实验准备：

1. 在安装ubuntu系统虚拟机和配置环境时出现了很多问题：
2. 误操作导致安装虚拟机失败，反复重装了好多次
3. 虚拟机安装完成后，由于安装的是最新的18.03版本系统，导致系统GCC版本与实验文件GCC版本不同导致失败，花费了半天时间才找到了解决办法(从MIT上下载最新的实验文件)继续实验
4. 在使用gdb调试时，由于翻译原因直接输入gdb失效，最后查看英文原版实验步骤才解决问题(输入make gdb)

练习2：起初对使用gdb和si命令得到的代码一头雾水，最后翻找查询了很多资料才明白

练习3：

1. 在追踪boot.S和main.c代码过程中晕头转向，完全懵逼，最后反复了好几遍才完成
2. 在追踪main.c时对其代码理解有难度
3. 问题回答：
4. 在ljmp $PROT\_MODE\_CSEG, $protcseg处开始执行32位代码。在经过64端口和60端口控制下加载GDT表之后，使CRO的bit0位置1，让机器进入保护模式，转换16位为32位
5. 引导加载程序的最后一条指令是 ((void (\*)(void)) (ELFHDR->e\_entry))(); 加载完的内核的第一条指令movw $0x1234, 0x472
6. entry.S文件中
7. 通过ELF program headers决定读取多少扇区，在操作系统内核映像文件的ELF头部信息里能找到这些信息

练习5：

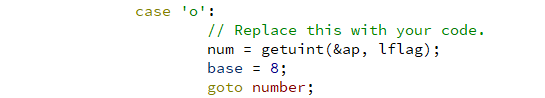
1. ELF文件理解有难度，最后网上查询资料得以了解。
2. 修改链接地址后重新编译产生了错误，错误是在将处理器转成32位时发生的

练习6：8个字不一样是bootmain将内核的一个section存入了该地址

第二次打印的东西可能是.text段的内容

练习7：在注释掉movl %eax, %cr0后重新编译之后断点在0x0010000c处，单步运行后，在movl $(bootstacktop),%esp处出错，实验时无法找到错误原因，求助网上得知movl %eax, %cr0后完成了分页操作，注释之后，没有分页，无法访问0xf010002c，即访问的逻辑地址超出内存了。分页后的变化不太了解

练习8：



1. 开始不太理解三个程序功能与联系，后来知道了
2. 回答：
3. console.c的功能时实现I/O设备的输入输出，console.c中cputchar函数被printf.c在putch函数中调用
4. 实现写满一个屏幕后向上滚动一行
5. fmt指x %d, y %x, z %d\n，ap指x, y, z，cons\_putc所有输入参数：x 1, y 3, z 4。va\_arg调用前ap中包括x，y，z三个参数的内容：1，3，4。调用完成后只剩下y，z的内容：3,4。vcprintf两个参数为x %d, y %x, z %d\n和x, y, z
6. 输出是He110 World 大端格式的话：还是57616

i=0x726c6400

1. y=后打印一个不确定的值，因为y的值没有给定
2. 挑战：修改console.c中的cga\_putc函数中的c|=后的值可改变颜色

练习9：

1. 代码真的难懂啊！！
2. 回答：设置内核堆栈的指令就应该是entry.S中的：movl $0x0,%ebp和movl $(bootstacktop),%esp。堆栈在内核什么地方，这可咋找啊，没头绪，只能上网搜索，用于堆栈的地址空间为 0xf0108000-0xf0110000，其中栈顶指针指向0xf0110000. 那么这个堆栈实际坐落在内存的 0x00108000-0x00110000物理地址空间中。在entry.S中的数据段里面声明一块大小为32Kb的空间作为堆栈使用。从而为内核保留了一块空间。堆栈指针指向最高地址bootstacktop的值

练习10：我好像没看懂

练习11：写的有点懵

练习12：结构体Eipdebuinfo理解错了 崩溃

debuginfo\_eip()看不懂 崩溃

stab\_binsearch()看不懂，崩溃