DHRUV GARG 16 BCE 1190

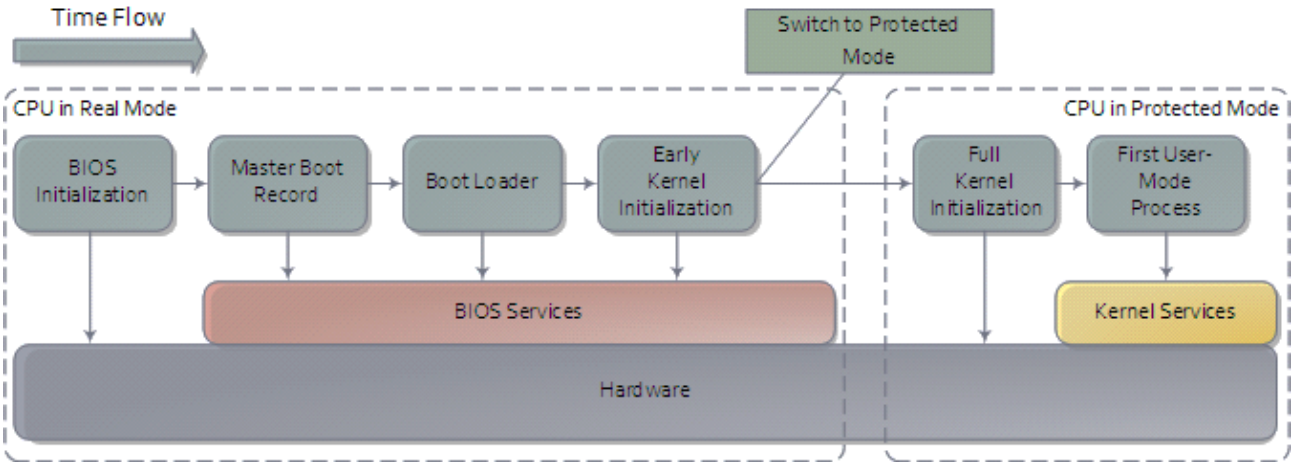
BOOT SEQUENCE

LAB 2

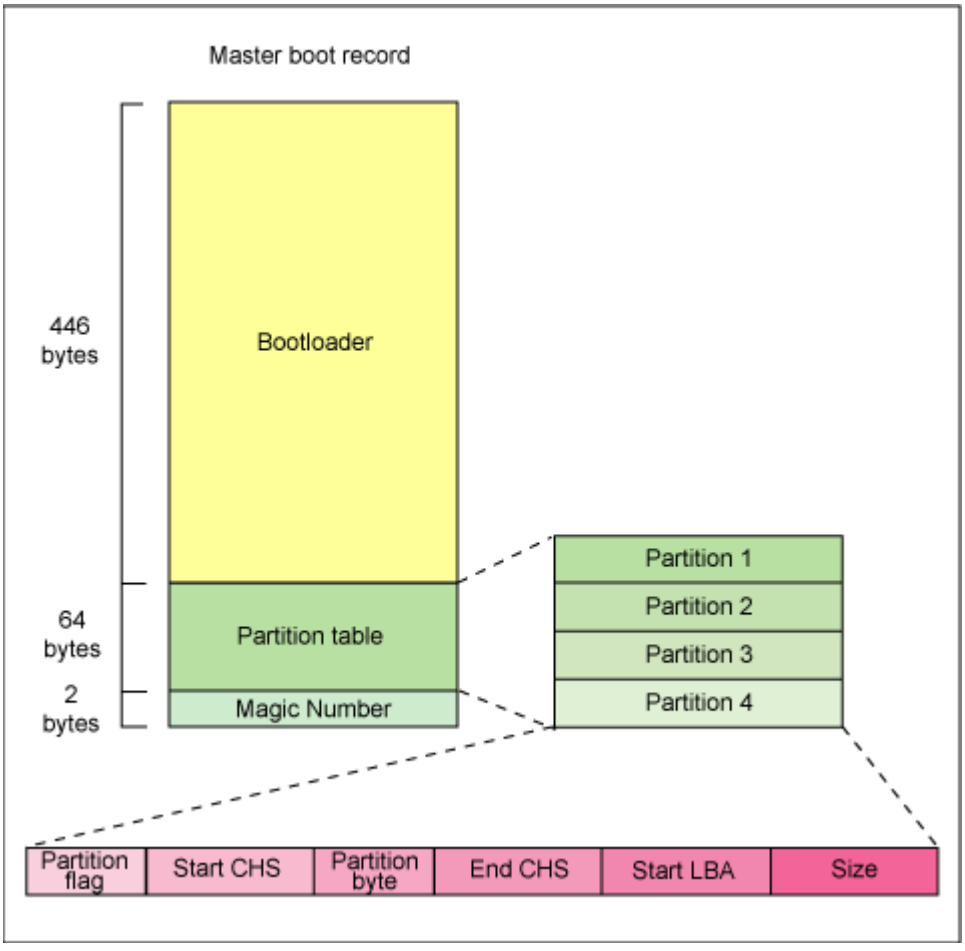
Aim: To understand the functioning of the boot loader and to implement the same.

THEORY:

Boot sequence



Boot loader



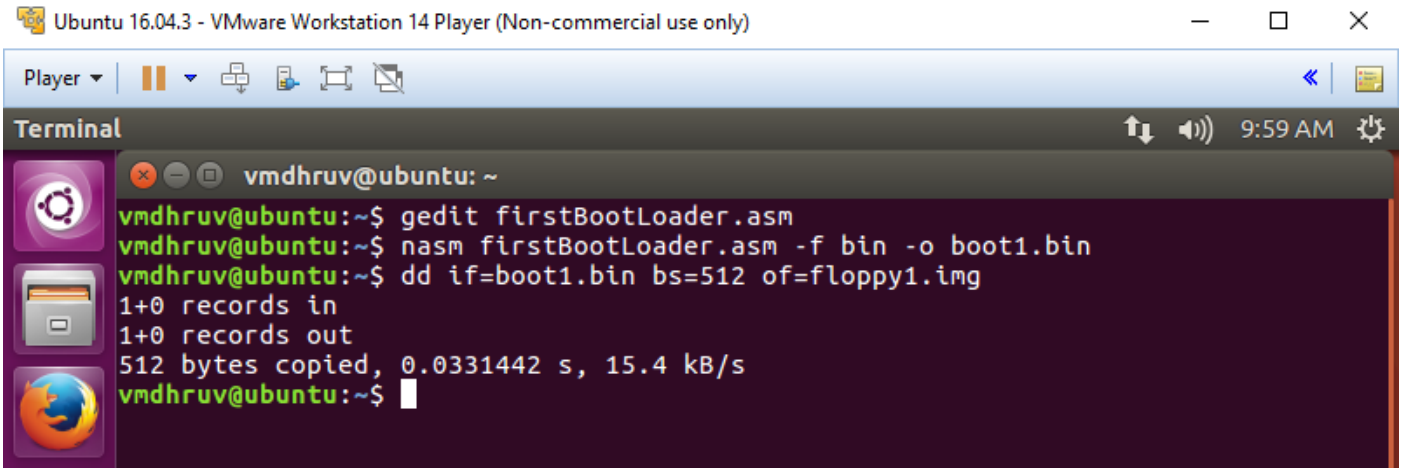
DHRUV GARG 16 BCE 1190

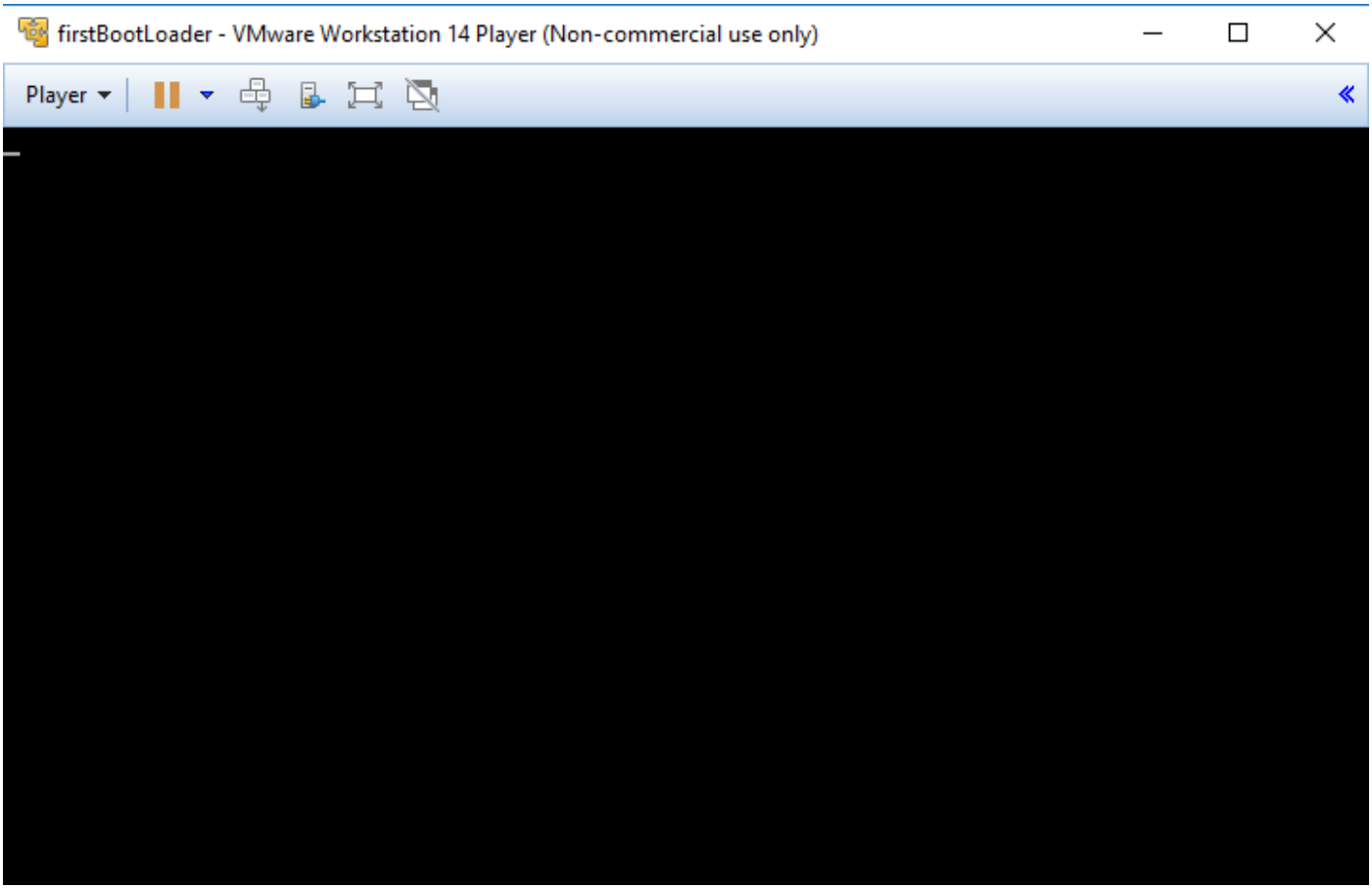
BOOT LOADER 1: Blank black screen

CODE

```
[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
```

SCREENSHOTS





DHRUV GARG 16 BCE 1190

BOOT LOADER 2: Prints character 'A' on the screen

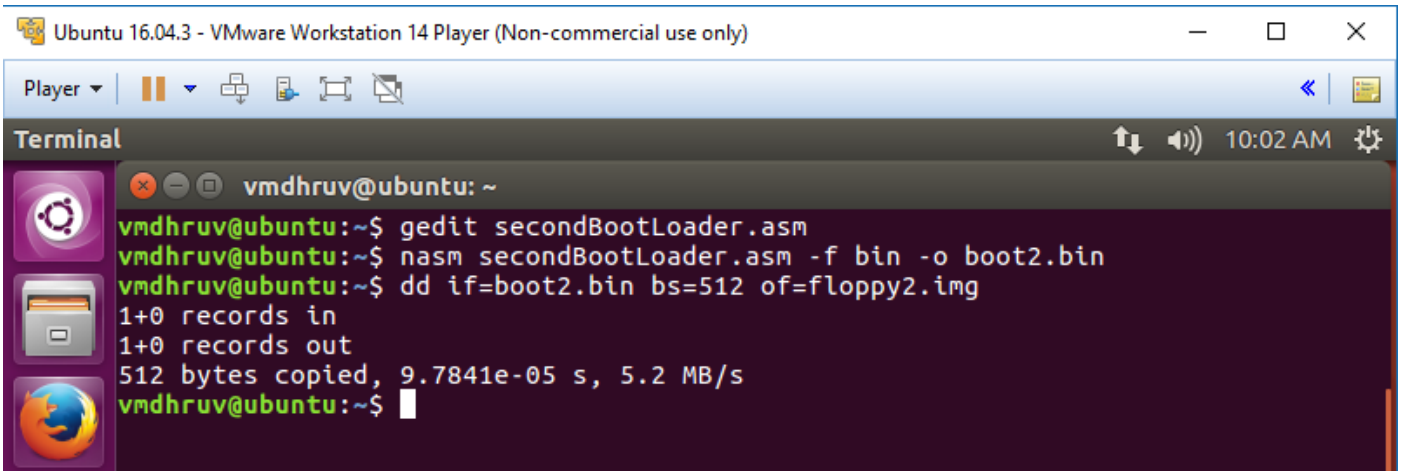
CODE

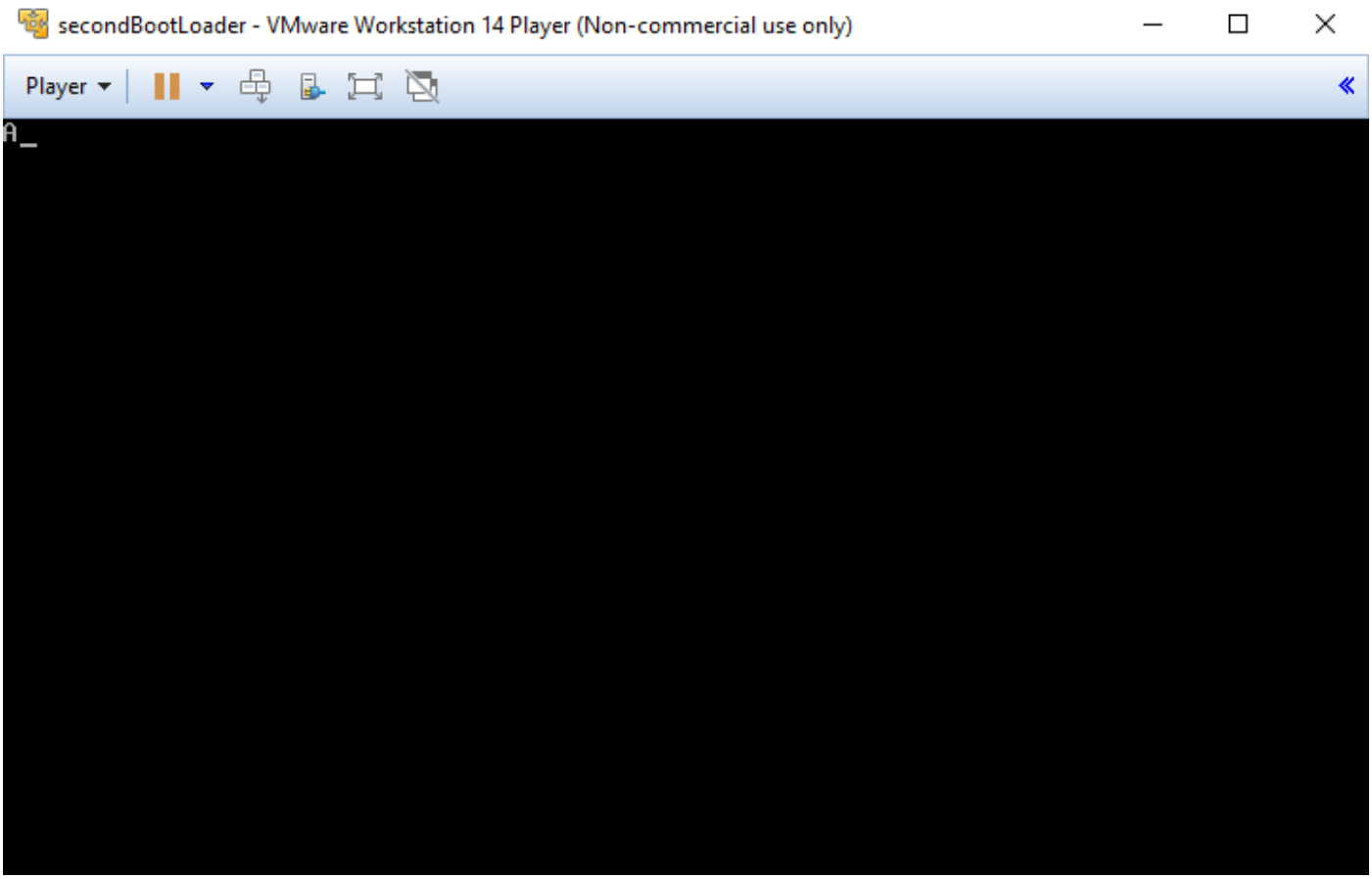
```
[BITS 16]          ;Tells the assembler that its a 16 bit code
[ORG 0x7C00]       ;Origin, tell the assembler that where the code will be in memory after it is been loaded
MOV AL, 65
CALL PrintCharacter
JMP $              ;Infinite loop, hang it here.


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
```

SCREENSHOTS





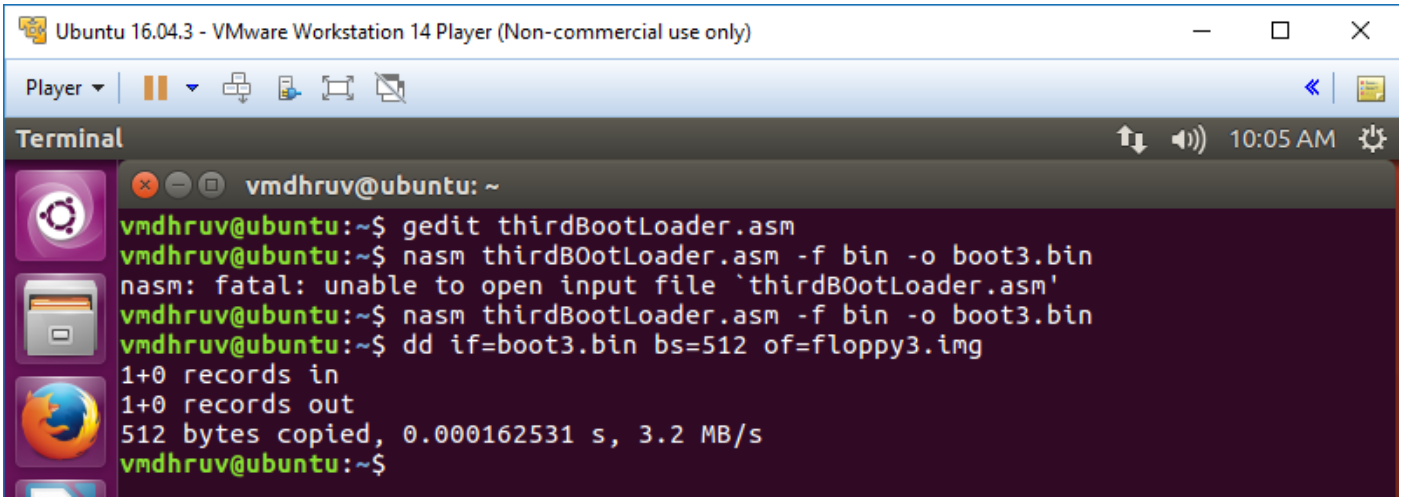
BOOT LOADER 3: Prints 'Hello world'

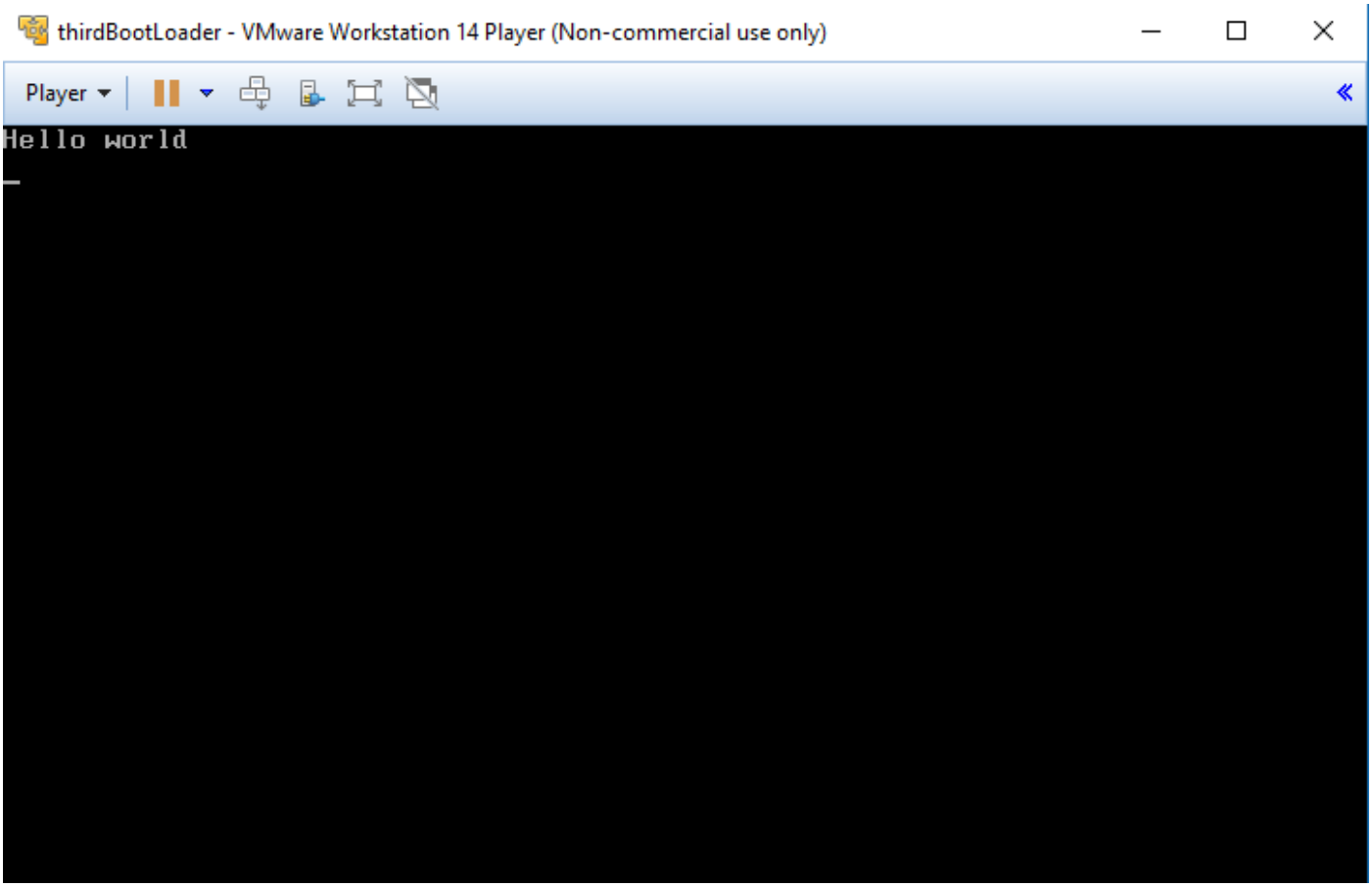
CODE:

```
[BITS 16]
jmp main
nop
main:
        mov ax, 07C0h
        add ax, 288
        mov ss, ax
        mov sp, 4096
        mov ax, 07C0h
        mov ds, ax
        call GetPressedKey
        jmp .InfiniteLoop
        .InfiniteLoop:
                jmp .InfiniteLoop
HelloWorld      db      "Hello world",0x0d,0x0a,0x00
```

```
GetPressedKey:
        mov ah, 0
        int 0x16
        CALL PrintHelloWorld
        ret
PrintHelloWorld:
        mov si, HelloWorld
        call PrintStr
        ret
PrintStr:
        push ax
        mov ah, 0Eh
        .loop:
                lodsb
                cmp al, 0x00
                je .done
                int 10h
                jmp .loop
        .done:
                pop ax
                ret
times 510-($-$$) db 0
dw 0xAA55
```

SCREENSHOTS





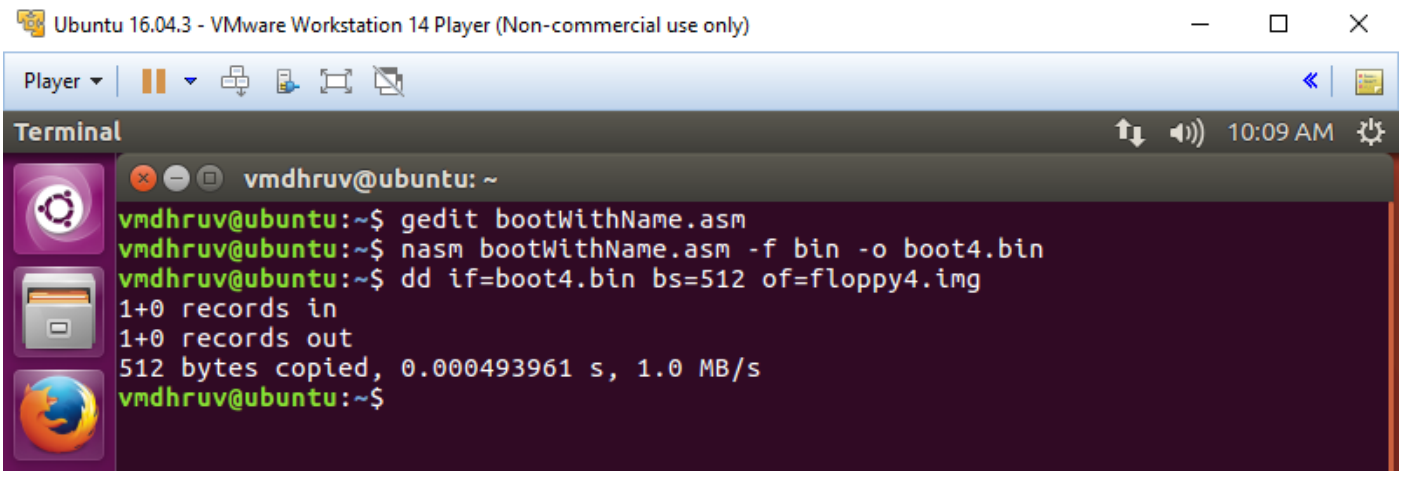
BOOT LOADER 4: Prints 'Hello, Dhruv Garg'

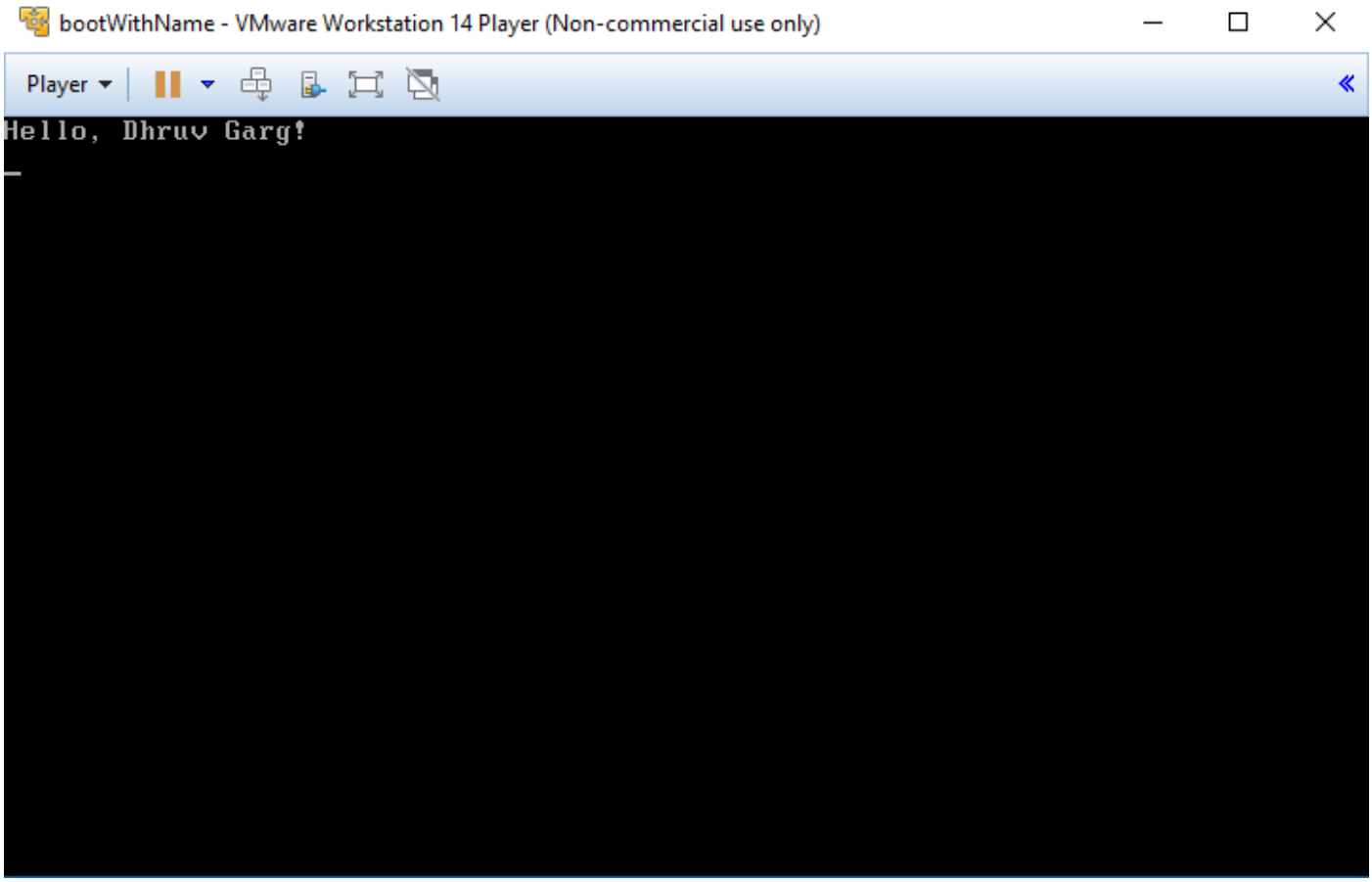
CODE

```
[BITS 16]
jmp main
nop
main:
        mov ax, 07C0h
        add ax, 288
        mov ss, ax
        mov sp, 4096
        mov ax, 07C0h
        mov ds, ax
        call GetPressedKey
        jmp .InfiniteLoop
        .InfiniteLoop:
                jmp .InfiniteLoop
HelloUser       db      "Hello, Dhruv Garg!",0x0d,0x0a,0x00
```

```
GetPressedKey:
        mov ah, 0
        int 0x16
        CALL PrintHelloUser
        ret
PrintHelloUser:
        mov si, HelloUser
        call PrintStr
        ret
PrintStr:
        push ax
        mov ah, 0Eh
        .loop:
                lodsb
                cmp al, 0x00
                je .done
                int 10h
                jmp .loop
        .done:
                pop ax
                ret
times 510-($-$$) db 0
dw 0xAA55
```

SCREENSHOTS





SCREENSHOT: **Different bootloaders created**