

## Assignment-5

**Shashank Dargar** 2019107

### Steps to boot:

Compile the .asm file using nasm compiler to generate a binary .bin file which can be further used in booting with the qemu emulator. The command for the same is:

```
nasm boot.asm -f bin -o boot.bin
```

Open the binary file of the bootloader using the qemu emulator. The command that can be used is:

```
qemu-system-x86_64 boot.bin
```

### Implementation Details:

To make a bootloader using nasm and print the contents of 'cr0' along with "Hello world!", we performed the following in this order:

- Starting the bootloader in 16 bit Real mode.
  - Bits 16 tells the nasm to use real mode.
  - 'Org' command is used to specify the location used to load the boot file
  - Loading the pointer to the GDT table that we made in the boot.asm file.
  - Making the first bit of CR0 as 1, which represents the Protected Mode.
- Enabling GDT to enable protected mode.
  - The GDT is in the asm file
  - It represents the memory bounds for different segments of data.
- Switching to 32 bit protected mode.
  - By making 1st bit of CR0 as 1.
- Printing Hello World using the VGA buffer
  - Adding destination of VGA buffer to edi.
  - Making EAX = 0, to make screen black
  - Storing pointer to hello-world in ESI.
  - Running the loop to print.
- Printing contents of CR0 register bit by bit.
  - Looping over each of the 32- bits stored in CR0, and then printing it.
  - We fixed the counter to 31 (for 0 to 31 bits) and then inside it we ran another loop with repeated right shift operator to fetch the ith bit in each iteration and then printing 0 or 1 for it.