态到核心态的转换依赖于中断完成。
2. 参考资料：
   1. <https://asmtutor.com/>
   2. <https://www.cnblogs.com/mlzrq/p/10223028.html>
   3. <https://www.cnblogs.com/mlzrq/p/10223060.html>