OPERATING SYSTEMS

Lab-3: BOOT LOADER CURSOR, ALPHABET AND DISPLAY MESSAGE

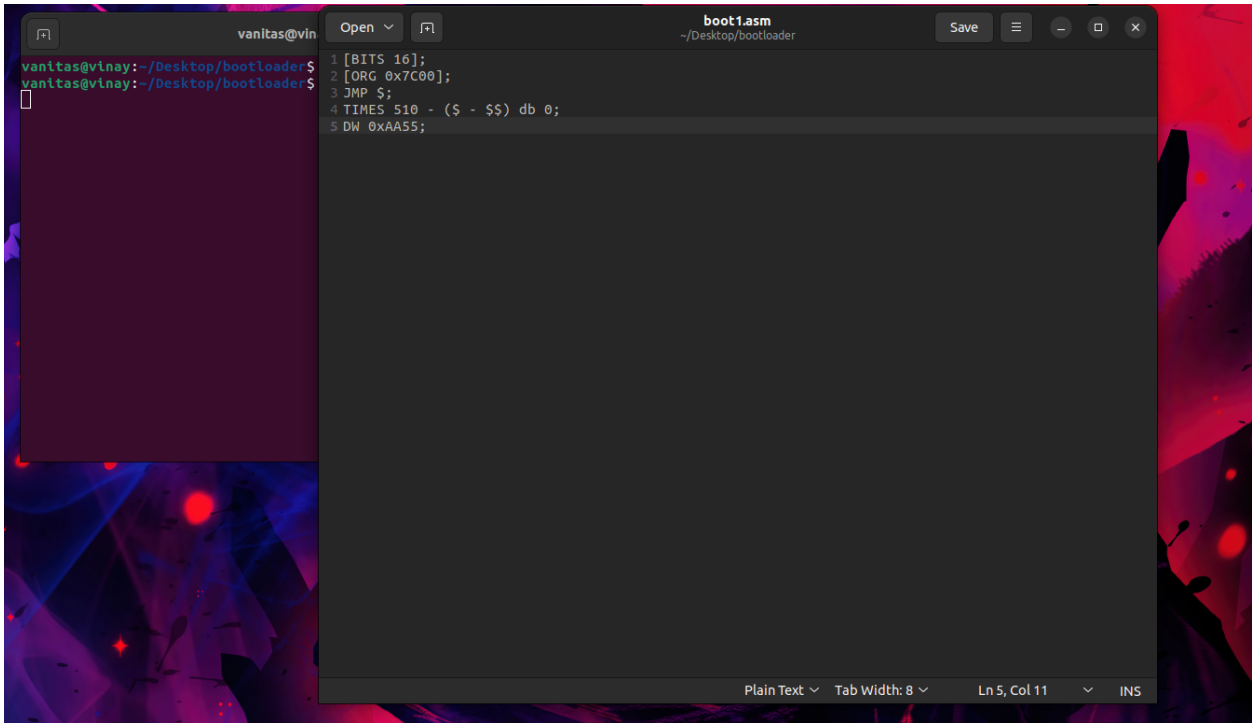
Name – Vinay Santosh Menon
Registration Number - 20BAI1103

TASK:

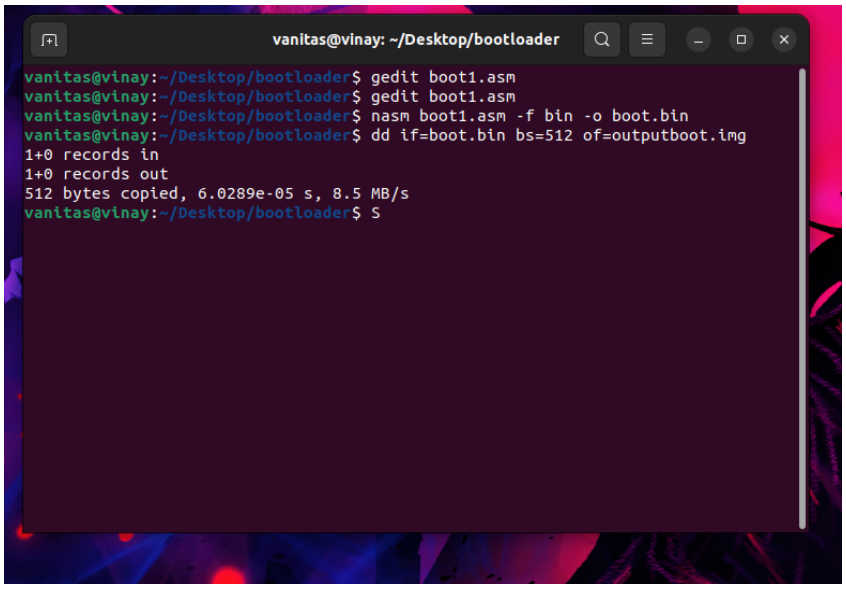
Creating a bootloader program in assembly language, compiling it and running it in a virtual machine.

Empty bootloader cursor:

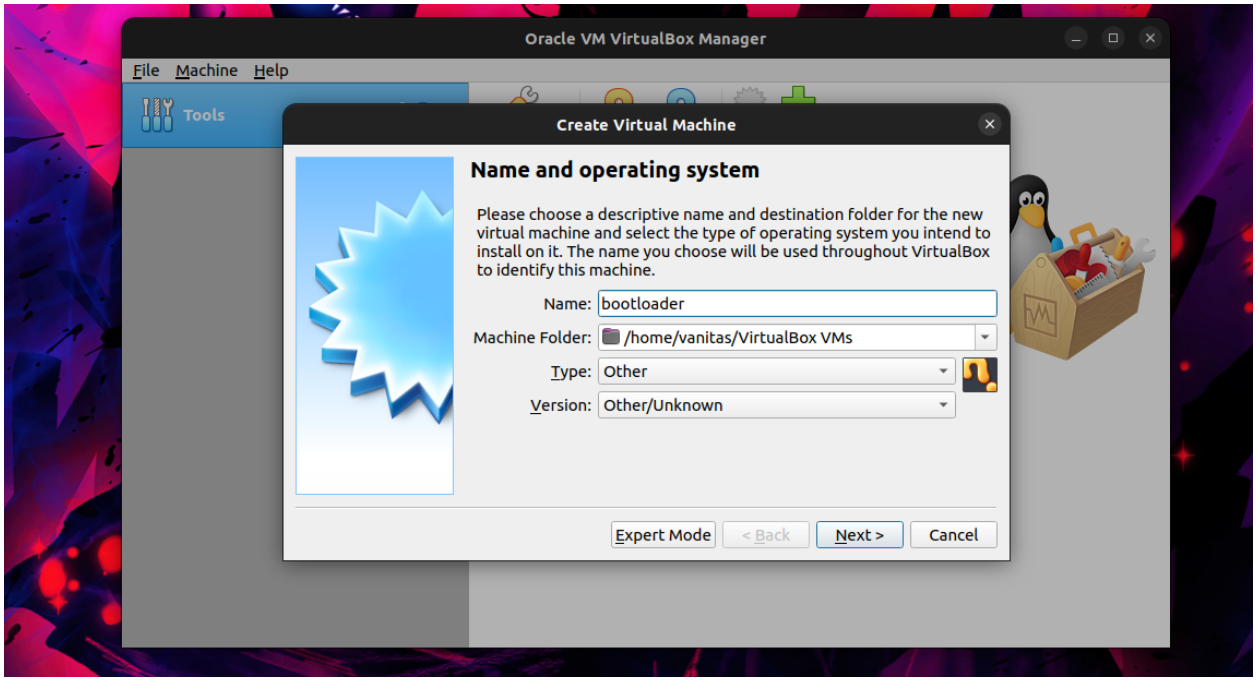
Write the below assembly code in a text editor and save it as a .asm file:

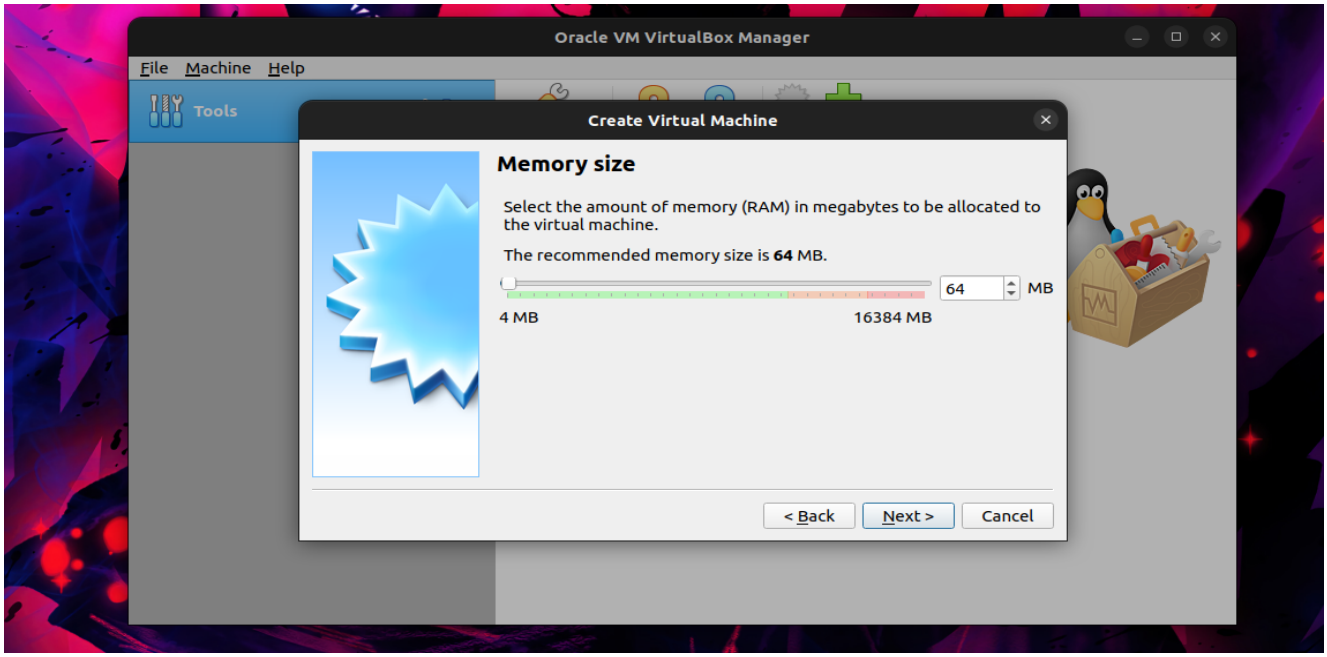


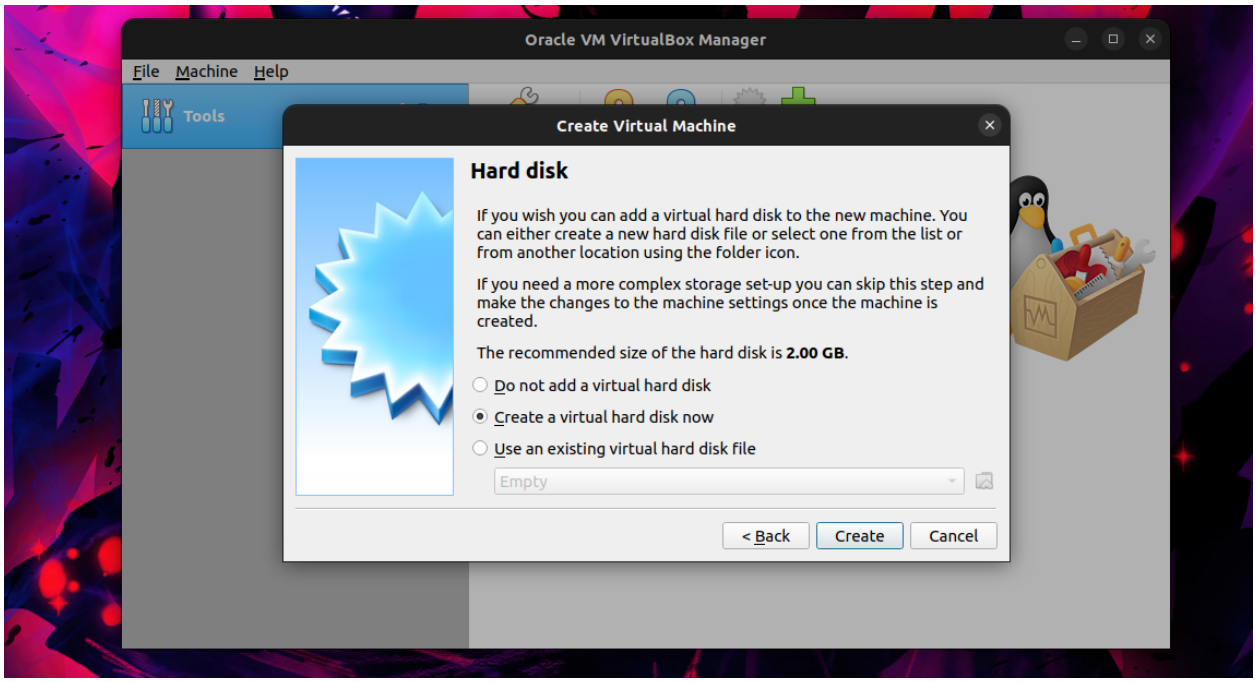
Compile the code using nasm in the Linux terminal to convert it first into a binary file and then into an iso img file. The commands below are used to achieve this.

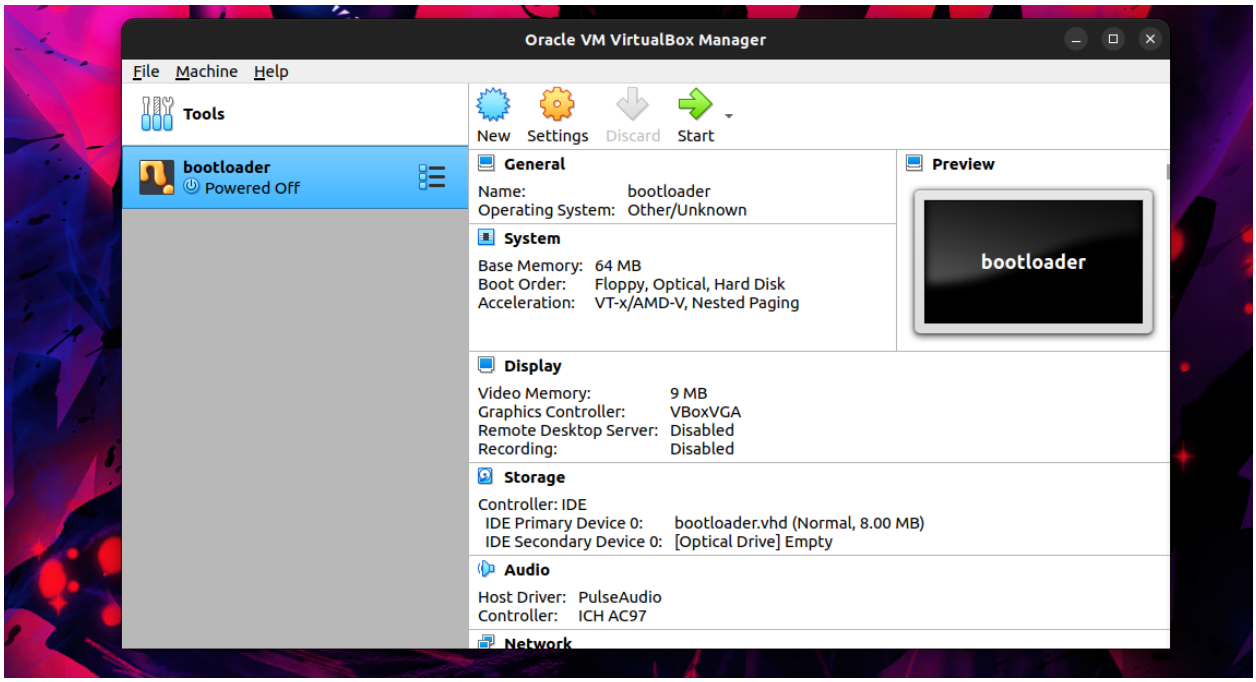


Start the virtual box to then create a virtual machine in it.

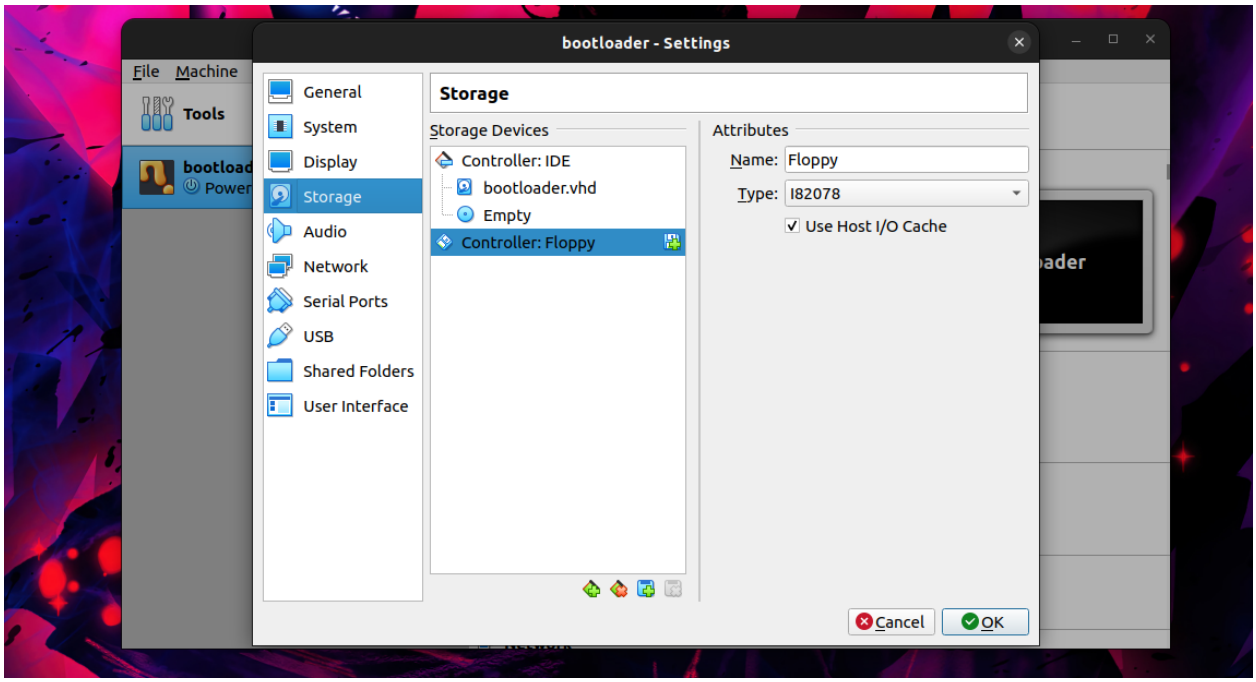




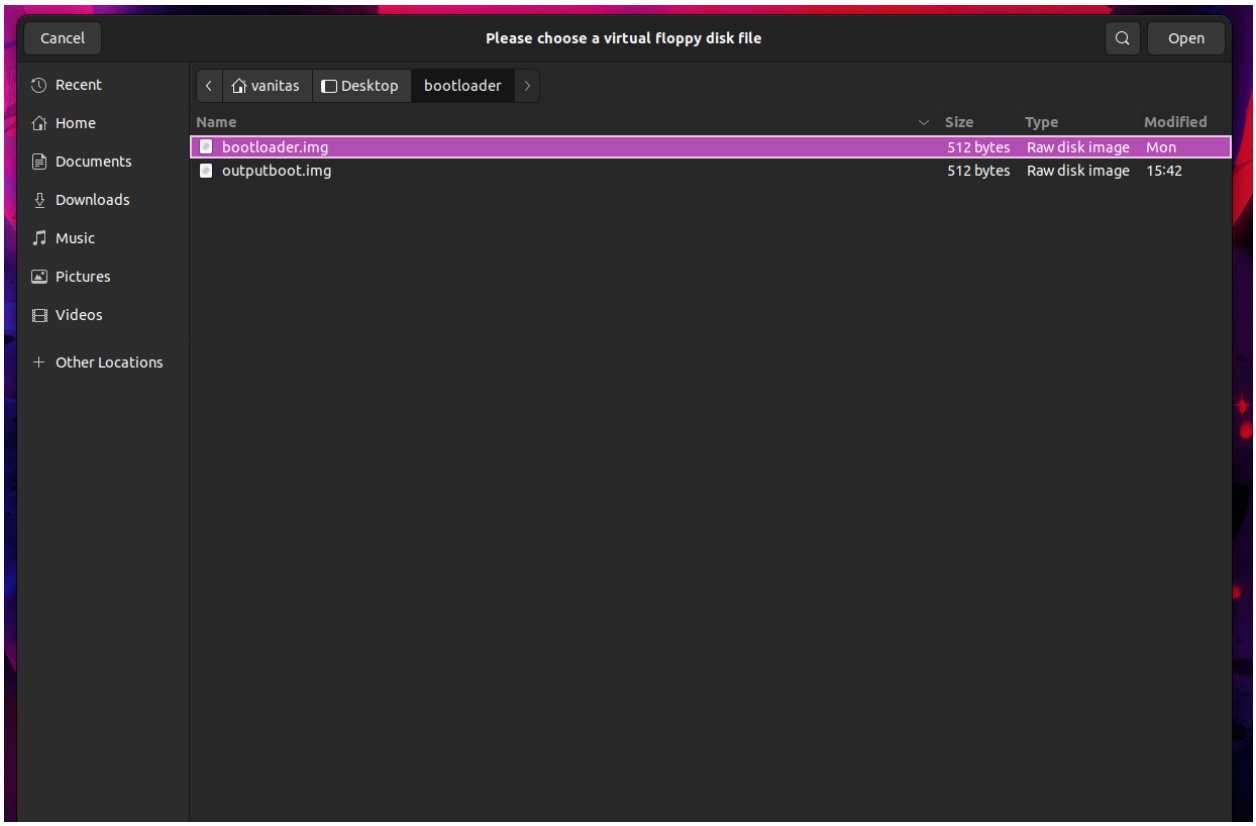


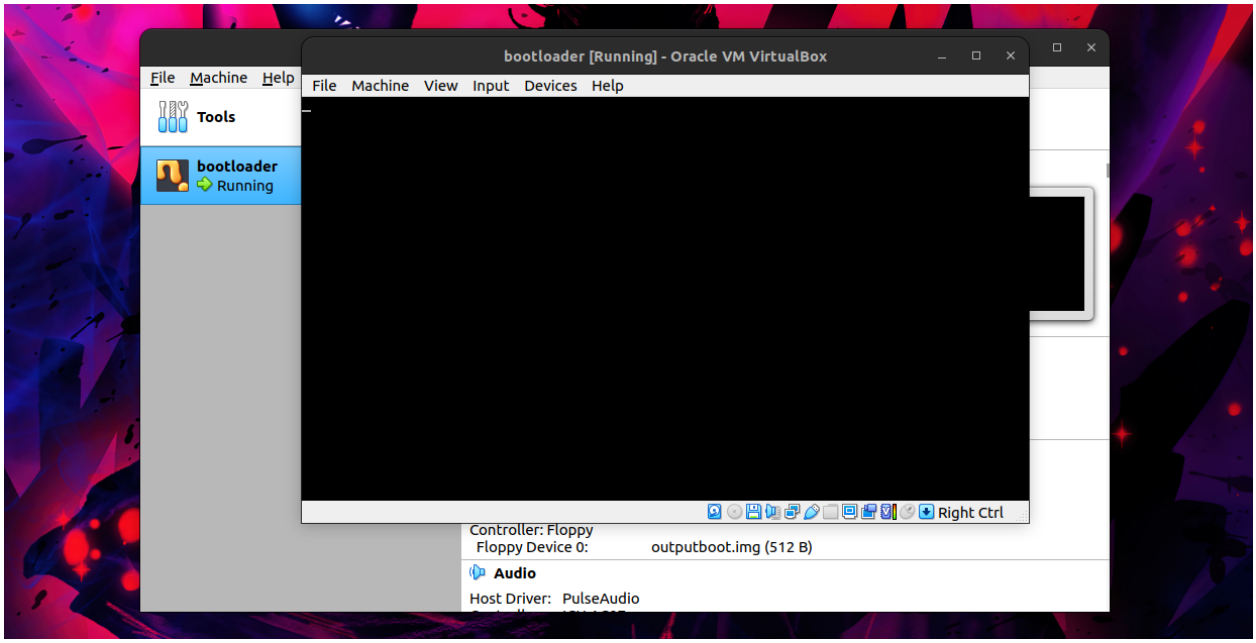


Click on settings > storage and add a floppy to the device.



Add the img file in the floppy.





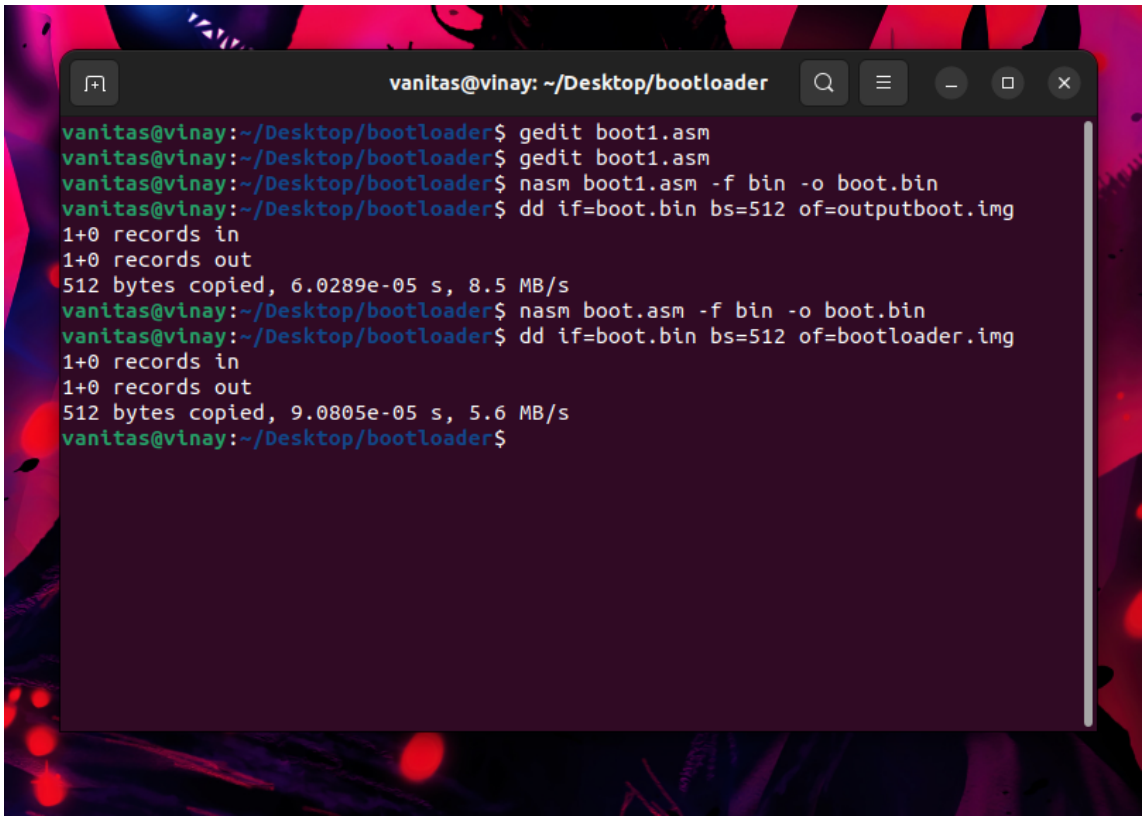
Now run the VM, you should see an empty screen with a blinking cursor.

Printing alphabet in bootloader

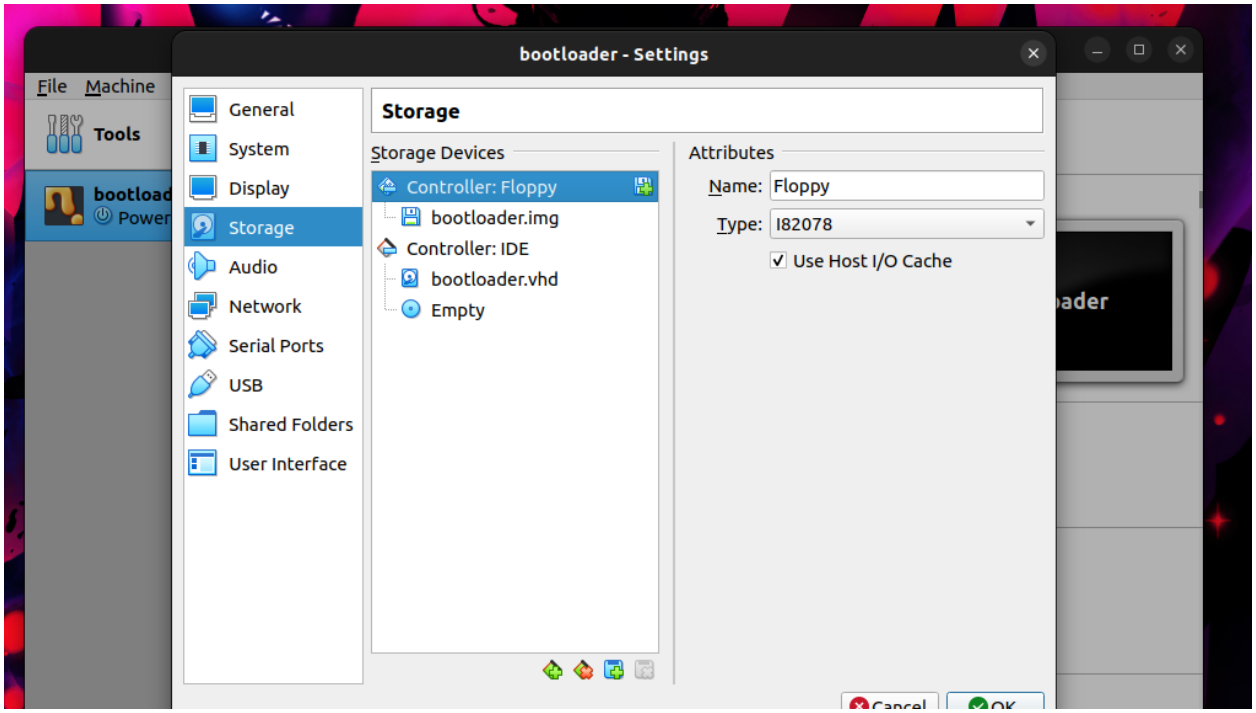
Write the assembly code in a simple text editor and save it as a ".asm" file.

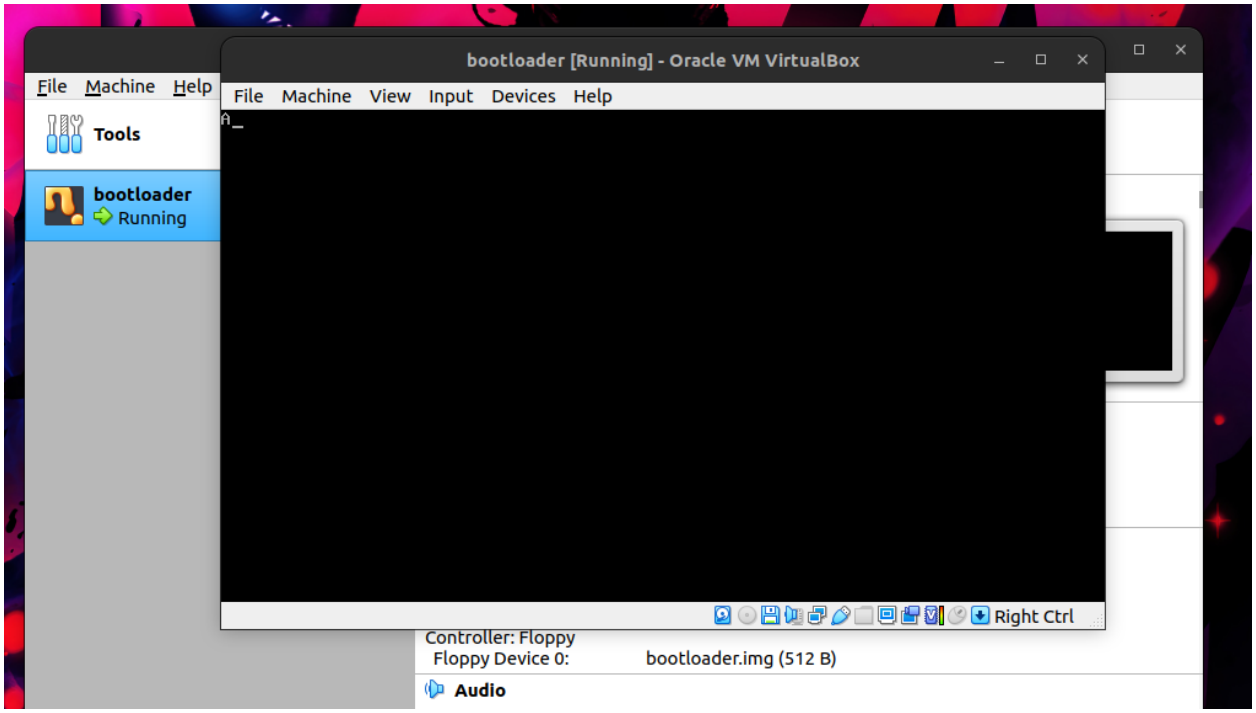
```
Open   boot.asm   Save
~/Desktop/bootloader

1 [BITS 16]
2 [ORG 0x7c00]
3 MOV AL, 65;
4 CALL PrintCharacter;
5 JMP $
6 PrintCharacter:
7 MOV AL, 65
8 MOV AH, 0x0E
9 MOV BH, 0x00
10 MOV BL, 0x07
11 INT 0x10
12 RET
13 TIMES 510-($-$$) db 0
14 DW 0xAA55
```

Compile the code using nasm in the Linux terminal to convert it first into a binary file and then into an iso img file.



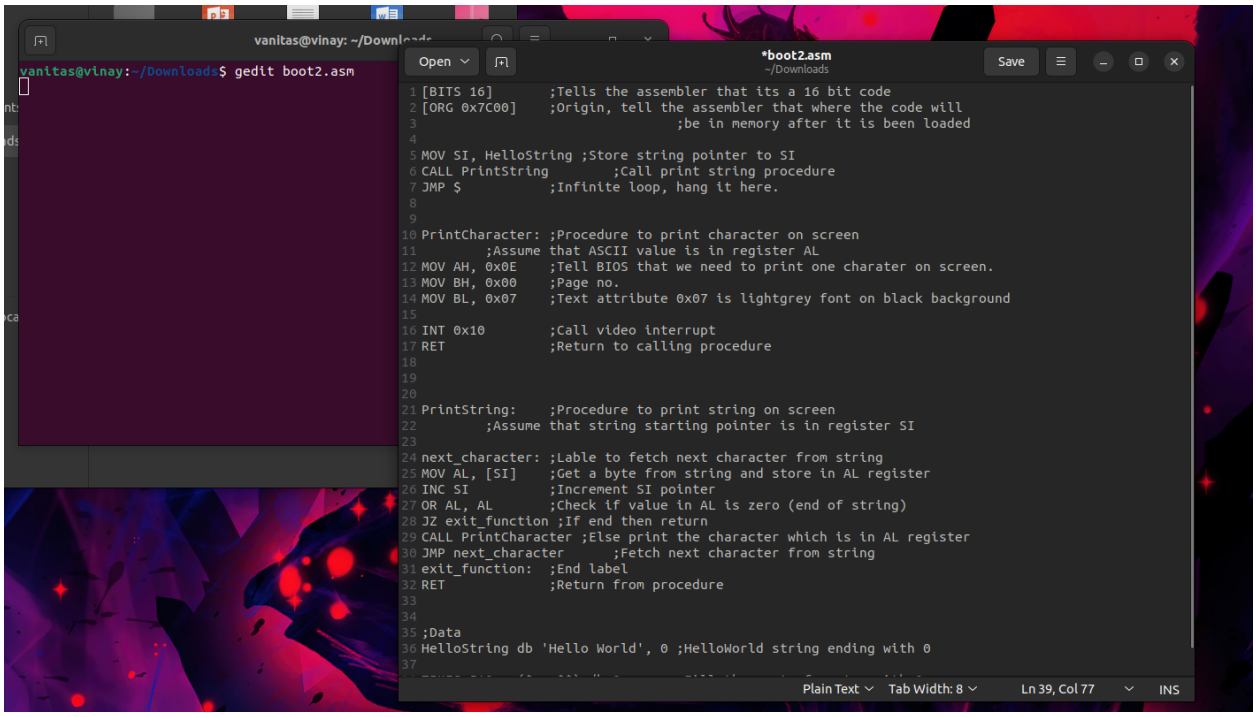
Start the virtual box to then create a virtual machine in it, and repeat the same steps.





Displaying “Hello World” in bootloader:

Write the assembly code in a simple text editor and save it as a “.asm” file.

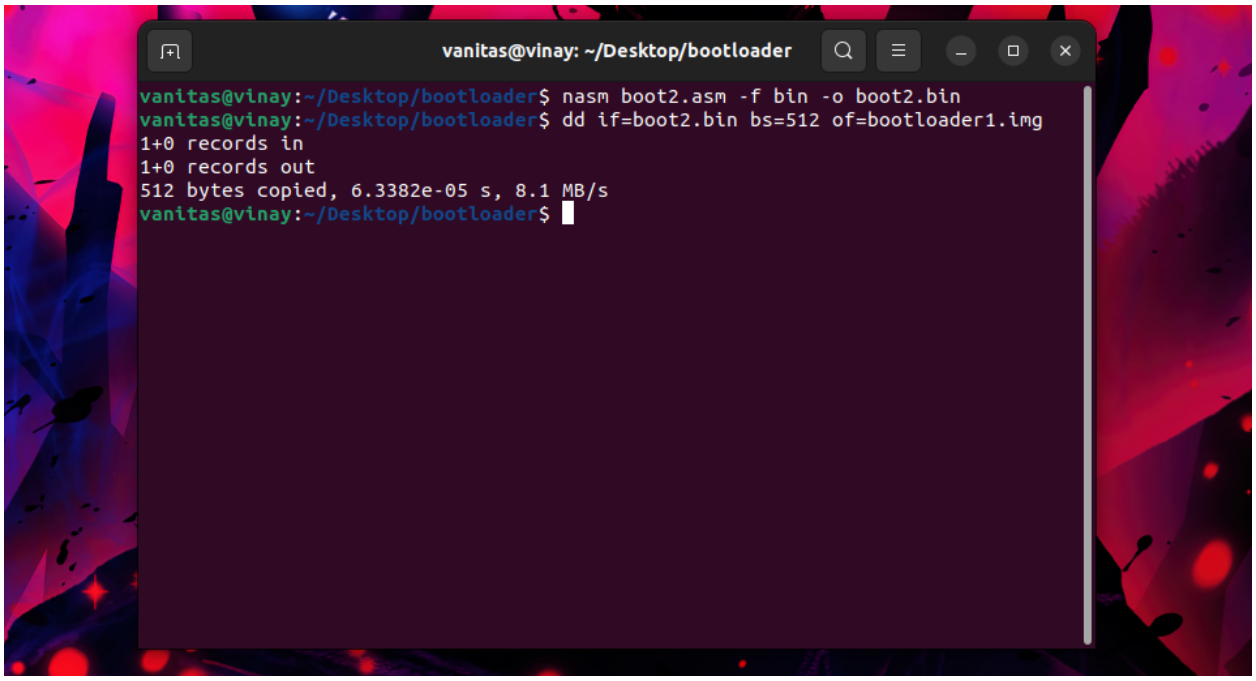


Code:

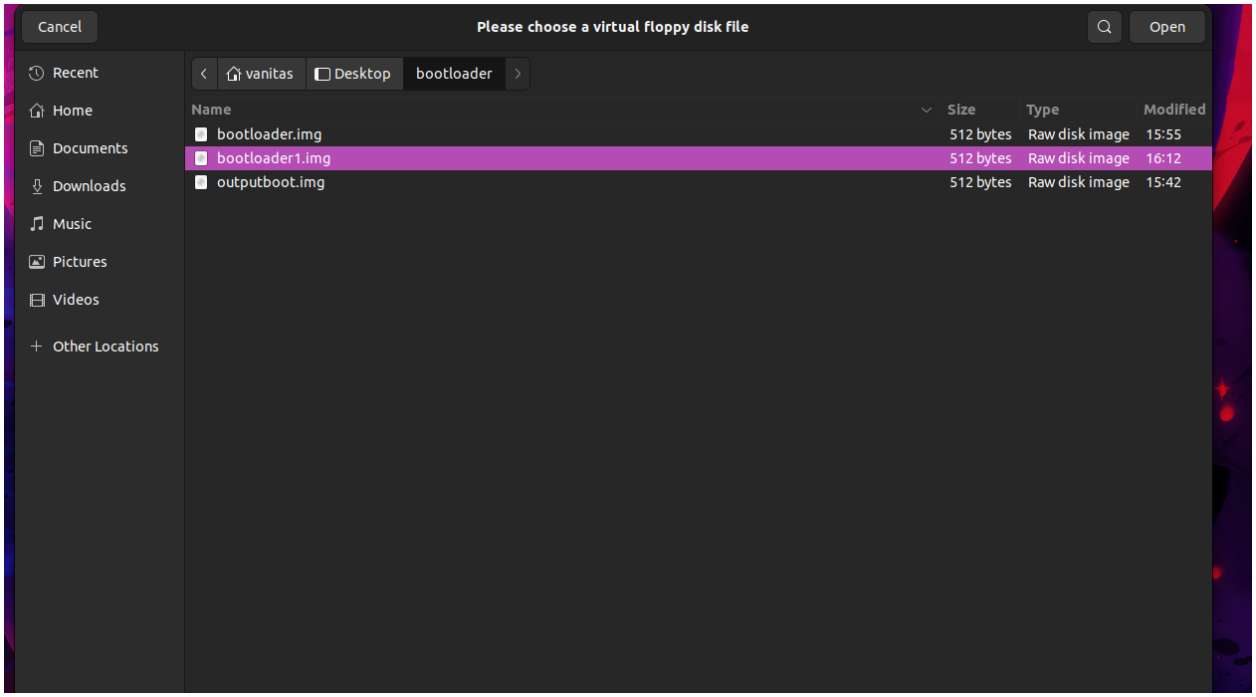
```
[BITS 16]       ;Tells the assembler that its a 16 bit code
[ORG 0x7C00]    ;Origin, tell the assembler that where the code will
                ;be in memory after it is been loaded

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 code using nasm in the Linux terminal to convert it first into a binary file and then into an iso img file.



Start the virtual box to then create a virtual machine in it, and repeat the same steps.



We can see “Hello World” printed.