Document - Lab 1

516030910422 赵樱

Exercise 2

```
0x7c00 : 进入boot loader
0x7d71 : 进入kernel
```

Exercise 5.

分别检测0x00100000处8个words，在进入boot loader时全为0，进入kernel时被映射后填入了指令。

```
(gdb) x/8x 0x00100000
0x100000:   0x1badb002  0x00000000  0xe4524ffe  0x7205c766
0x100010:   0x34000004  0x3000b812  0x220f0011  0xc0200fd8
(gdb) x/8i 0x00100000
   0x100000:    add    0x1bad(%eax),%dh
   0x100006:    add    %al,(%eax)
   0x100008:    decb   0x52(%edi)
   0x10000b:    in     $0x66,%al
   0x10000d:    movl   $0xb81234,0x472
   0x100017:    xor    %dl,(%ecx)
   0x100019:    add    %cl,(%edi)
   0x10001b:    and    %al,%bl
```

Exercise 7

在置cr0后建立了页表和new mapping。之后的第一条指令是```jmp *%eax```，如果映射未刷新会jmp到错误的地方。

Formatted Printing to the Console

Exercise 8：Octal

阅读十进制、十六进制代码后，修改printfmt.c中reswitch case 'o'： Putch 一个'0'，读取uint参数，将base置为8后goto number。

Exercise 9: '+'

- 在vprintfmt函数中增加一个变量sign，标记当前是否要显示加减号。
- 当从字符串中switch到'+'号时将sign置为1。
- 同时在每个数字类型中增加判断：若当前sign等于1，则putch '+'/'-'。

Exercise 10：'%n'

- case'n'中，利用va_arg()函数获取参数的指针

- NULL：若参数为空指针，输出null_error返回
- overflow: putdat指针指向当前输出的位置，若putdat包含的值大于127(signed char的最大值)，给参数赋值-1并输出overflow_error
- normal：将putdat所指的值赋给参数指针所指的值。

Exercise 11：Padding

- 由于需要在数字右侧输出空格对齐，考虑将打印数字和打印Padding的逻辑分开，增加了辅助函数printnum_only，先输出非空格的左侧Padding → 通过除法计算数字长度num_len, 调用printnum_only递归输出数字 → 计算空格长，输出。

The Stack

Exercise 12

```
movl $0x0, %ebp
movl $(bootstacktop),%esp

  # Set the stack pointer
f0100034:  bc 00 10 11 f0   mov  $0xf0111000, %esp
```

- entry.S中的初始化%ebp %esp时初始化栈，查看相应反汇编可知 **%esp**的精确位置是0xf0111000，%ebp的初始值是0.

Exercise 13

- test_backtrace()的地址：0xf0100040

  0xf0100040 <test_backtrace>: push %ebp

  push到栈上的是ret addr 、ebp、参数5 4 3 2 1

Exercise 14

- 栈帧中，新ebp指向的是调用者ebp，再上面是ret addr和所有参数，所以参数可以通过ebp+index进行索引，参数空间可以通过新旧ebp相减再减去返回地址长度得到，通过ebp = *ebp可以跳回上层调用的函数，直到ebp为0（第一个函数处）。
- ebp可以通过调用read_ebp()读到，eip = *(ebp+4), arg通过ebp+8, ebp+12 ...读取

Exercise 15

- 添加指令：在command数组中增加条目，使其调用mon_backtrace函数
- debuginfo_eip：用N_SLINE作为参数调用stab_binsearch查找line。
- 在mon_backtrace的循环中调用debuginfo_eip, 格式化输出，其中**函数名**的输出需要用%.s，传入namelen参数调整精度。

Exercise 16

- 在start_overflow中修改pret addr为do_overflow的入口地址，使得函数返回时跳到do_overflow，但同时也要修改ret addr为overflow_me后的位置，正常返回。
- 使用cprintf和%n，由于有127的溢出限制，考虑目前内核中地址从0xf0100000开始，只要填充后2-3位字节，将后3个字节控制在0x7f、并且控制整体代码长度来实现。

Exercise 17

- 同理，增加time指令，调用mon_time函数
- 在mon_time中调用runcmd执行mon_kerninfo（为了之前的overflow，可以不调用函数，而是复制有用的代码插入到当前函数里，尽可能缩短代码量），以`asm volatile("rdtsc" : "=A" (time))`读取前后的时间相减的到cycle数。