CS460 Lab #1 Questions

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

On the initial power on, the BIOS initializes itself and checks for attached memory and other devices. The BIOS then looks for a system to boot. If a system is found, the BIOS will load the first sector of that device into specific registers and let CPU execute that piece of code.

When booting from a hard disk, the BIOS loads the MBR to (0x0000, 0x7C00) and executes the loaded code.

**How many sectors does BIOS load from the boot device?**

The BIOS loads only the first sector of that device into memory. This sector is 512 bytes.

**Where in memory does BIOS load the booter?**

The BIOS loads the booter into (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?

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?

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

The file /boot/mtx is found by traversing the data in each DIR starting from the root INODE. Each block of the root INODE is loaded into memory and each DIR entry is searched until an entry with the name boot is found. If boot is found, the INODE where boot is located is loaded into memory. The same process is then repeated but this time searching for a DIR entry with a name that matches mtx.

**5. How to load the (disk) blocks of /boot/mtx to the segment 0x1000?**

Once the INODE of /boot/mtx has been found, the register ES must be set to the location 0x1000. One ES is set to that location, each of the 12 direct blocks of the INODE can then be loaded into program memory. This is accomplished by getting a specific i\_block, loading it into the register, and then incrementing the register. Once the all 12 direct blocks of the INODE have been loaded into memory, the indirect blocks can be loaded into memory if any exists. This is done in much the same way. An indirect block is loaded into the ES register, the register is incremented to the next memory address and the next block is grabbed from the INODE.