Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Lab 2:

Boot Loader – to load a particular OS. OS Image – code to access from BIOS to loading the OS

Programme	:	**BTech. CSE Core**	Semester	:	**Win 2021-22**
Course	:	**Operating Systems**	Code	:	**CSE2005**
Faculty	:	**Dr. Shyamala L**	Slot	:	**L25+L26**
Name	:	**Hariket Sukesh Kumar Sheth**	Register No.	:	**20BCE1975**

LAB 2

Aim: To write a boot loader – to load a particular OS. OS image – code to access from BIOS to loading the OS.

Steps:

Step 1: Installation of NASM and QEMU

```
hariketsheth@ubuntu: ~
hariketsheth@ubuntu:~$ sudo apt install nasm
[sudo] password for hariketsheth:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nasm
0 upgraded, 1 newly installed, 0 to remove and 113 not upgraded.
Need to get 375 kB of archives.
After this operation, 3,345 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu impish/universe amd64 nasm amd64 2.15.05-1 [375 kB]
Fetched 375 kB in 4s (95.1 kB/s)
Selecting previously unselected package nasm.
(Reading database ... 161471 files and directories currently installed.)
Preparing to unpack .../nasm_2.15.05-1_amd64.deb ...
Unpacking nasm (2.15.05-1) ...
Setting up nasm (2.15.05-1) ...
Processing triggers for man-db (2.9.4-2) ...
hariketsheth@ubuntu:~$
hariketsheth@ubuntu:~$
```

```
hariketsheth@ubuntu:~$ sudo apt install qemu
[sudo] password for hariketsheth:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  qemu
0 upgraded, 1 newly installed, 0 to remove and 58 not upgraded.
Need to get 16.0 kB of archives.
After this operation, 134 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu impish-updates/universe amd64 qemu amd64 1:6.0+dfsg-2expubuntu1.1 [16.0 kB]
Fetched 16.0 kB in 1s (25.1 kB/s)
Selecting previously unselected package qemu.
(Reading database ... 161501 files and directories currently installed.)
Preparing to unpack .../qemu_1%3a6.0+dfsg-2expubuntu1.1_amd64.deb ...
Unpacking qemu (1:6.0+dfsg-2expubuntu1.1) ...
Setting up qemu (1:6.0+dfsg-2expubuntu1.1) ...
hariketsheth@ubuntu:~$
hariketsheth@ubuntu:~$
```

Step 2: Create bootloader1.asm and compile after writing the code. Run the same using emulator

```
hariketsheth@ubuntu: ~/Desktop
hariketsheth@ubuntu:~/Desktop$ touch bootloader1.asm
hariketsheth@ubuntu:~/Desktop$ nano bootloader1.asm
hariketsheth@ubuntu:~/Desktop$ cat bootloader1.asm
[BITS 16]                       ;tell the assembler that its a 16 bit code
[ORG 0x7C00]                    ;Origin, tell the assembler that where the code will
                                ;be in memory after it is been loaded
JMP $                           ;infinite loop
TIMES 510 - ($ - $$) db 0       ;fill the rest of sector with 0
DW 0xAA55                       ; add boot signature at the end of bootloader
hariketsheth@ubuntu:~/Desktop$
hariketsheth@ubuntu:~/Desktop$
```

nasm bootloader1.asm -f bin -o boot.bin
qemu-system-x86_64 -drive file=boot.bin,index=0,media=disk,format=raw

Output:

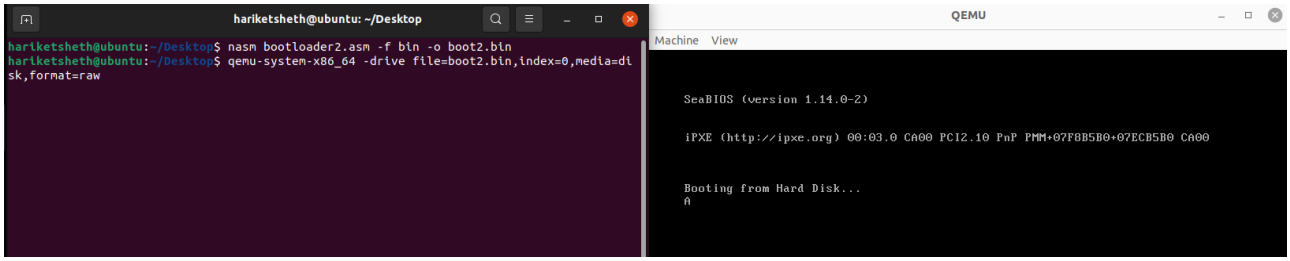
```
hariketsheth@ubuntu: ~/Desktop
hariketsheth@ubuntu:~/Desktop$ nasm /home/hariketsheth/Desktop/bootloader1.asm
  -f bin -o boot.bin
hariketsheth@ubuntu:~/Desktop$ qemu-system-x86_64 -drive file=boot.bin,index=0
,media=disk,format=raw
```

```
QEMU
Machine View

SeaBIOS (version 1.14.0-2)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8B5B0+07ECB5B0 CA00

Booting from Hard Disk...
```

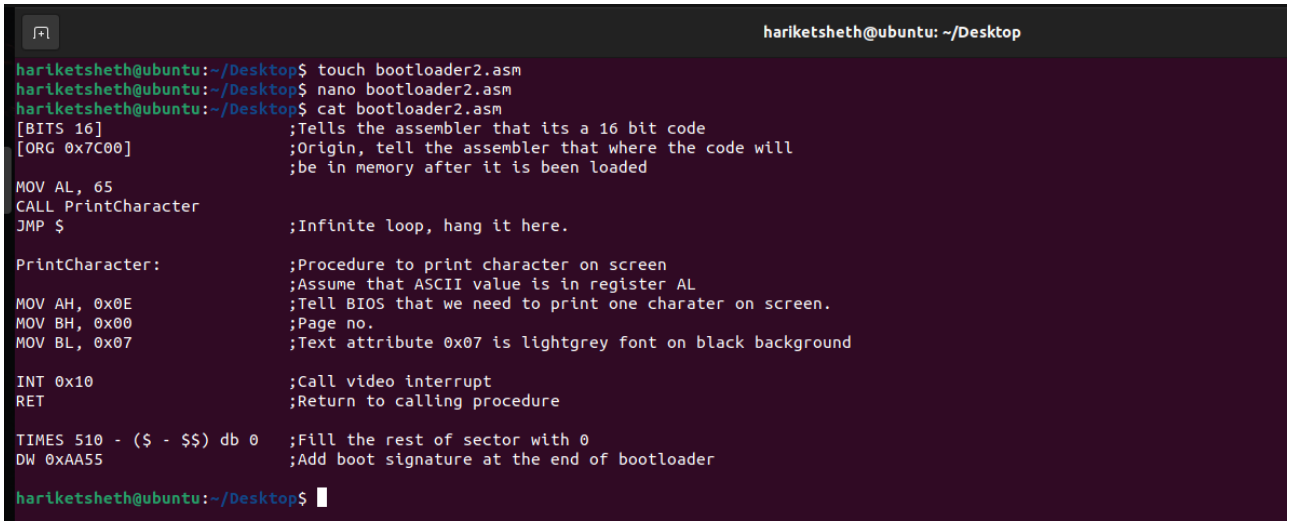
Step 3: Create 2nd Bootloader that prints 'A' on the screen

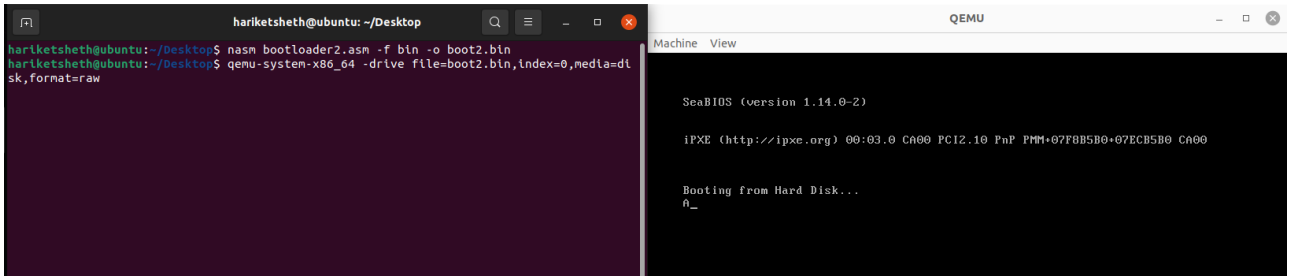
```
hariketsheth@ubuntu: ~/Desktop
hariketsheth@ubuntu:~/Desktop$ touch bootloader2.asm
hariketsheth@ubuntu:~/Desktop$ nano bootloader2.asm
hariketsheth@ubuntu:~/Desktop$ cat bootloader2.asm
bits 16
org 0x7c00
boot:
        mov si,hello
        mov ah, 0x0e
.loop:
        lodsb
        or al,al
        jz halt
        int 0x10
        jmp .loop
halt:
        cli
        hlt
hello: db "A",0
times 510 - ($-$$) db 0
dw 0xaa55
hariketsheth@ubuntu:~/Desktop$
```

nasm bootloader2.asm -f bin -o boot2.bin
qemu-system-x86_64 -drive file=boot2.bin,index=0,media=disk,format=raw



ANOTHER WAY OF BOOTLOADER





Step 4: Create 3rd Bootloader that prints name “Hariket”

```
hariketsheth@ubuntu: ~/Desktop
hariketsheth@ubuntu:~/Desktop$ touch bootloader3.asm
hariketsheth@ubuntu:~/Desktop$ nano bootloader3.asm
hariketsheth@ubuntu:~/Desktop$ cat bootloader3.asm
[BITS 16]                       ;Tells the assembler that its a 16 bit code
[ORG 0x7C00]                    ;Origin, tell the assembler that where the code will
                                ;be in memory after it is been loaded
MOV SI, HelloString             ;Store string pointer to SI
CALL PrintString                ;Call print string procedure
JMP $                           ;Infinite loop, hang it here.

PrintCharacter:                 ;Procedure to print character on screen
                                ;Assume that ASCII value is in register AL
MOV AH, 0x0E                    ;Tell BIOS that we need to print one charater on screen.
MOV BH, 0x00                    ;Page no.
MOV BL, 0x07                    ;Text attribute 0x07 is lightgrey font on black background
INT 0x10                        ;Call video interrupt
RET                             ;Return to calling procedure

PrintString:                    ;Procedure to print string on screen
                                ;Assume that string starting pointer is in register SI
next_character:                 ;Lable to fetch next character from string
MOV AL, [SI]                    ;Get a byte from string and store in AL register
INC SI                          ;Increment SI pointer
OR AL, AL                       ;Check if value in AL is zero (end of string)
JZ exit_function                ;If end then return
CALL PrintCharacter             ;Else print the character which is in AL register
JMP next_character              ;Fetch next character from string
exit_function:                  ;End label
RET                             ;Return from procedure
                                ;Data
HelloString db 'Hariket', 0     ;Hariket string ending with 0

TIMES 510 - ($ - $$) db 0       ;Fill the rest of sector with 0
DW 0xAA55                       ;Add boot signature at the end of bootloader
hariketsheth@ubuntu:~/Desktop$
```

```
hariketsheth@ubuntu: ~/Desktop
hariketsheth@ubuntu:~/Desktop$ nasm bootloader3.asm -f bin -o boot3.bin
hariketsheth@ubuntu:~/Desktop$ qemu-system-x86_64 -drive file=boot3.bin,index=0,media=disk,format=raw
```

```
QEMU
Machine View

SeaBIOS (version 1.14.0-2)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F0B5B0+07ECB5B0 CA00

Booting from Hard Disk...
Hariket_
```

```
[BITS 16]                   ;Tells the assembler that its a 16 bit code
[ORG 0x7C00]                ;Origin, tell the assembler that where the code will
                            ;be in memory after it is been loaded
MOV SI, HelloString         ;Store string pointer to SI
CALL PrintString            ;Call print string procedure
JMP $                       ;Infinite loop, hang it here.

PrintCharacter:             ;Procedure to print character on screen
                            ;Assume that ASCII value is in register AL
MOV AH, 0x0E                ;Tell BIOS that we need to print one charater on screen.
MOV BH, 0x00                ;Page no.
MOV BL, 0x07                ;Text attribute 0x07 is lightgrey font on black background
INT 0x10                    ;Call video interrupt
RET                         ;Return to calling procedure

PrintString:                ;Procedure to print string on screen
                            ;Assume that string starting pointer is in register SI
next_character:             ;Lable to fetch next character from string
MOV AL, [SI]                ;Get a byte from string and store in AL register
INC SI                      ;Increment SI pointer
OR AL, AL                   ;Check if value in AL is zero (end of string)
JZ exit_function            ;If end then return
CALL PrintCharacter         ;Else print the character which is in AL register
JMP next_character          ;Fetch next character from string
exit_function:              ;End label
RET                         ;Return from procedure
                            ;Data
HelloString db 'Hariket', 0 ;Hariket string ending with 0

TIMES 510 - ($ - $$) db 0   ;Fill the rest of sector with 0
DW 0xAA55                   ;Add boot signature at the end of bootloader
```