Jingcheng Lu

42326170

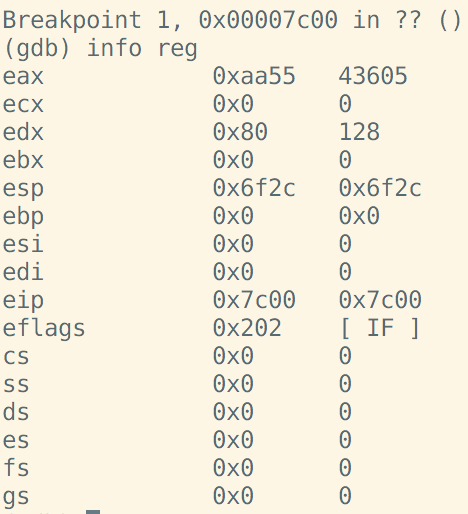
CS 238P: HW2

E1Q1: What is on the stack:

Write a short (3-5 word) comment next to each zero and non-zero value of the printout explaining what it is. Which part of the printout is actually the stack? (Hint: not all of it.)

Answer:

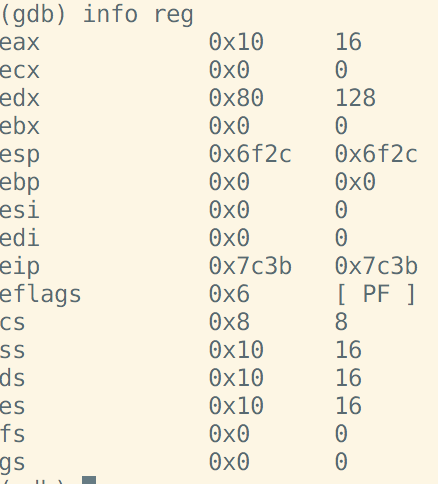
At the beginning the SS (stack segment) is 0, the system is at real mode. The esp point to 0x6f2c.



Then after code

0x7c3b: mov $0x0,%ax

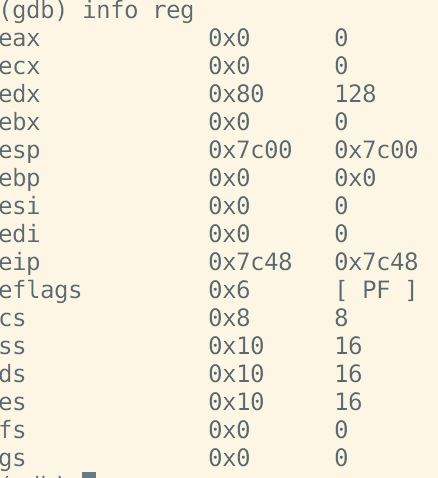
The ss is not zero anymore, enter the protection mode.



After code

mov $0x7c00, %esp

The esp pointed to 0x7c00, which is the top of stack.



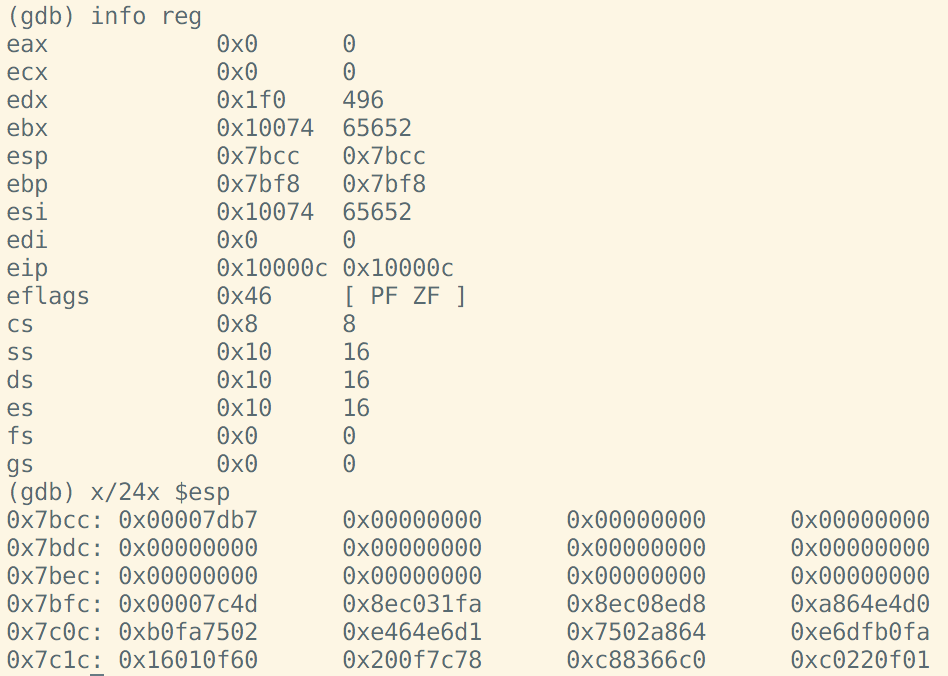
Then it calls bootmain, which start at 7d2f



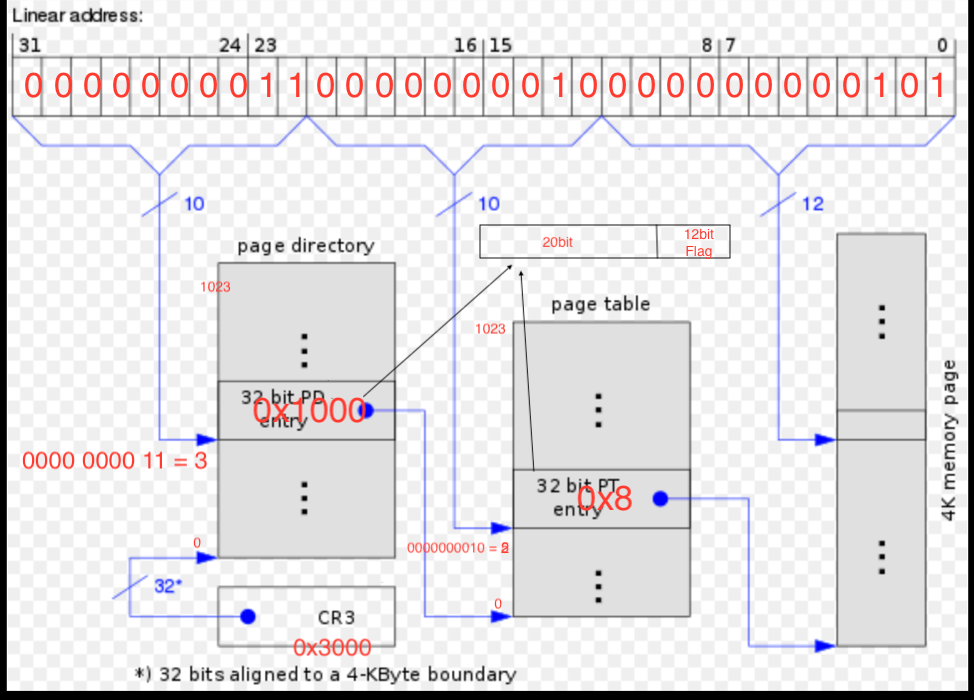
The return address is 7c4d.

The in the bootmain fuction, it loads kernel and jump to the start execute kernel which start at 0x10000 without return.

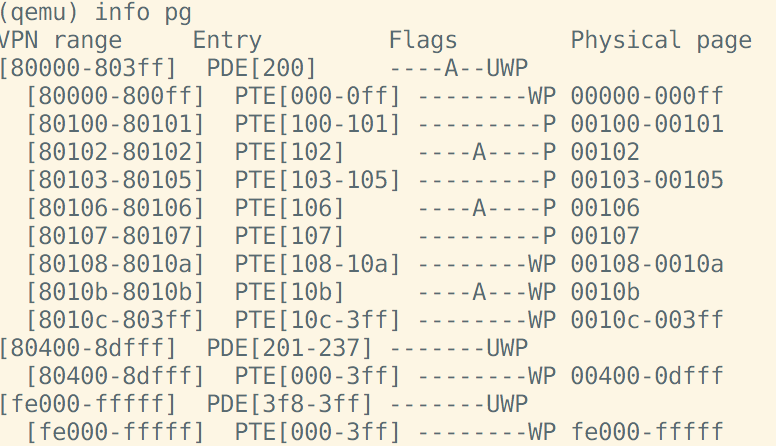
The stack is from 0x7c00 to 0x7bcc, the rest shown on this picture is bootloader.



E2Q1 Explain how virtual to physical address translation works



E3Q1 What is the state of page tables after xv6 is done initializing the first 4K page table?



Answer:

In this picture, we can see 3 pages table, the first one begins at 80000 and ends at 803ff. 3ff = 1111 1111 11 which include 1024 spaces. And its flags are “----A—UMP”, means that the page can be accessed, wrote, run in user mode and is present (If P is not set then is page fault). The entry is 0x200, and in this page there are 9 PTE. And the Physical page column shows that It maps Virtual address [80000-803ff] to Physical address[00000-003ff]. The second and third one is same with the first one. With this we can get the details of page numbers, page entries, page flags as well as their relationship between physical and virtual address.