Exercise 报告

515030910252 黄柏翰

Exercise 3.

At what point does the processor start executing 32-bit code? What exactly causes the switch from 16- to 32-bit mode?

```
# Switch from real to protected mode, using a bootstrap GDT
# and segment translation that makes virtual addresses
# identical to their physical addresses, so that the
# effective memory map does not change during the switch.
lgdt    gdtdesc
movl    %cr0, %eax
orl     $CR0_PE_ON, %eax
movl    %eax, %cr0

# Jump to next instruction, but in 32-bit code segment.
# Switches processor into 32-bit mode.
ljmp    $PROT_MODE_CSEG, $protcseg

.code32                     # Assemble for 32-bit mode
protcseg:
# Set up the protected-mode data segment registers
movw    $PROT_MODE_DSEG, %ax    # Our data segment selector
movw    %ax, %ds                # -> DS: Data Segment
movw    %ax, %es                # -> ES: Extra Segment
movw    %ax, %fs                # -> FS
movw    %ax, %gs                # -> GS
movw    %ax, %ss                # -> SS: Stack Segment
```

•What is the *last* instruction of the boot loader executed, and what is the *first* instruction of the kernel it just loaded?

```
    // load each program segment (ignores ph flags)
    ph = (struct Proghdr *) ((uint8_t *) ELFHDR + ELFHDR->e_phoff);
    eph = ph + ELFHDR->e_phnum;
    for (; ph < eph; ph++)
7d66:       83 c4 0c                add    $0xc,%esp
7d69:       eb e6                   jmp    7d51 <bootmain+0x3c>
            // as the physical address)
            readseg(ph->p_pa, ph->p_memsz, ph->p_offset);

    // call the entry point from the ELF header
    // note: does not return!
    ((void (*)(void)) (ELFHDR->e_entry))();
7d6b:       ff 15 18 00 01 00       call   *0x10018
```

•How does the boot loader decide how many sectors it must read in order to fetch the entire kernel from disk? Where does it find this information?

```
for (; ph < eph; ph++)
```

Be able to answer the following questions:

1.console 是最终一个一个输出到终端，printf 对字符串进行处里分段

```
        outb(0x378+0, c);
        outb(0x378+2, 0x08|0x04|0x01);
        outb(0x378+2, 0x08);
```

2.

```
/***** Text-
mode CGA/VGA
display output *****/
```

保证缓冲区中的最后显示出去的内容的大小不要超过显示的大小界限 CRT_SIZE

3.fmt points to string, ap points to others

4.He110 World 因 57616->0xe110->rld\0

5.The stack at 3+4bytes is not a specific value

6.cprintf reads backwards

EX11:

```
1.movl    $0x0,%ebp             # nuke frame pointer
   movl    $(bootstacktop),%esp
```

2.为 0xf0110000

3.提高空间地址，让操作系统在高地址处

4.bootstacktop

EX12:

2,(ebp,ebx)//返回地址、参数

EX14:

```
obj/kern/kernel:     file format elf32-i386

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00001cf1  f0100000  00100000  00001000  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .rodata       00000868  f0101d00  00101d00  00002d00  2**5
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .stab         00003f49  f0102568  00102568  00003568  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .stabstr      00001a8e  f01064b1  001064b1  000074b1  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data         0000a304  f0108000  00108000  00009000  2**12
                  CONTENTS, ALLOC, LOAD, DATA
  5 .bss          00000660  f0112320  00112320  00013304  2**5
                  ALLOC
  6 .comment      00000034  00000000  00000000  00013304  2**0
                  CONTENTS, READONLY
```

```
oslab@ubuntu:~/oslabs/jos-2018-spring$ objdump -G obj/kern/kernel

obj/kern/kernel:     file format elf32-i386

Contents of .stab section:

Symnum n_type n_othr n_desc n_value  n_strx String

-1     HdrSym 0      1349   00001a8d 1
0      SO     0      0      f0100000 1      {standard input}
1      SOL    0      0      f010000c 18     kern/entry.S
2      SLINE  0      44     f010000c 0
3      SLINE  0      56     f0100015 0
4      SLINE  0      57     f010001a 0
5      SLINE  0      59     f010001d 0
6      SLINE  0      60     f0100020 0
7      SLINE  0      61     f0100025 0
8      SLINE  0      66     f0100028 0
9      SLINE  0      67     f010002d 0
10     SLINE  0      73     f010002f 0
11     SLINE  0      76     f0100034 0
12     SLINE  0      79     f0100039 0
13     SLINE  0      82     f010003e 0
14     SO     0      2      f0100040 31     kern/entrypgdir.c
15     OPT    0      0      00000000 49     gcc2_compiled.
16     LSYM   0      0      00000000 64     int:t(0,1)=r(0,1);-2147483648;2147483647;
17     LSYM   0      0      00000000 106    char:t(0,2)=r(0,2);0;127;
18     LSYM   0      0      00000000 132    long int:t(0,3)=r(0,3);-2147483648;2147483647;
```

```
        .file   "init.c"
        .stabs  "kern/init.c",100,0,2,.Ltext0
        .text
.Ltext0:
        .stabs  "gcc2_compiled.",60,0,0,0
        .stabs  "int:t(0,1)=r(0,1);-2147483648;2147483647;",128,0,0,0
        .stabs  "char:t(0,2)=r(0,2);0;127;",128,0,0,0
        .stabs  "long int:t(0,3)=r(0,3);-0;4294967295;",128,0,0,0
        .stabs  "unsigned int:t(0,4)=r(0,4);0;4294967295;",128,0,0,0
        .stabs  "long unsigned int:t(0,5)=r(0,5);0;-1;",128,0,0,0
        .stabs  "__int128:t(0,6)=r(0,6);0;-1;",128,0,0,0
        .stabs  "__int128 unsigned:t(0,7)=r(0,7);0;-1;",128,0,0,0
        .stabs  "long long int:t(0,8)=r(0,8);-0;4294967295;",128,0,0,0
        .stabs  "long long unsigned int:t(0,9)=r(0,9);0;-1;",128,0,0,0
        .stabs  "short int:t(0,10)=r(0,10);-32768;32767;",128,0,0,0
        .stabs  "short unsigned int:t(0,11)=r(0,11);0;65535;",128,0,0,0
        .stabs  "signed char:t(0,12)=r(0,12);-128;127;",128,0,0,0
        .stabs  "unsigned char:t(0,13)=r(0,13);0;255;",128,0,0,0
        .stabs  "float:t(0,14)=r(0,1);4;0;",128,0,0,0
        .stabs  "double:t(0,15)=r(0,1);8;0;",128,0,0,0
        .stabs  "long double:t(0,16)=r(0,1);16;0;",128,0,0,0
        .stabs  "_Decimal32:t(0,17)=r(0,1);4;0;",128,0,0,0
        .stabs  "_Decimal64:t(0,18)=r(0,1);8;0;",128,0,0,0
        .stabs  "_Decimal128:t(0,19)=r(0,1);16;0;",128,0,0,0
        .stabs  "void:t(0,20)=(0,20)",128,0,0,0
        .stabs  "./inc/stdio.h",130,0,0,0
        .stabs  "./inc/stdarg.h",130,0,0,0
        .stabs  "va_list:t(2,1)=(2,2)=ar(2,3)=r(2,3);0;-1;;0;0;(2,4)=xs__va_list_tag:",128,0,0,0
        .stabn  162,0,0,0
        .stabn  162,0,0,0
        .stabs  "./inc/string.h",130,0,0,0
        .stabs  "./inc/types.h",130,0,0,0
        .stabs  "bool:t(4,1)=(0,1)",128,0,0,0
        .stabs  "int8_t:t(4,2)=(0,12)",128,0,0,0
        .stabs  "uint8_t:t(4,3)=(0,13)",128,0,0,0
        .stabs  "int16_t:t(4,4)=(0,10)",128,0,0,0
        .stabs  "uint16_t:t(4,5)=(0,11)",128,0,0,0
```