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

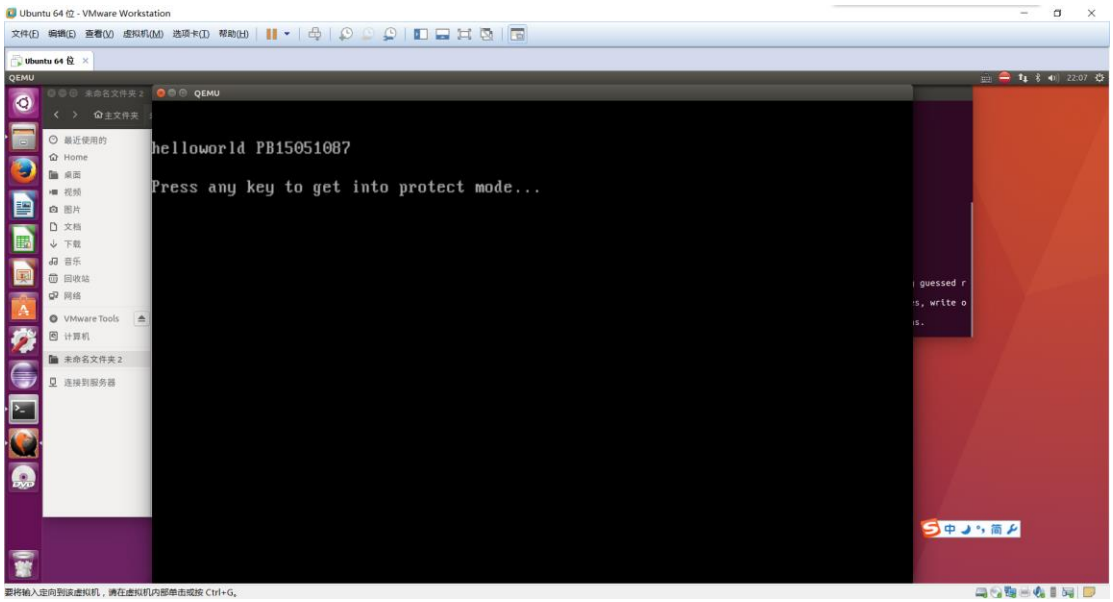
学号：PB15051087

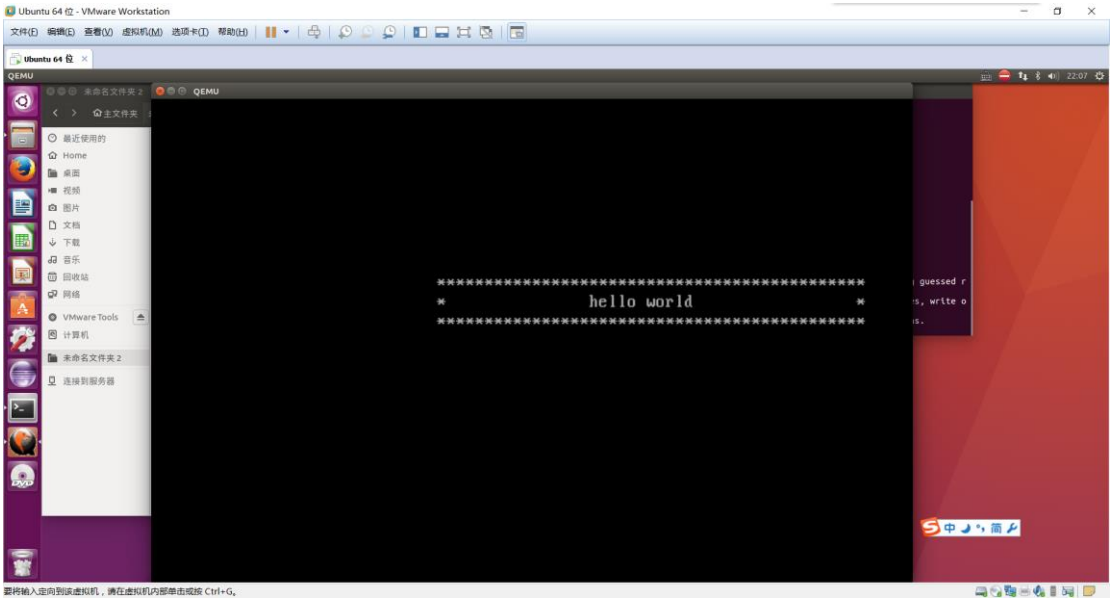
- 实验内容：

  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


Ubuntu 64 位 - VMware Workstation
QEMU
helloworld PB15051087
Press any key to get into protect mode...

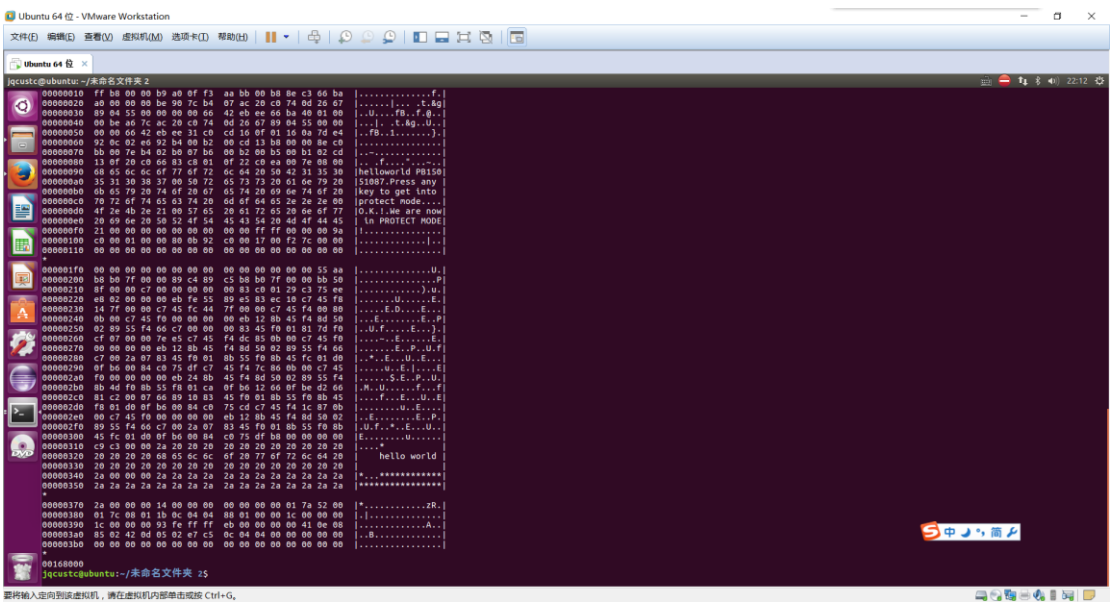

Ubuntu 64 位 - VMware Workstation
QEMU
hello world

bin 文件

```
jqcustc@ubuntu: ~/未命名文件夹 2
jqcustc@ubuntu:~/未命名文件夹 2$ hexdump -C start16.bin
00000000  fa 8c c8 8e d8 8e c0 8e  d0 fc b8 00 b8 8e c0 31  |...............1|
00000010  ff b8 00 00 b9 a0 0f f3  aa bb 00 b8 8e c3 66 ba  |..............f.|
00000020  a0 00 00 00 be 90 7c b4  07 ac 20 c0 74 0d 26 67  |......|... .t.&g|
00000030  89 04 55 00 00 00 00 66  42 eb ee 66 ba 40 01 00  |..U....fB..f.@..|
00000040  00 be a6 7c ac 20 c0 74  0d 26 67 89 04 55 00 00  |...|. .t.&g..U..|
00000050  00 00 66 42 eb ee 31 c0  cd 16 0f 01 16 0a 7d e4  |..fB..1.......}.|
00000060  92 0c 02 e6 92 b4 00 b2  00 cd 13 b8 00 00 8e c0  |................|
00000070  bb 00 7e b4 02 b0 07 b6  00 b2 00 b5 00 b1 02 cd  |..~.............|
00000080  13 0f 20 c0 66 83 c8 01  0f 22 c0 ea 00 7e 08 00  |.. .f...."...~..|
00000090  68 65 6c 6c 6f 77 6f 72  6c 64 20 50 42 31 35 30  |helloworld PB150|
000000a0  35 31 30 38 37 00 50 72  65 73 73 20 61 6e 79 20  |51087.Press any |
000000b0  6b 65 79 20 74 6f 20 67  65 74 20 69 6e 74 6f 20  |key to get into |
000000c0  70 72 6f 74 65 63 74 20  6d 6f 64 65 2e 2e 2e 00  |protect mode....|
000000d0  4f 2e 4b 2e 21 00 57 65  20 61 72 65 20 6e 6f 77  |O.K.!.We are now|
000000e0  20 69 6e 20 50 52 4f 54  45 43 54 20 4d 4f 44 45  | in PROTECT MODE|
000000f0  21 00 00 00 00 00 00 00  00 00 ff ff 00 00 00 9a  |!...............|
00000100  c0 00 01 00 00 80 0b 92  c0 00 17 00 f2 7c 00 00  |.............|..|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000200
jqcustc@ubuntu:~/未命名文件夹 2$
```

```
jqcustc@ubuntu: ~/未命名文件夹 2
jqcustc@ubuntu:~/未命名文件夹 2$ hexdump -C myOS.bin
00000000  b8 b0 7f 00 00 89 c4 89  c5 b8 b0 7f 00 00 bb 50  |...............P|
00000010  8f 00 00 c7 00 00 00 00  00 83 c0 01 29 c3 75 ee  |............).u.|
00000020  e8 02 00 00 00 eb fe 55  89 e5 83 ec 10 c7 45 f8  |.......U......E.|
00000030  14 7f 00 00 c7 45 fc 44  7f 00 00 c7 45 f4 00 80  |.....E.D....E...|
00000040  0b 00 c7 45 f0 00 00 00  00 eb 12 8b 45 f4 8d 50  |...E........E..P|
00000050  02 89 55 f4 66 c7 00 00  00 83 45 f0 01 81 7d f0  |..U.f.....E...}.|
00000060  cf 07 00 00 7e e5 c7 45  f4 dc 85 0b 00 c7 45 f0  |....~..E......E.|
00000070  00 00 00 00 eb 12 8b 45  f4 8d 50 02 89 55 f4 66  |.......E..P..U.f|
00000080  c7 00 2a 07 83 45 f0 01  8b 55 f0 8b 45 fc 01 d0  |..*..E...U..E...|
00000090  0f b6 00 84 c0 75 df c7  45 f4 7c 86 0b 00 c7 45  |.....u..E.|....E|
000000a0  f0 00 00 00 00 eb 24 8b  45 f4 8d 50 02 89 55 f4  |......$.E..P..U.|
000000b0  8b 4d f0 8b 55 f8 01 ca  0f b6 12 66 0f be d2 66  |.M..U......f...f|
000000c0  81 c2 00 07 66 89 10 83  45 f0 01 8b 55 f0 8b 45  |....f...E...U..E|
000000d0  f8 01 d0 0f b6 00 84 c0  75 cd c7 45 f4 1c 87 0b  |........u..E....|
000000e0  00 c7 45 f0 00 00 00 00  eb 12 8b 45 f4 8d 50 02  |..E........E..P.|
000000f0  89 55 f4 66 c7 00 2a 07  83 45 f0 01 8b 55 f0 8b  |.U.f..*..E...U..|
00000100  45 fc 01 d0 0f b6 00 84  c0 75 df b8 00 00 00 00  |E........u......|
00000110  c9 c3 00 00 2a 20 20 20  20 20 20 20 20 20 20 20  |....*           |
00000120  20 20 20 20 68 65 6c 6c  6f 20 77 6f 72 6c 64 20  |    hello world |
00000130  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
00000140  2a 00 00 00 2a 2a 2a 2a  2a 2a 2a 2a 2a 2a 2a 2a  |*...************|
00000150  2a 2a 2a 2a 2a 2a 2a 2a  2a 2a 2a 2a 2a 2a 2a 2a  |****************|
*
00000170  2a 00 00 00 14 00 00 00  00 00 00 00 01 7a 52 00  |*............zR.|
00000180  01 7c 08 01 1b 0c 04 04  88 01 00 00 1c 00 00 00  |.|..............|
00000190  1c 00 00 00 93 fe ff ff  eb 00 00 00 00 41 0e 08  |.............A..|
000001a0  85 02 42 0d 05 02 e7 c5  0c 04 04 00              |..B.........|
000001ac
jqcustc@ubuntu:~/未命名文件夹 2$
```

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。

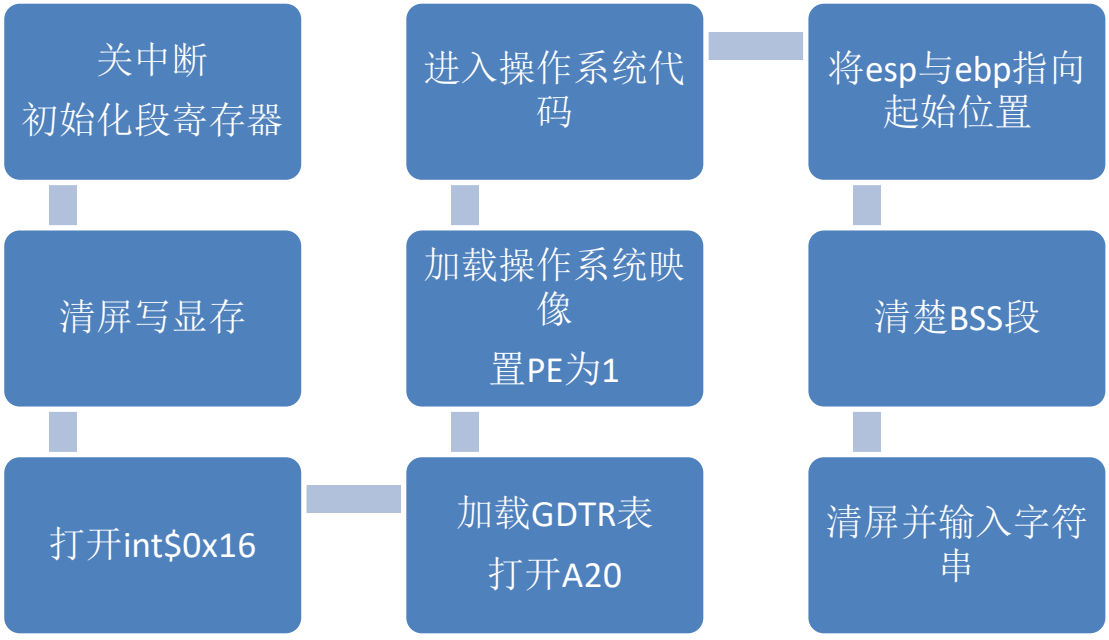
Makefile 语法规则：

```
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。