



# 课程实验报告

实验名称      简单引导程序的实现

课程名称      操作系统

院      系      计算机科学与技术系

学      号      191220129

姓      名      邢尚禹

邮      箱      191220129@smail.nju.edu.cn

实验日期      2021 年 3 月

# 目录

<b>1</b>	<b>实验进度</b>	<b>2</b>
<b>2</b>	<b>实验过程及结果</b>	<b>2</b>
2.1	在实模式下打印 hello world . . . . .	2
2.2	在保护模式下打印 hello world . . . . .	3
2.2.1	从实模式到保护模式的切换 . . . . .	3
2.2.2	在保护模式中打印 hello world . . . . .	4
2.3	在保护模式下装载程序 . . . . .	6
<b>3</b>	<b>困难与建议</b>	<b>6</b>

# 1 实验进度

已完成所有内容。

## 2 实验过程及结果

### 2.1 在实模式下打印 hello world

查阅相关资料知，通过 `int 0x10` 可以打印字符串，其中 (dl, dh) 为坐标，al, bl 为属性，bx 为页码，cx 为长度，ah 为 0x13，bp 为字符串地址。据此写汇编代码如下：

```
1  .code16
2
3  .global start
4  start:
5      movw %cs, %ax
6      movw %ax, %ds
7      movw %ax, %es
8      movw %ax, %ss
9
10     movw $0x7d00, %ax
11     movw %ax, %sp # setting stack pointer to 0x7d00
12
13     movw $message, %bp
14     movb $0x13, %ah
15     movb $1, %al
16     movw $14, %cx
17     movb $0, %dl
18     movb $0, %dh
19     movw $0x000f, %bx
20     int $0x10
21
22 loop:
23     jmp loop
24
25 message:
```

```
26 .string "Hello, World!\n"
```

运行结果:

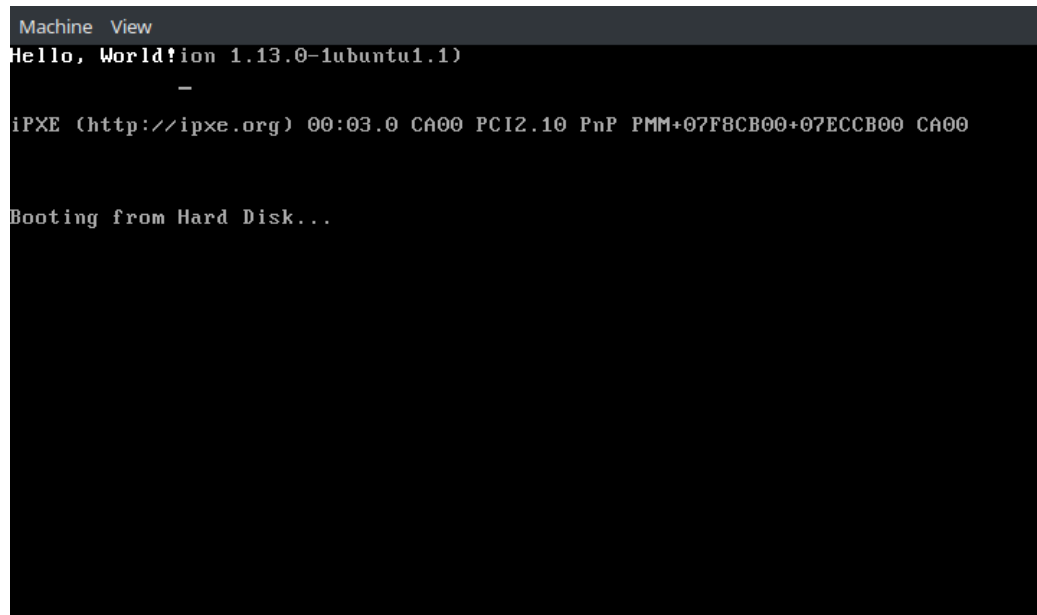


图 1: 实模式下的运行结果

## 2.2 在保护模式下打印 hello world

### 2.2.1 从实模式到保护模式的切换

关闭中断, 打开 A20 数据总线, 加载 GDTR, 设置 CR0 的 PE 位 (第 0 位) 为 1, 通过长跳转设置 CS 进入保护模式。

```
1 cli
2 inb $0x92, %al
3 orb $2, %al
4 outb %al, $0x92
5 data32 addr32 lgdt gdtDesc
6 data32 mov %cr0, %eax
7 orb $1, %al
8 data32 mov %eax, %cr0
9 data32 addr32 ljmp $0x08, $start32
```

```

10
11 .p2align 2
12 gdt:
13     # GDT definition here
14     .word 0, 0
15     .byte 0, 0, 0, 0
16
17     .word 0xffff, 0
18     .byte 0, 0x9a, 0xcf, 0
19
20     .word 0xffff, 0
21     .byte 0, 0x92, 0xcf, 0
22
23     .word 0xffff, 0x8000
24     .byte 0xb, 0x92, 0xcf, 0
25
26 gdtDesc:
27     # gdtDesc definition here
28     .word (gdtDesc - gdt - 1)
29     .long gdt

```

### 2.2.2 在保护模式中打印 hello world

只需要把字符串（带字符属性信息）写入显存 0xb8000 处即可。为输出白色字符，可将属性信息设置为字节 0x0f。

```

1     # for(int i=0; i<24; ++i)      *(0xb8000 + i) = message
2         [i];
3
4     movl $0, %eax
5
6 .L1:
7     movl %eax, %ebx
8     shl $2, %ebx
9     movl $message, %ecx
10    addl %ebx, %ecx
11    addl $0xb8000, %ebx
12    movl (%ecx), %edx
13    movl %edx, (%ebx)
14    incl %eax

```

```

12         cml $7, %eax
13         jb .L1
14
15 loop32:
16         jmp loop32
17
18 message:
19         # Hello, World!\n\0
20         .byte 0x48, 0x0f, 0x65, 0x0f, 0x6c, 0x0f, 0x6c, 0x0f, 0
           x6f, 0x0f, 0x2c, 0x0f, 0x20, 0x0f, 0x57, 0x0f, 0x6f,
           0x0f, 0x72, 0x0f, 0x6c, 0x0f, 0x64, 0x0f, 0x21, 0x0f,
           0x00, 0x0f

```

```

Machine View
Hello, World! on 1.13.0-1ubuntu1.1)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8CB00+07ECCB00 CA00

Booting from Hard Disk...

```

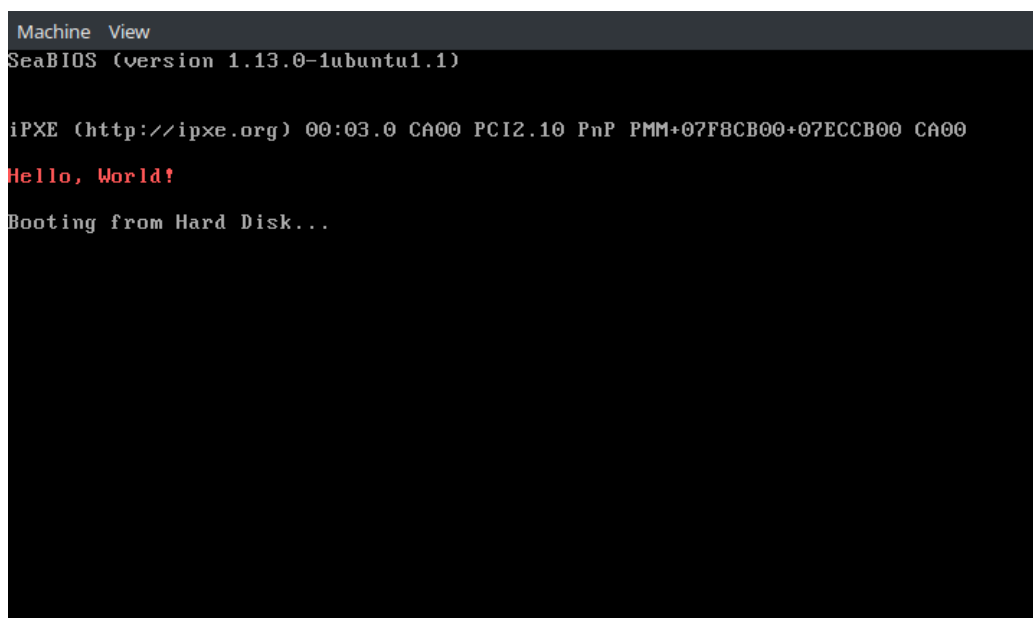
图 2: 保护模式下的运行结果

## 2.3 在保护模式下装载程序

调用 readSect 函数装载程序，再跳转执行即可。

```
1 void bootMain(void)
2 {
3     void (*program)(void) = 0x8c00;
4     readSect(program, 1);
5     program();
6 }
```

测试结果：



```
Machine View
SeaBIOS (version 1.13.0-1ubuntu1.1)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8CB00+07ECCB00 CA00
Hello, World!
Booting from Hard Disk...
```

图 3: 程序装载的运行结果

## 3 困难与建议

实验指导文档总体设计不太合理。例如，从实模式到保护模式的切换部分，大量篇幅在介绍 GDTR 等内容，但这些已经在 ics 课上学过，没有太大必要；但是 A20 数据总线的内容并没有详细介绍，需要自己查阅资料，比较耗费时间。另外，建议增加一些对汇编语言的介绍，例如如何定义字符

串，如何设置全局变量等。`.word`，`.byte` 这些内容都没有学过，加一些简单介绍会更好。