

## Phase 1

The screenshot shows the File Editor Neo application. The menu bar includes File, Edit, View, Select, Operations, Bookmarks, NTFS Streams, Tools, History, Window, and Help. The toolbar contains icons for file operations, navigation, and editing. The main window displays a hex editor for a file named 'helloos.img'. The hex dump shows the following data:

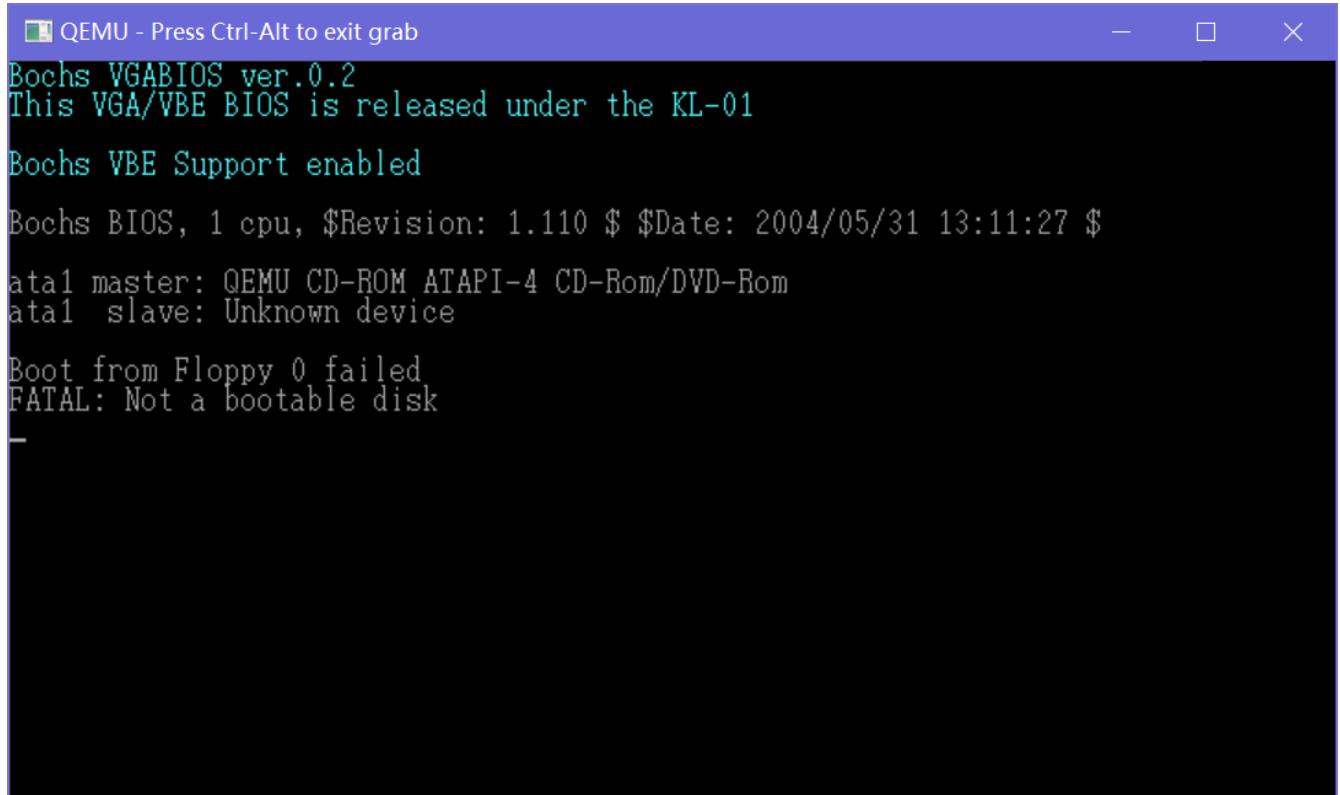
Address	Hex	ASCII
0000004d	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	
00000000	eb 4e 90 48 45 4c 4c 4f 49 50 4c 00 02 01 01 00	曉 志 ELLOIPL...
00000010	02 e0 00 40 0b f0 09 00 12 00 02 00 00 00 00 00	.?.@.?. . . . . .
00000020	40 0b 00 00 00 00 29 ff ff ff ff 48 45 4c 4c 4f	@. . . . .) HELLO
00000030	2d 4f 53 20 20 20 46 41 54 31 32 20 20 20 00 00	-OS FAT12 . .
0000004d	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000050	b8 00 00 8e d0 bc 00 7c 8e d8 8e c0 be 74 7c 8a	?..懶?. 序幫總!??
00000060	04 83 c6 01 3c 00 74 09 b4 0e bb 0f 00 cd 10 eb	煙 .<.t.? ? .? 肺
00000070	ee f4 eb fd 0a 0a 68 65 6c 6c 6f 2c 20 77 6f 72	暴? .hello, wor
00000080	6c 64 0a 00 00 00 00 00 00 00 00 00 00 00 00 00	ld. . . . . . . . .
00000090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
000000a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
000000b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
000000c0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
000000d0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
000000e0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
000000f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .
00000160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . .

The right sidebar contains the following panels:

- History:** Shows a list of recent files and folders.
- Volume Navigator:** Shows the current volume and its structure.
- Selection:** Shows the selected file and its properties.
- Information:** Shows the total size and number of fragments of the selected file.
- Details:** Shows the file's attributes and permissions.

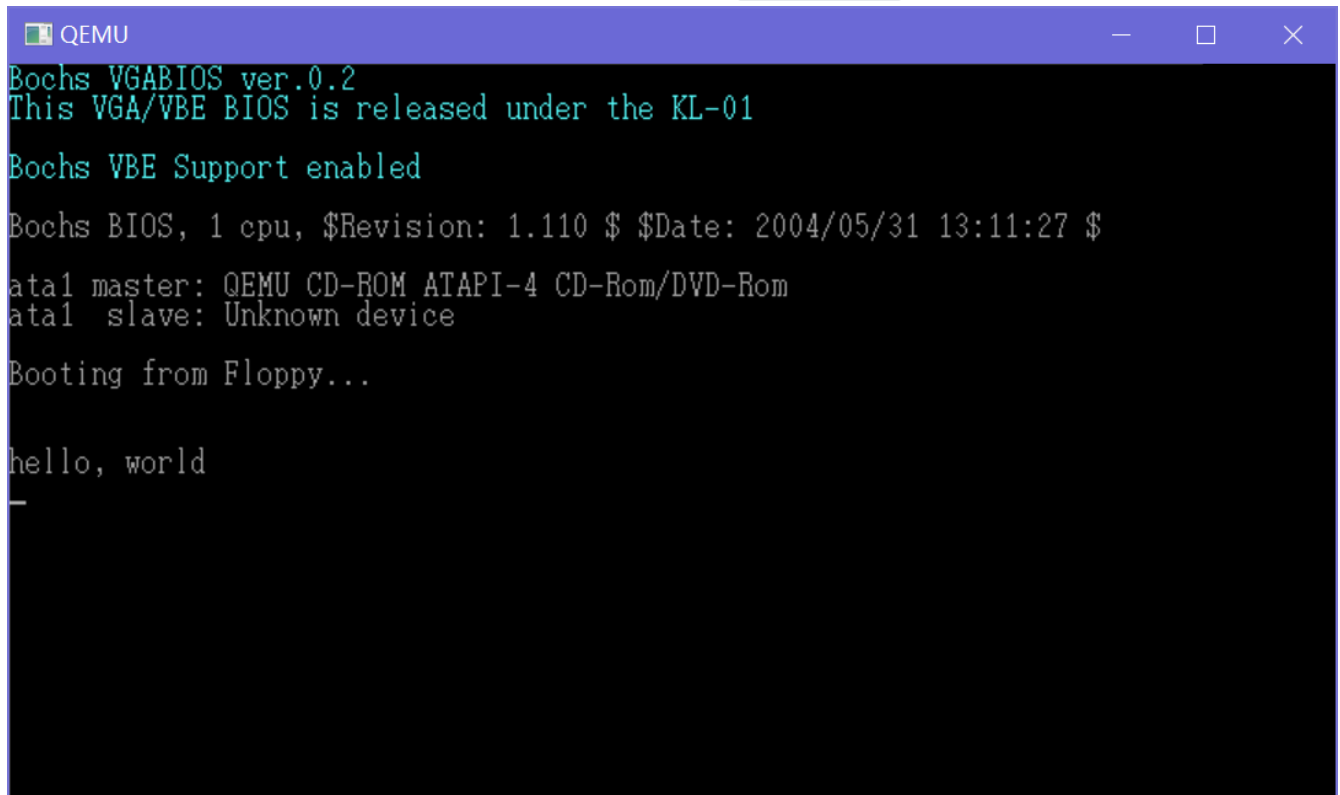
The status bar at the bottom displays the current offset (0x0000004d (77)), size (0x00168000 (1,474,560)), and file size (1.41 MB). It also shows the hex bytes, default ANSI OVI, and the file's name (helloos.img).

先按照书上的指导编制 `helloos.img`，然后解压光盘内容，尝试使用作者提供的模拟器载入这个镜像。先在 `tolset` 中建好文件夹并打开 `!cons_nt.bat`，运行 `run.bat`，提示 `FATAL: Not a bootable disk`



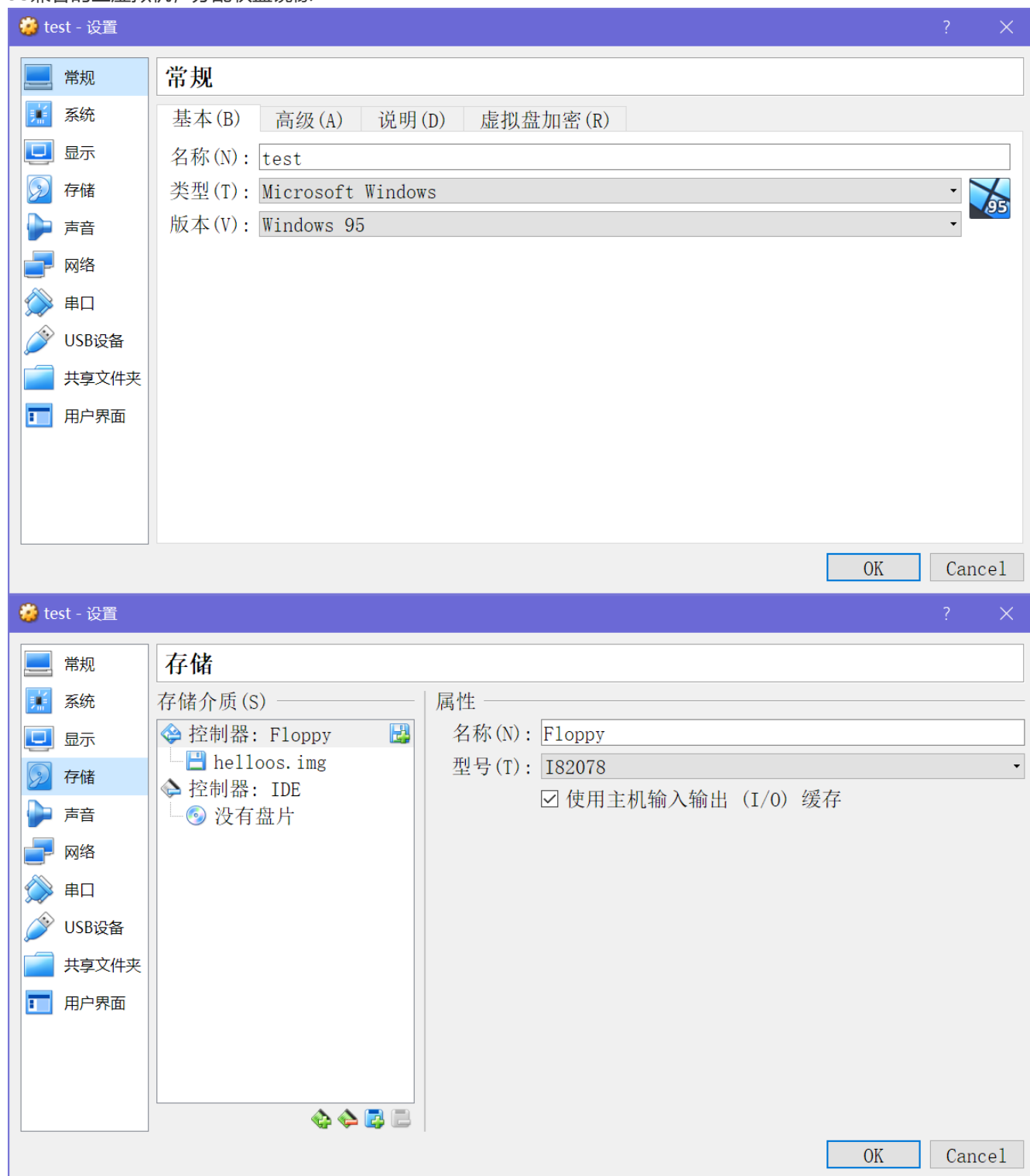
```
QEMU - Press Ctrl-Alt to exit grab
Bochs VGABIOS ver.0.2
This VGA/VBE BIOS is released under the KL-01
Bochs VBE Support enabled
Bochs BIOS, 1 cpu, $Revision: 1.110 $ $Date: 2004/05/31 13:11:27 $
ata1 master: QEMU CD-ROM ATAPI-4 CD-Rom/DVD-Rom
ata1 slave: Unknown device
Boot from Floppy 0 failed
FATAL: Not a bootable disk
```

咋回事呢？我多次对照书本检查，看起来无误。最后直接去看光盘里的 `helloos.img`，发现书上只写了前70行代码中的一部分，实际上并不是完整的代码。直接使用光盘里提供的 `helloos.img`，成功载入

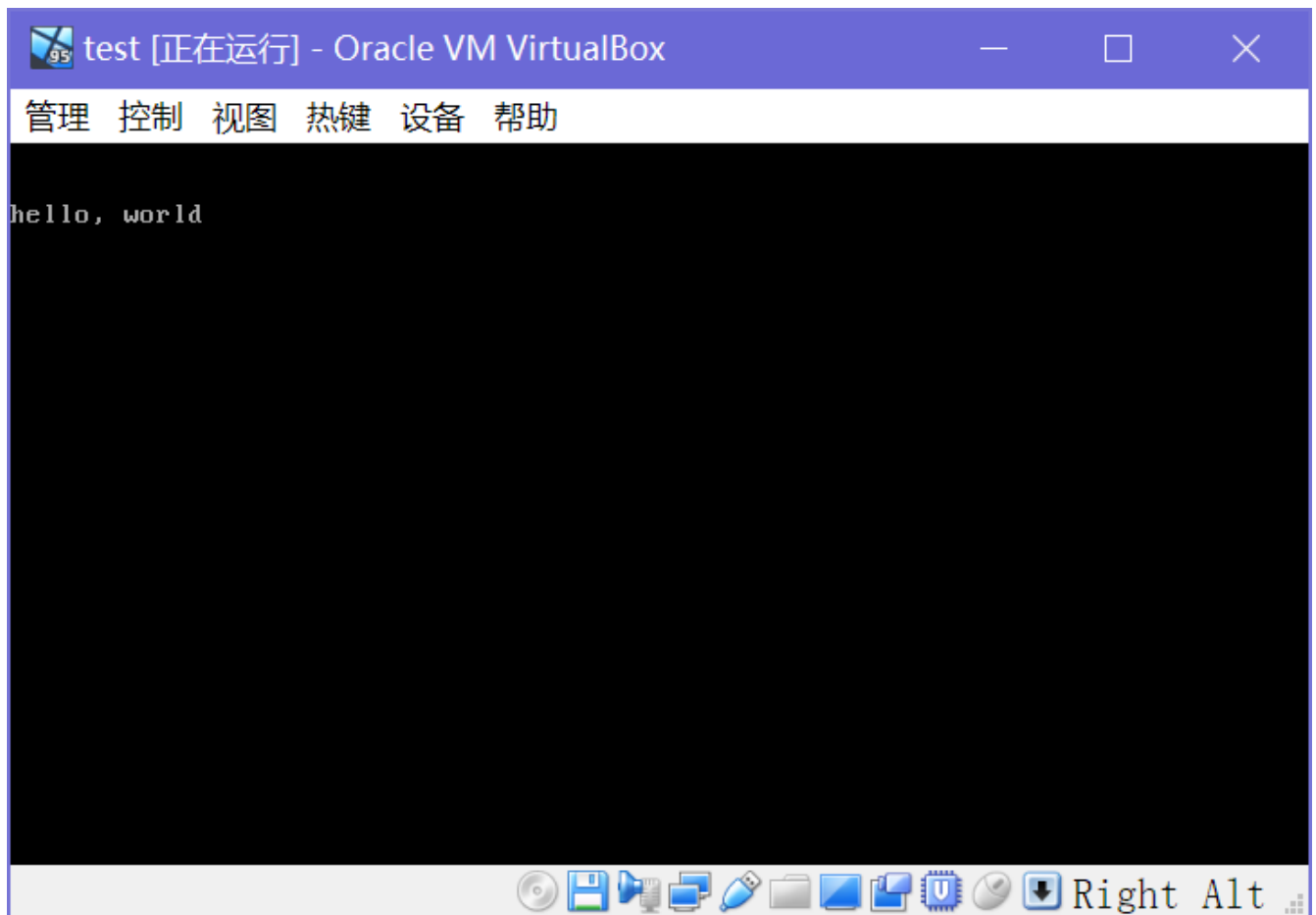


```
QEMU
Bochs VGABIOS ver.0.2
This VGA/VBE BIOS is released under the KL-01
Bochs VBE Support enabled
Bochs BIOS, 1 cpu, $Revision: 1.110 $ $Date: 2004/05/31 13:11:27 $
ata1 master: QEMU CD-ROM ATAPI-4 CD-Rom/DVD-Rom
ata1 slave: Unknown device
Booting from Floppy...
hello, world
```

除此之外，我还尝试了使用虚拟机程序 Oracle VM VirtualBox 直接载入 helloos.img。我新建了一个Windows 95兼容的空虚拟机，分配软盘镜像



点击启动，成功！



## Phase 2

### 无脑转成汇编

将 `helloos.img` 用汇编语言的 `DB` 指令无脑转写。`DB` 指令就是插入若干二进制数据，我们用其他语言可以写一个程序来无脑的完成 `helloos.img` 到 `nas` 的转换。

只需要读入字节并以16进制方式打印出来，中间加逗号，行首加 `DB<SPACE>` 即可。

### 初步缩小代码大小

使用 `RESB` 指令来替代连续的0

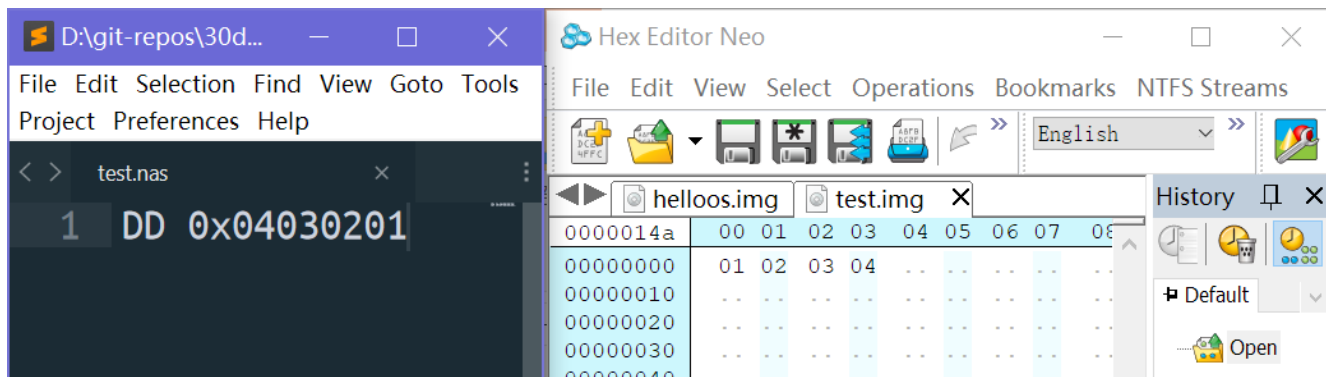
加判断，出现连续为0字节时计数，用 `RESB <字节数>` 替代

### 继续加工润色

我们来分析一下书上的代码

`DB` 中数据的顺序与 `helloos.img` 相同，为什么 `DW 512` 对应的是 `00 02` 呢？这是因为 `WORD` 是两个字节，而我们的模拟环境/英特尔的CPU使用的是小端序，低有效字节处于低地址。

以下代码验证了 `nask` 使用小端序



\$ 这个关键字可以减少我们修改代码之后所要进行的二次修改工作

## Phase 3

小小的总结一下：借用十六进制编辑器，我们可以制作任意一个文件，精确的控制文件中的每一个bit。而汇编语言，则提供给了我们一个更方便的用十六进制组成文件方法，减少我们直接用十六进制编辑器的工作量。