代码阅读

为什么`mov bp, ax`后，`int 10h`就能够取到BootMessage的地址了？

答案

因为这个指令之后，es:bp的值就是"Hello OS"字符串的首地址。

过程

16号中断用来触发一个关于视频服务的功能。当功能号是0x13时，打印字符串，es:bp是字符串地址。

`mov bp, ax`是把字符串首地址赋值给bp。

而es在函数调用之前被赋值成了CS就是段首地址。

这样，es:bp就是BootMessage所在的实际地址。16号中断里就可以这样找到实际的BootMessage地址了。

https://zh.wikipedia.org/wiki/INT_10H

运行到这行代码的时候ax里面的值是多少？

答案

0x7c1e

过程

由`ndisasm -o 0x7c00 boot.bin >> disboot.asm`得到反汇编代码`disboot.asm`。

在`./disboot.asm`中有如下代码：

```
00007C0B  B81E7C  mov ax,0x7c1e
00007C0E  89C5    mov bp,ax
```

可见ax的值是0x7c1e

http://chuanwang66.iteye.com/blog/1426351

这个值是不是BootMessage所在内存中的位置（即相对地址还是绝对地址）？

答案

在此例中ax的值是BootMessage所在内存中的位置。

过程

es是段首地址，bp是相对地址，两者结合是绝对地址（回答1）。

此例中，计算机刚启动，所有段寄存器都是0，所以bp的值就是其字符串的绝对地址。

如果es中的值不是0，则回答为否。