Hien Duong
11587750
CPTS 460

Lab 1 Questions and Answers

1. On Intel x86 based PCs,during booting, what does BIOS do?

BIOS performs POST (Power-on Self Test) to check to see if the system hardware is ready for proper operation. Then it will search for a system to boot. The usual booting order is A drive (floppy disk), B drive (hard disk), cdrom, etc. The booting order can be changed by programming the BIOS. If BIOS finds a bootable device it will try to boot from that device, else there will be an error they will require user intervention.

How many sectors does BIOS load from the boot device?

BIOS only loads one sector called the boot sector. A floppy disk only has ONE boot sector, while a hard disk is divided into several partitions and each has its own boot sector. The very first sector on a hard disk is called the master boot record (MBR).

Where in memory does BIOS load the booter?

BIOS loads the first 512 bytes of the booter to memory location
(segment,offset) = (0x0000,0x7C00)

2. BIOS loads only 512 bytes of a booter into memory, which is only the beginning part of the booter. How does the booter load the remaining part of the booter into memory?

It calls BIOS int13 to load boot block to (ES,BX) = (0x9000,0). Load B0=|S0 S1| to (0x9000, 0). After loading the booter to the new segment it makes a jump. Causing the CPU to jump to (0x9000, start) to continue execution. Set CS, DS, SS for CPU all point at 0x9000. Call the main() in C, which is the boot program, after main(), if successful, jump to (0x1000, 0) to execute kernel.

3. Assume a COMPLETE booter is loaded at the segment 0x9000. WHY do we have to set the CPU's segment registers CS,DS,SS,ES to 0x9000?

Because of the one-segment memory model which says that CS=DS=SS = loaded segment, in this case the booter is loaded at the segment 0x9000 so CS,DS,SS,ES are set to the loaded segment 0x9000.

4. How do you find the file /boot/mtx?

We first have to read in the 0th group descriptor to find the start block of the inodes table. We will then read in the root inode, which is the number 2 inode in the inodes table. With the root inode’s data block we can search /boot and get it’s inode number we can use mailman’s algorithm to convert the inode number to the disk block containing the inode and its offset in that block. Now we read in the inode of boot and search for /mtx. Once we have found /boot/mtx we can load the image by loading it’s data blocks.

5. How to load the (disk) blocks of /boot/mtx to the segment 0x1000?
After finding the file /boot/mtx and obtained its inode we call setes(0x1000) to set ES to the loading segment (0x1000). We then load the direct blocks and indirect blocks into segment 0x1000.