Name: Shangirne Kharbanda
Registration Number: 20BAI1154

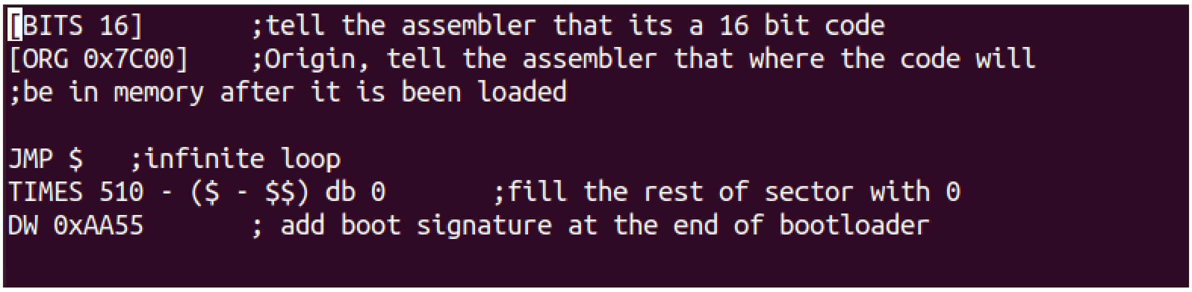
OS LAB-2

NASM installation and bootimage:

sudo apt-get install nasm

vi firstBootLoader.asm

Code inside firstBootLoader.asm file:



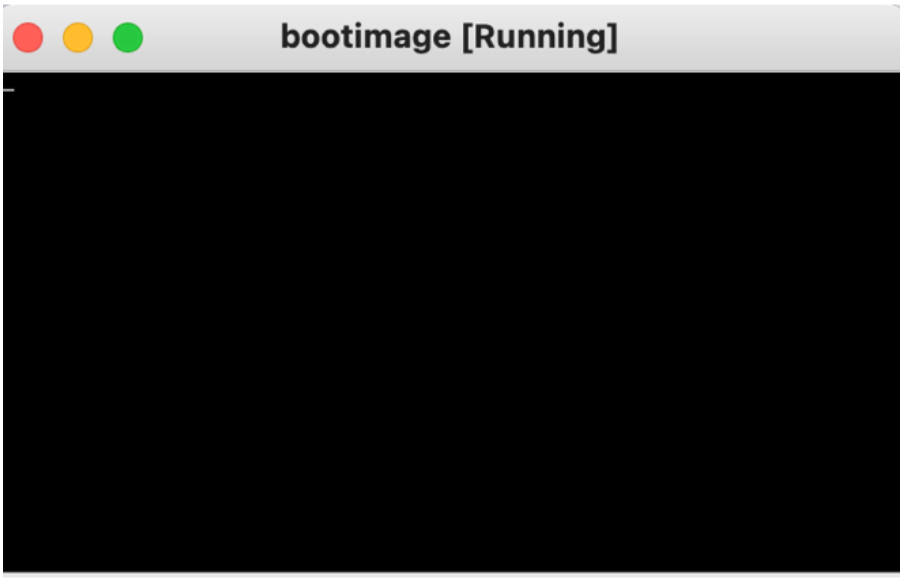
```
[BITS 16]       ;tell the assembler that its a 16 bit code
[ORG 0x7C00]    ;Origin, tell the assembler that where the code will
;be in memory after it is been loaded

JMP $   ;infinite loop
TIMES 510 - ($ - $$) db 0       ;fill the rest of sector with 0
DW 0xAA55       ; add boot signature at the end of bootloader
```

Compile the program:

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

Create a floppy image using:

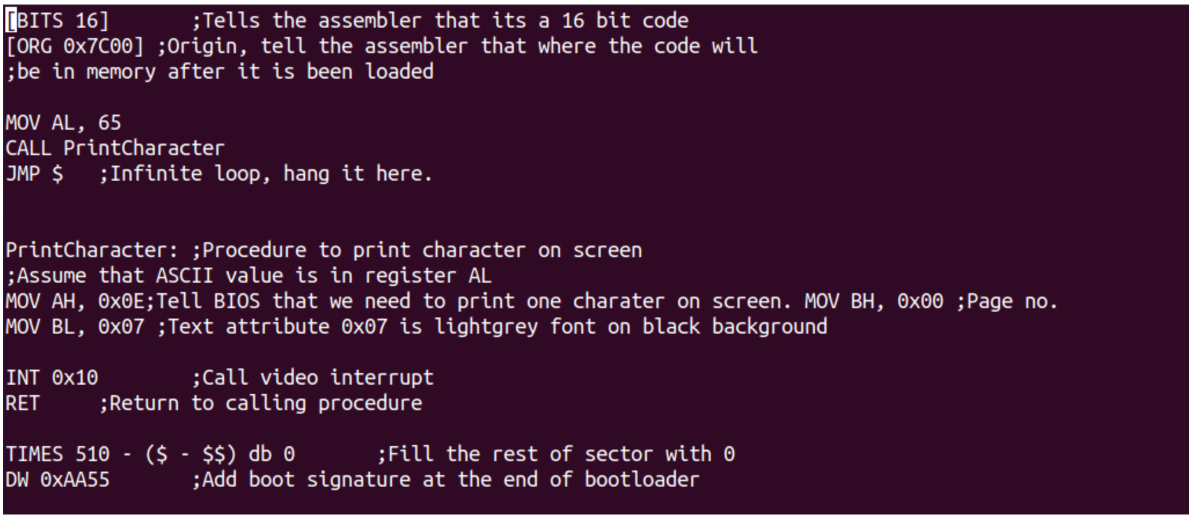
dd if=boot.bin bs=512 of=floppy1.img



1. Print A

vi a.asm

Code inside a.asm file:



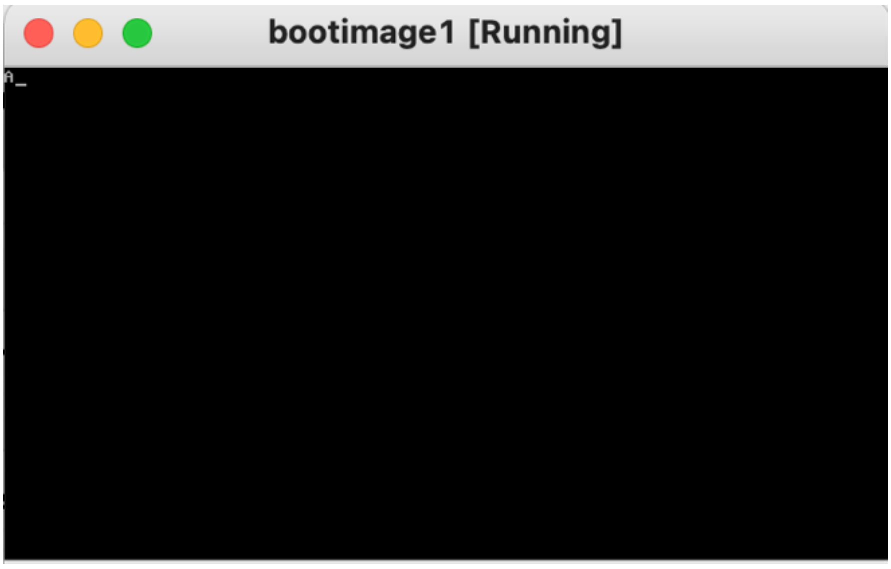
```
[BITS 16]       ;Tells the assembler that its a 16 bit code
[ORG 0x7C00] ;Origin, tell the assembler that where the code will
;be in memory after it is been loaded

MOV AL, 65
CALL PrintCharacter
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

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

Compile the program:

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

Create a floppy image using:

dd if=boot.bin bs=512 of=floppya.img

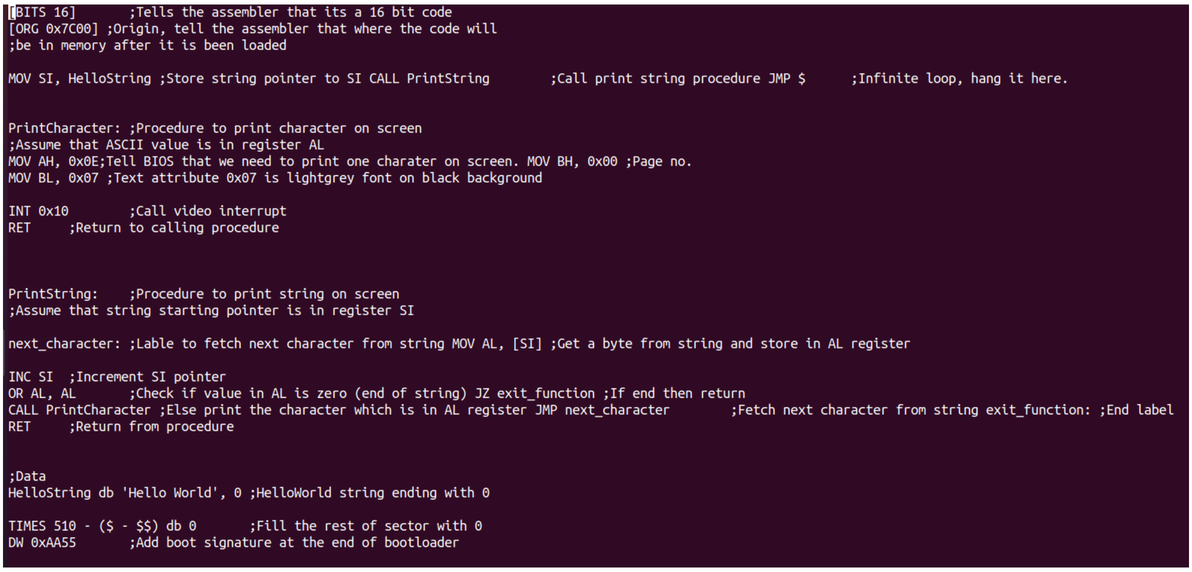


```
A_
```

2. Print hello world

vi hello.asm

Code inside hello.asm file:



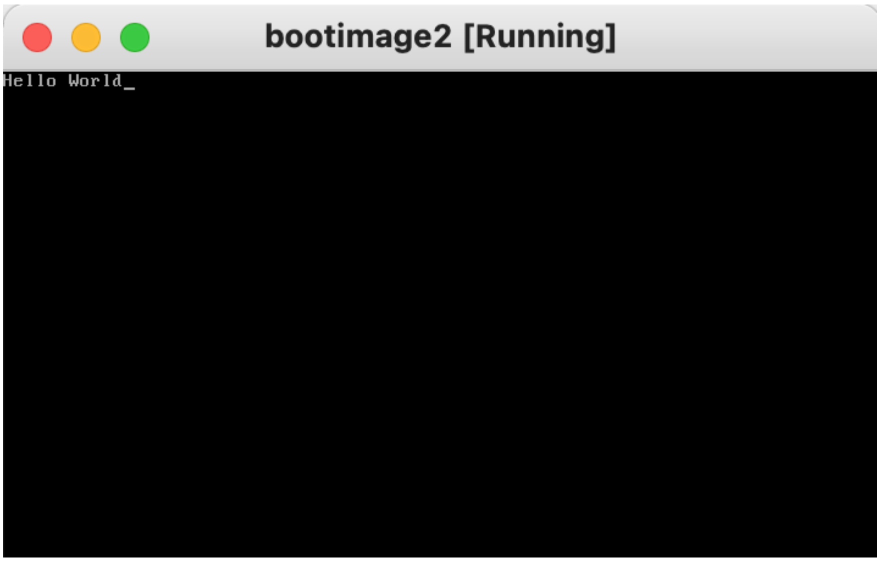
```
[BITS 16]       ;Tells the assembler that its a 16 bit code
[ORG 0x7C00] ;Origin, tell the assembler that where the code will
;be in memory after it is been loaded

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 'Hello World', 0 ;HelloWorld 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
```

Compile the program:

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

Create a floppy image using:

dd if=boot.bin bs=512 of=floppyhello.img



```
Hello World_
```