**实验5:加载操作系统映像并进入C语言编写的main函数**

**姓名：姜庆彩**

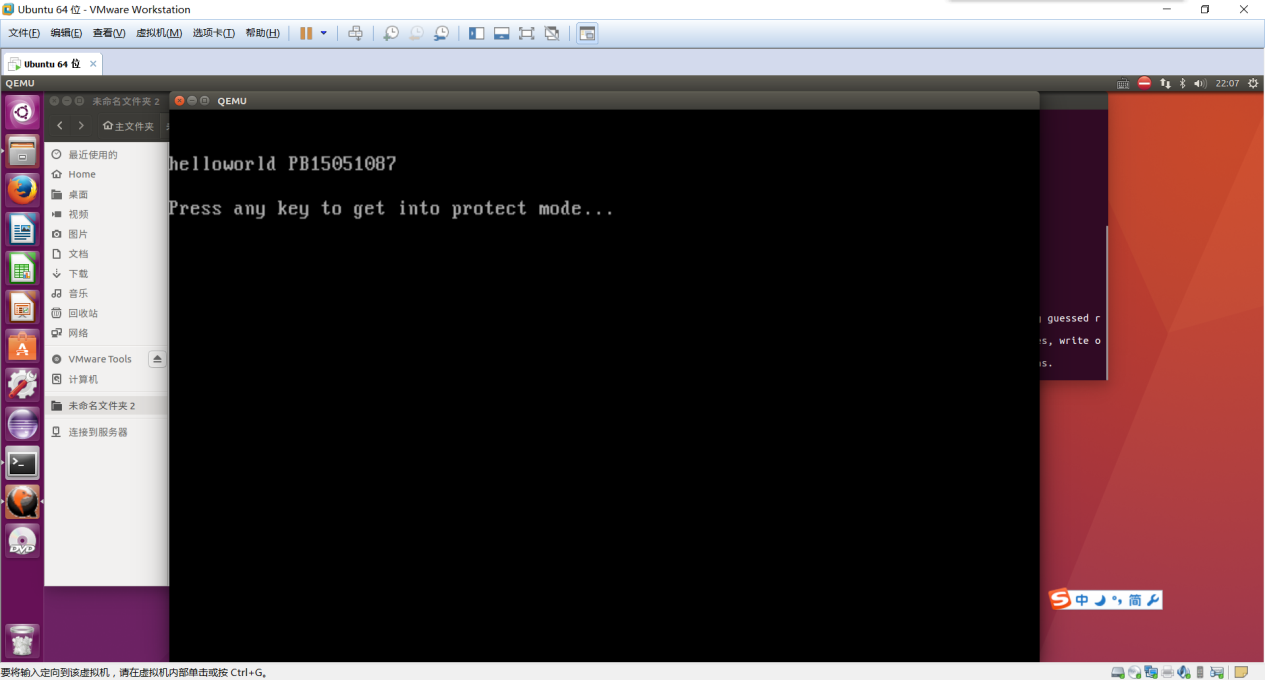
**学号：PB15051087**

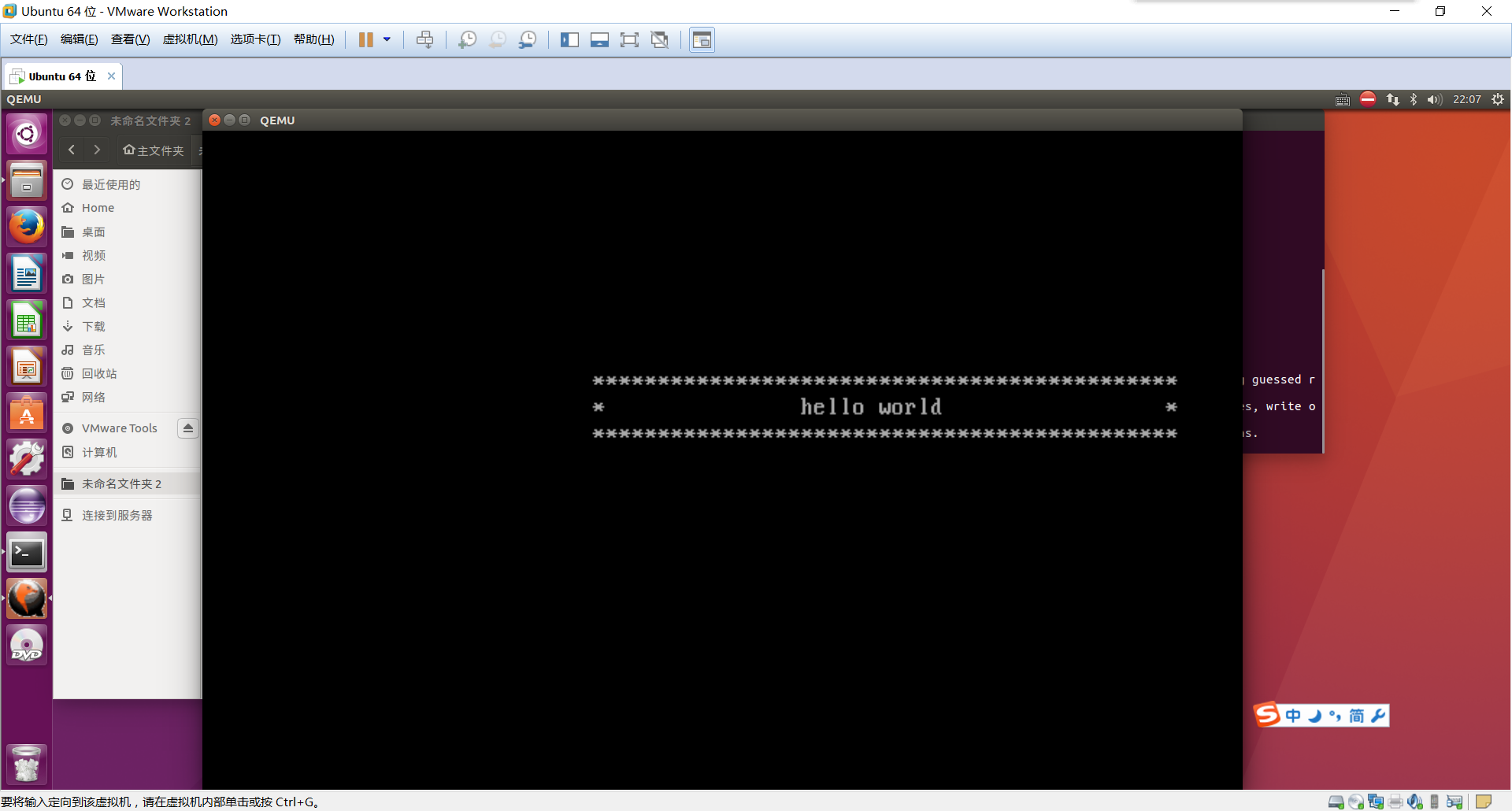
* **实验内容：**

1. **学习制作简单的makefile**
2. **学习制作一个简单的操作系统映像**
3. **学习从启动代码转入操作系统代码运行**
4. **为C函数的运行做好适当的准备**
5. **文件清单：start16.S, start32.S, main.c, start16.ld, myOS.ld, makefile, source2img.sh**

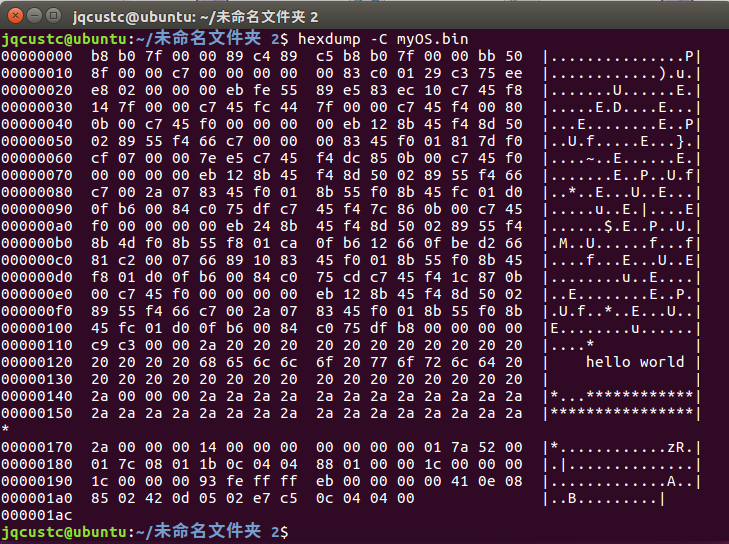
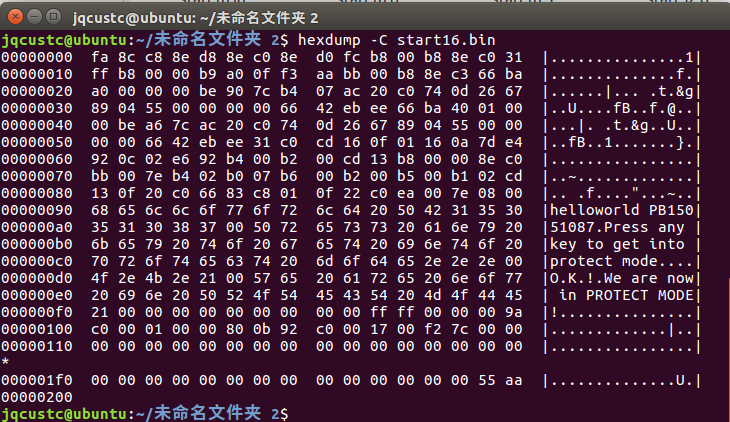
* **实验结果**

**执行shell脚本指令后**

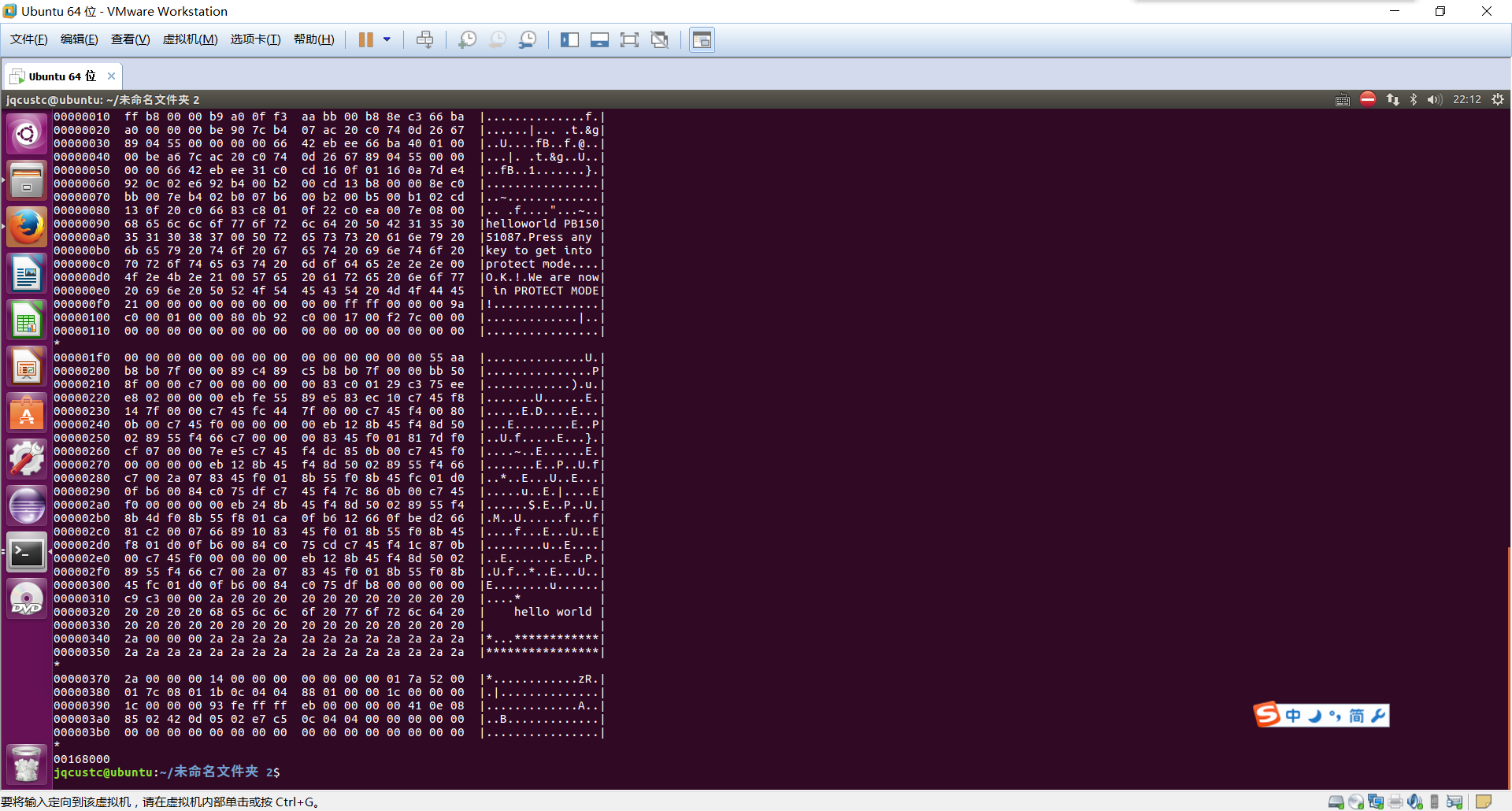
****



**bin文件**



**img文件**



* **make工具的作用是什么？在命令行上输入“make OS”的含义是什么？Makefile中规则的语法是什么？**

**make工具的作用就是当一个模块被修改后，保证其他模块与之相关的部分也随之改变，进而不会影响模块之间的协调关系。**

**make本身是一个单独工作的程序，可以根据程序模块的修改情况重新编译链接目标代码，以保证目标代码总是由它的最新模块组成。**

**在命令行上输入“make OS”就会执行标签为OS 后面的指令。**

**Makefile语法规则：**

**target... : prerequisites ...**

**command**

**...**

**...**

**-----------------------------------------------------------------**

**target也就是一个目标文件，可以是Object File，也可以是执行文件。还可以是一个标签（Label），对于标签这种特性，在后续的“伪目标”章节中会有叙述。prerequisites就是，要生成那个target所需要的文件或是目标。command也就是make需要执行的命令。（任意的Shell命令）**

* **何时加载操作系统映像合适？加载多少个扇区合适？**

**在进入保护模式前合适，因为此时PE还未设置为1,；加载7个扇区合适。**

* **如何利用BIOS软盘驱动加载操作系统映像？**

**利用BIOS中断指令int$0x13后对扇区读写，加载操作系统映像。**

* **从汇编进入C需要做哪些准备？栈在什么位置比较合适？什么是BSS段？BSS段清0有什么好处**

**准备：需要找到起始位置，将esp与ebp指向起始位置，将BSS清空。在代码段与数据段后面比较合适。**

**BSS（Block Started by Symbol）通常是指用来存放程序中未初始化的全局变量和静态变量的一块内存区域。特点是:可读写的，在程序执行之前BSS段会自动清0。**

**好处：未初始的全局变量在程序执行之前已经成0了。**

* **使用C语言编写写VGA缓存和汇编写VGA缓存有什么不一样？附加问题：你能不能从C语言中调用汇编写的VGA输出函数？你能不能从汇编调用C写的VGA输出函数？说出你的方法。**

**C语言写VGA缓存只需要把指针指向待写入的内存地址，再写入值。**

**汇编写VGA缓存通过lodsb读入字符串，再循环输出至VGA内。**

**C语言的函数名相当于汇编的函数标签入口，用内嵌汇编操作可以完成。**

* **代码的流程图**
* **文件作用：**
* start16.S：加载操作系统映像，输入提示字符串，进入保护模式，并跳转到0x7e00执行操作系统代码；

start32.S：初始化栈指针，清楚bss 栈，调用main 函数

main.c：清屏，写VGA 显存，输出hello world

start16.ld：对start16.S 的各个段进行链接

myOS.ld：对start32.S 的各个段进行链接

makefile：自动编译和链接上述文件

* **如果你的main函数最后不是死循环，请说明main函数返回后你的操作系统在执行什么？**

**由于main在start32.中调用，返回后执行调用后的指令。**