Operating Systems Lab

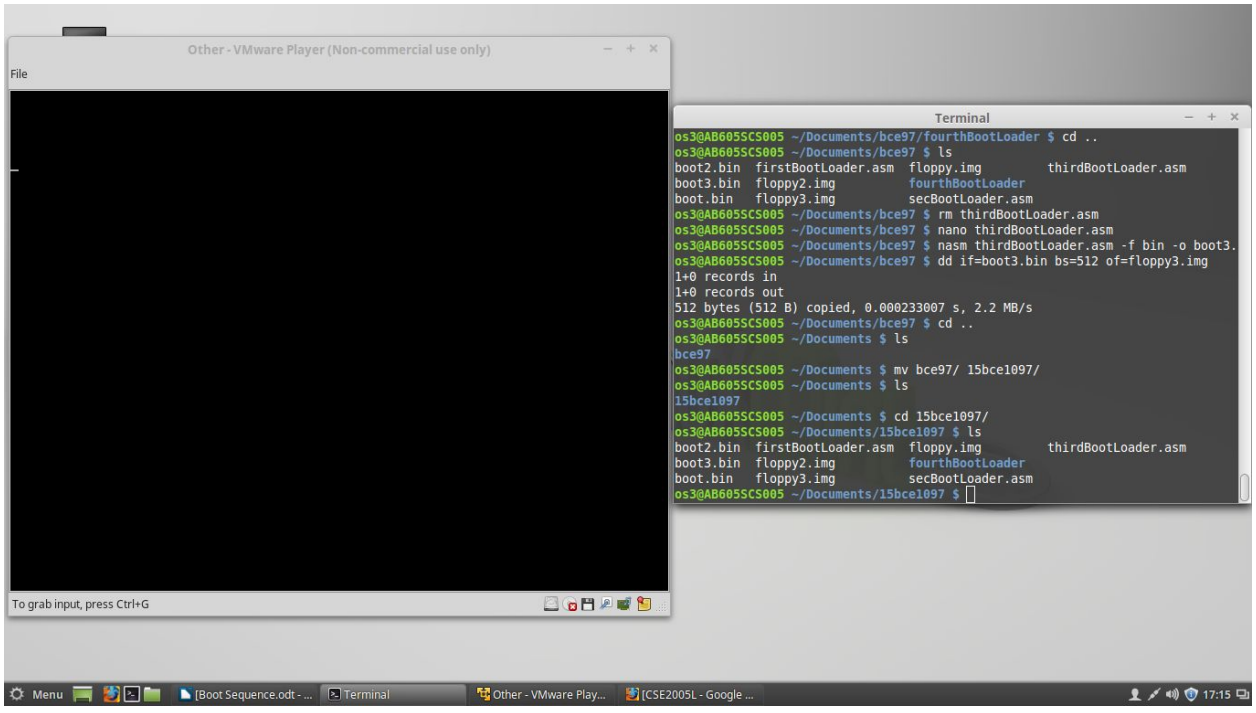
Prof. L Shyamala

24th January, 2016

PRANAVCHENDUR T K - 15BCE1097

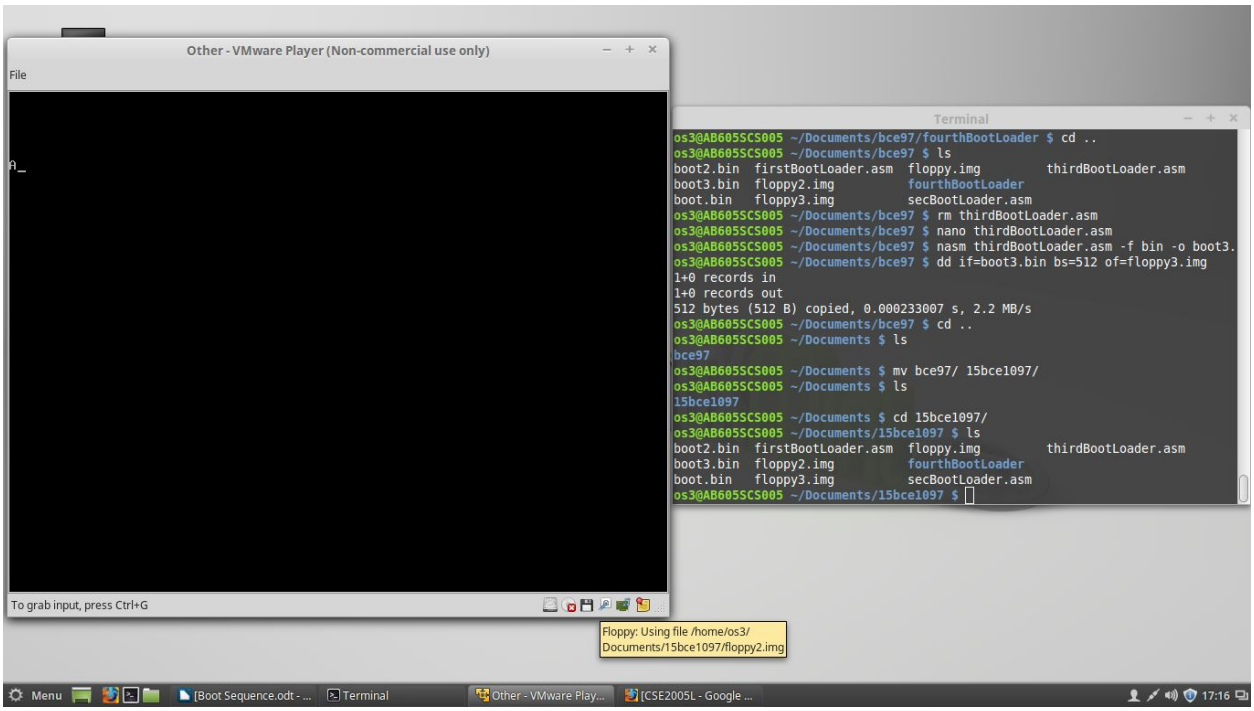
firstBootLoader.asm

```
[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
```



secBootLoader.asm

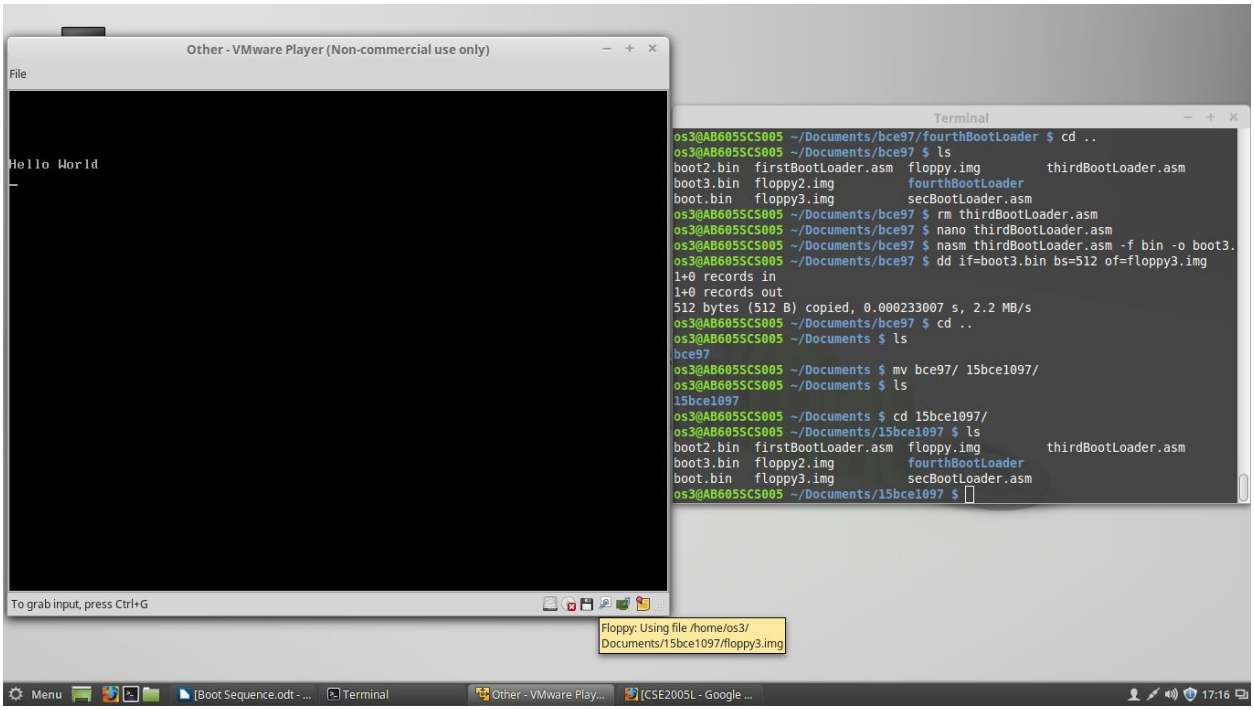
```
[BITS 16]       ;Tells the assembler that its a 16 bit code
[ORG 0x7C00]        ;Origin, tell the assembler that where the code will
        ;be in memory after it is been loaded

MOV 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
```



thirdBootLoader.asm

```
[BITS 16]       ; 16 bit code generation
[ORG 0x7C00]   ; Origin location

; Main program
main:           ; Label for the start of the main program

 mov ax,0x0000       ; Setup the Data Segment register
        ; Location of data is DS:Offset
 mov ds,ax    ; This can not be loaded directly it has to be in two steps.
        ; 'mov ds, 0x0000' will NOT work due to limitations on the CPU

 mov si, HelloWorld ; Load the string into position for the procedure.
 call PutStr    ; Call/start the procedure

jmp $           ; Never ending loop

; Procedures
PutStr:         ; Procedure label/start
 ; Set up the registers for the interrupt call
 mov ah,0x0E         ; The function to display a chacter (teletype)
 mov bh,0x00         ; Page number
 mov bl,0x07 ; Normal text attribute

.nextchar:      ; Internal label (needed to loop round for the next character)
 lodsb          ; I think of this as LOaD String Block
        ; (Not sure if thats the real meaning though)
        ; Loads [SI] into AL and increases SI by one
 ; Check for end of string '0'
 or al,al       ; Sets the zero flag if al = 0
        ; (OR outputs 0's where there is a zero bit in the register)
 jz .return      ; If the zero flag has been set go to the end of the procedure.
        ; Zero flag gets set when an instruction returns 0 as the answer.
 int 0x10       ; Run the BIOS video interrupt
 jmp .nextchar         ; Loop back round tothe top
.return:        ; Label at the end to jump to when complete
 ret     ; Return to main program

; Data
```

```
HelloWorld db 'Hello World',13,10,0

; End Matter
times 510-($-$$) db 0        ; Fill the rest with zeros
dw 0xAA55                    ; Boot loader signature
```



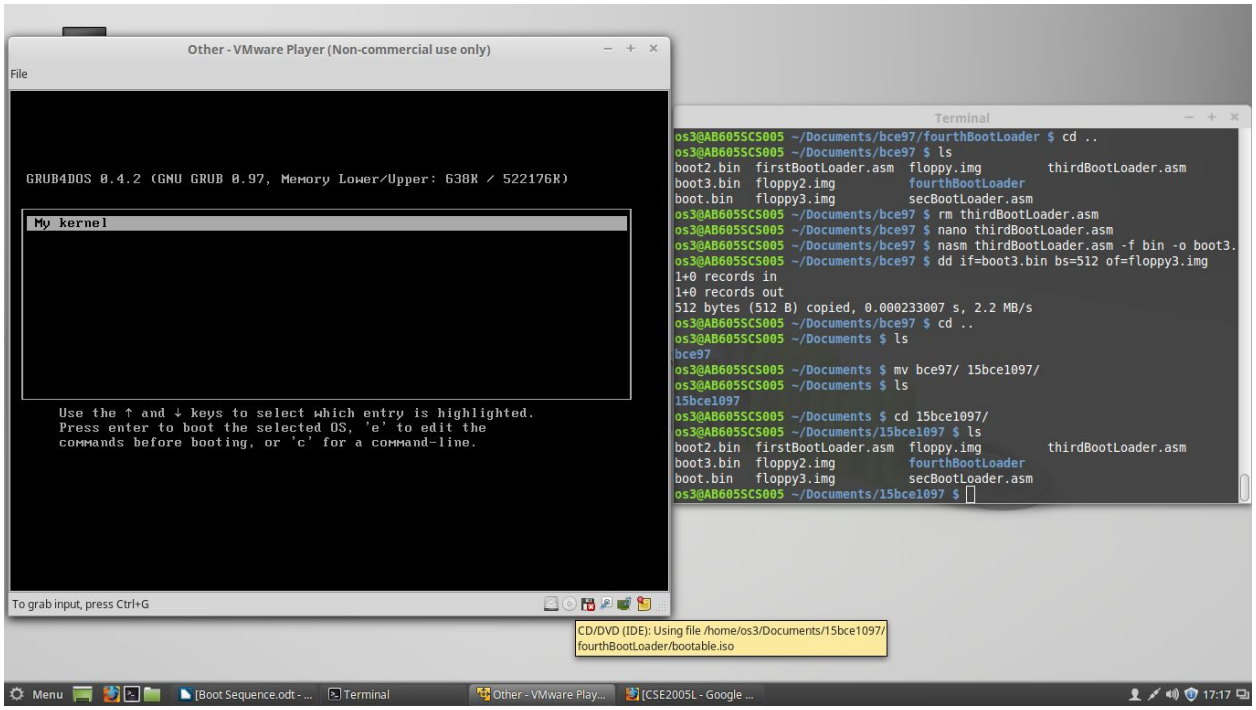
Fourth - Kernel
menu.lst
default 0
#timeout 30

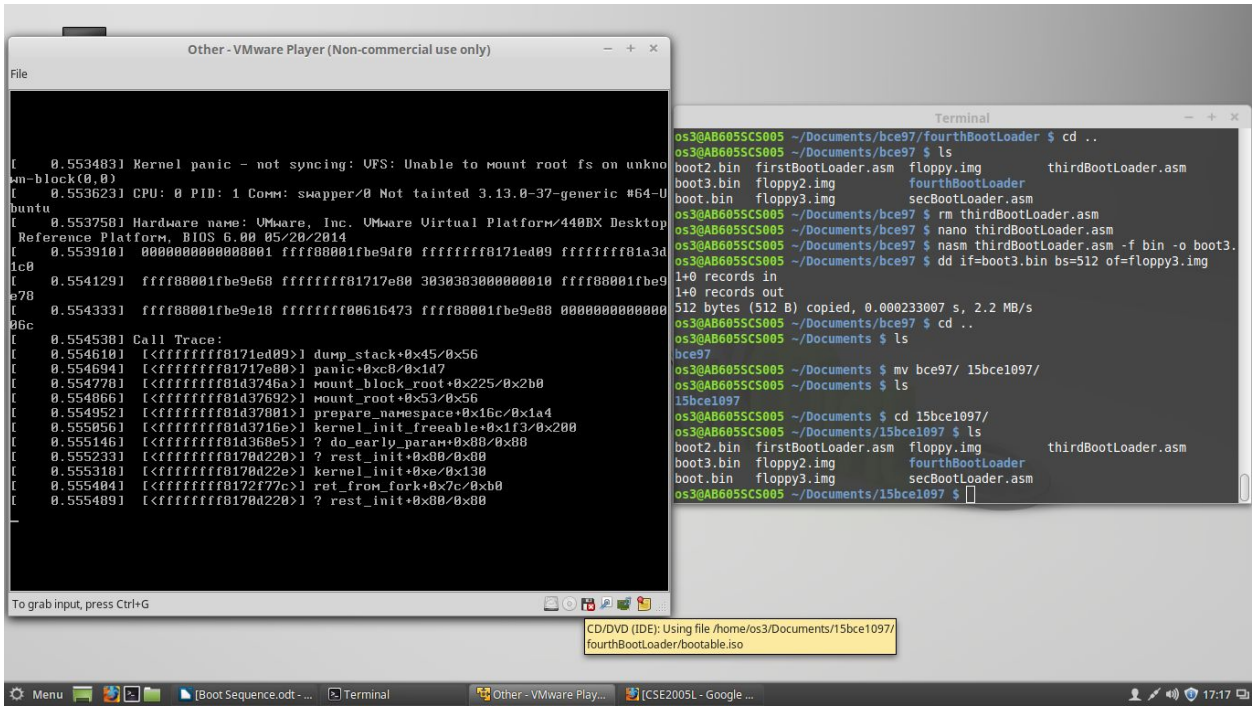
#title Boot from hard disk
#chainloader (hd0)+1

title My kernel
kernel /boot/vmlinuz-3.13.0-37-generic # Edit it to the filename of your kernel.

Directory Structure
isofiles/
`-- boot
    |-- grub
    | |-- menu.lst
    | `-- stage2_eltorito
    `-- vmlinuz-3.13.0-37-generic

2 directories, 3 files




Other - VMware Player (Non-commercial use only)
File
[0.553483] Kernel panic - not syncing: VFS: Unable to mount root fs on unkno
wn-block(0,0)
[0.553623] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 3.13.0-37-generic #64-U
buntu
[0.553758] Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop
Reference Platform, BIOS 6.00 05/20/2014
[0.553910] 0000000000000001 ffff88001fbe9df0 ffffffff8171ed09 ffffffff81a3d
1c0
[0.554129] ffff88001fbe9e68 ffffffff81717e80 3030383000000010 ffff88001fbe9
e78
[0.554333] ffff88001fbe9e18 ffffffff00616473 ffff88001fbe9e88 0000000000000
06c
[0.554530] Call Trace:
[0.554610] [<ffffffff8171ed09>] dump_stack+0x45/0x56
[0.554694] [<ffffffff81717e80>] panic+0xc8/0x1d7
[0.554778] [<ffffffff81d3746a>] mount_block_root+0x225/0x2b0
[0.554866] [<ffffffff81d37692>] mount_root+0x53/0x56
[0.554952] [<ffffffff81d37801>] prepare_namespace+0x16c/0x1a4
[0.555056] [<ffffffff81d3716e>] kernel_init_freeable+0x1f3/0x200
[0.555146] [<ffffffff81d368e5>] ? do_early_param+0x88/0x88
[0.555233] [<ffffffff8170d220>] ? rest_init+0x80/0x80
[0.555318] [<ffffffff8170d22e>] kernel_init+0xe/0x130
[0.555404] [<ffffffff8172f77c>] ret_from_fork+0x7c/0xb0
[0.555489] [<ffffffff8170d220>] ? rest_init+0x80/0x80
To grab input, press Ctrl+G
Terminal
os3@AB605SCS005 ~/Documents/bce97/fourthBootLoader $ cd ..
os3@AB605SCS005 ~/Documents/bce97 $ ls
boot2.bin firstBootLoader.asm floppy.img thirdBootLoader.asm
boot3.bin floppy2.img fourthBootLoader
boot.bin floppy3.img secBootLoader.asm
os3@AB605SCS005 ~/Documents/bce97 $ rm thirdBootLoader.asm
os3@AB605SCS005 ~/Documents/bce97 $ nano thirdBootLoader.asm
os3@AB605SCS005 ~/Documents/bce97 $ nasm thirdBootLoader.asm -f bin -o boot3.
os3@AB605SCS005 ~/Documents/bce97 $ dd if=boot3.bin bs=512 of=floppy3.img
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000233007 s, 2.2 MB/s
os3@AB605SCS005 ~/Documents/bce97 $ cd ..
os3@AB605SCS005 ~/Documents $ ls
bce97
os3@AB605SCS005 ~/Documents $ mv bce97/ 15bce1097/
os3@AB605SCS005 ~/Documents $ ls
15bce1097
os3@AB605SCS005 ~/Documents $ cd 15bce1097/
os3@AB605SCS005 ~/Documents/15bce1097 $ ls
boot2.bin firstBootLoader.asm floppy.img thirdBootLoader.asm
boot3.bin floppy2.img fourthBootLoader
boot.bin floppy3.img secBootLoader.asm
os3@AB605SCS005 ~/Documents/15bce1097 $
CD/DVD (IDE): Using file /home/os3/Documents/15bce1097/
fourthBootLoader/bootable.iso
Menu
[Boot Sequence.odt - ...
Terminal
Other - VMware Play...
[CSE2005L - Google ...
17:17