

NIM : L200180112

- ```

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\USER>cd C:/os

C:\OS>setpath

C:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;C:\OS\Perl;C:\Windows;C:\Windows\System32
C:\OS>cd LAB\LAB3

C:\OS\LAB\LAB3>S

C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsrcdbg -q -f bochsrc.bxrc
0000000000i[APIC?] local apic in initializing
=====
 Bochs x86 Emulator 2.3.5
 Build from CVS snapshot, on September 16, 2007
=====
0000000000i[] reading configuration from bochsrc.bxrc
0000000000i[] installing win32 module as the Bochs GUI
0000000000i[] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1>

```

- The image shows a Windows desktop with two open command prompt windows. The left window is titled 'Bochs for Windows - Console' and shows the output of running 'bochsrc.bxrc' in a Bochs x86 Emulator 2.3.5 environment. The output includes the initialization of the local apic and the execution of the BIOS. The right window is titled 'Bochs for Windows - Display' and shows the Bochs BIOS display, including the version (0.6a 19 Aug 2006), release date (2007/09/10), and options (apmbios pcibios eltorito rombios32).

**Bochs for Windows - Console**

```
C:\OS\LAB\LAB3>.\\.\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
0000000000[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
0000000001[] reading configuration from bochsrc.bxrc
0000000001[] installing win32 module as the Bochs GUI
0000000001[] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:0000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
r08: 0x00000000:00000000 r09: 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:2> s
Next at t=1
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax ; 31c0
<bochs:3> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:0000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
r01: 0x00000000:00000000 r02: 0x00000000:00000000
r03: 0x00000000:00000000 r04: 0x00000000:00000000
r05: 0x00000000:00000000 r06: 0x00000000:00000000
r07: 0x00000000:00000000 r08: 0x00000000:00000000
r09: 0x00000000:00000000 r10: 0x00000000:00000000
r11: 0x00000000:00000000 r12: 0x00000000:00000000
r13: 0x00000000:00000000 r14: 0x00000000:00000000
r15: 0x00000000:00000000
rip: 0x00000000:0000e05b
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:4> vb 0x07c00
<bochs:5> c
(0x0007c0) breakpoint 10285624, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6>
```

**Bochs for Windows - Display**

Plex86/Bochs UGABios 0.6a 19 Aug 2006  
This UGA/UEFI Bios is released under the GNU LGPL

Please visit :  
· <http://bochs.sourceforge.net>  
· <http://www.nongnu.org/ugabios>

Bochs UEFI Display Adapter enabled

Bochs BIOS - build: 09/10/07  
\$Revision: 1.183 \$ \$Date: 2007/09/10 20:00:29 \$  
Options: apmbios pcibios eltorito rombios32

Booting From Floppy...

Ctrl + Lbutton + Rbutton enables mo A NUM CAPS SCRL

6. Lalu ketik “s” sebanyak 10 kali dan bandingkan dengan teks yang terdapat pada boot.asm, jika terjadi ketidakcocokan maka hentikan dengan ketik “q”

```
Bochs for Windows - Console
rdx: 0x00000000:000000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000e05b
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:4> vb 0:0x7c00
<bochs:5> c
(19301656) Breakpoint 10285624, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> s
Next at t=2082129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:7> s
Next at t=2082130
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b8c007
<bochs:8> s
Next at t=2082131
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:9> s
Next at t=2082132
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:10> s
Next at t=2082133
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
<bochs:11> s
Next at t=2082134
(0) [0x00007c48] 0000:7c48 (unk. ctxt): mov gs, ax ; 8ee8
<bochs:12> s
Next at t=2082135
(0) [0x00007c4a] 0000:7c4a (unk. ctxt): mov ax, 0x0000 ; b80000
<bochs:13> s
Next at t=2082136
(0) [0x00007c4d] 0000:7c4d (unk. ctxt): mov ss, ax ; 8ed0
<bochs:14> s
Next at t=2082137
(0) [0x00007c4f] 0000:7c4f (unk. ctxt): mov sp, 0xffff ; bcffff
<bochs:15> s
Next at t=2082138
(0) [0x00007c52] 0000:7c52 (unk. ctxt): sti ; fb
<bochs:16>
```

```
Bochs for Windows - Console
ext at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> s
ext at t=2082129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:7> s
ext at t=2082130
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b8c007
<bochs:8> s
ext at t=2082131
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:9> s
ext at t=2082132
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:10> s
ext at t=2082133
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
<bochs:11> s
ext at t=2082134
(0) [0x00007c48] 0000:7c48 (unk. ctxt): mov gs, ax ; 8ee8
<bochs:12> s
ext at t=2082135
(0) [0x00007c4a] 0000:7c4a (unk. ctxt): mov ax, 0x0000 ; b80000
<bochs:13> s
ext at t=2082136
(0) [0x00007c4d] 0000:7c4d (unk. ctxt): mov ss, ax ; 8ed0
<bochs:14> s
ext at t=2082137
(0) [0x00007c4f] 0000:7c4f (unk. ctxt): mov sp, 0xffff ; bcffff
<bochs:15> s
ext at t=2082138
(0) [0x00007c52] 0000:7c52 (unk. ctxt): sti ; fb
<bochs:16>
```

```
boot - Notepad
File Edit Format View Help
;(3) Blok BOOT CODE
;=====
START:
; Mengatur lokasi kode program pada alamat 7C00:0000, dan mengatur REGISTER SEGMENT
cli ; matikan aktifitas interupsi
mov ax, 0x07C0
mov ds, ax
mov es, ax
mov fs, ax
mov gs, ax

; Mengatur lokasi stack
mov ax, 0x0000
mov ss, ax
mov sp, 0xFFFF ; sp bergerak dari alamat atas ke bawah
sti ; aktifkan aktifitas interupsi

; Menampilkan text di layar
mov si, msgLoading ; mengambil lokasi text yang di simpan dalam 'msg'
call DisplayMessage

LOAD_ROOT:
; menghitung ukuran 'root directory' dan menyimpannya dalam register 'cx'
xor cx, cx
xor dx, dx
mov ax, 0x0020 ; Ukuran satu nama direktori sepanjang 32 byt
mul WORD [MaxRootEntries] ; Total lokasi direktori 32 x 224
div WORD [BytesPerSector] ; lokasi sektor yang digunakan untuk
xchg ax, cx ; Ukuran 'root direktori' = 1

; menghitung jumlah sektor yang digunakan untuk menyimpan FAT
; untuk mencari lokasi awal sektor ROOT DIREKTORI di simpan di register 'ax'
; hasil disimpan pada variabel 'datasector'
mov al, BYTE [TotalFATs] ; jumlah FAT (2 copy)
mul WORD [SectorsPerFAT] ; dikalikan dengan jumlah sektor ya
```

7. Kemudian ketik “q” untuk menghentikan ‘debugging’
8. Lalu ketik “s” untuk memulai dari awal
9. Lalu ketik “vb 0x0100:0x0000” untuk menghentikan langkah saat pc mulai mengeksekusi instruksi dari program “kernel.bin”
10. kemudian ketik “c” untuk melanjutkan pekerjaan
11. lalu ketik “s” pada step pertama dan lakukan step berikutnya dengan menekan enter sebanyak 9x

```

Bochs for Windows - Console
<bochs:13> s
Next at t=2082136
(0) [0x0007c4d] 0000:7c4d (unk. ctxt): mov ss, ax ; 8ed0
<bochs:14> s
Next at t=2082137
(0) [0x0007c4f] 0000:7c4f (unk. ctxt): mov sp, 0xffff ; bcffff
<bochs:15> s
Next at t=2082138
(0) [0x0007c52] 0000:7c52 (unk. ctxt): sti ; fb
<bochs:16> q
In bx_win32_gui.c::exit(void)!
Bochs is exiting. Press ENTER when you're ready to close this window.

C:\OS\LAB\LAB3>s
C:\OS\LAB\LAB3>.\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
000000000000i[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000000i[] reading configuration from bochsrc.bxrc
000000000000i[] installing win32 module as the Bochs GUI
000000000000i[] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> vb 0x0100:0x0000
<bochs:2> c
(13075736) Breakpoint 10285624, in 0100:0000 (0x00001000)
Next at t=2945013
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100 ; b80001
<bochs:3> s
Next at t=2945014
(0) [0x00001003] 0100:0003 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:4>
Next at t=2945015
(0) [0x00001005] 0100:0005 (unk. ctxt): mov es, ax ; 8ec0
<bochs:5>
Next at t=2945016
(0) [0x00001007] 0100:0007 (unk. ctxt): cli ; fa
<bochs:6>
Next at t=2945017
(0) [0x00001008] 0100:0008 (unk. ctxt): mov ss, ax ; 8ed0
<bochs:7>

```

12. kemudian bandingkan source-code yang terdapat di layar dengan yang terdapt pada program ‘kernel.asm’

```

Bochs for Windows - Console
b) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
bochs:1> vb 0x0100:0x0000
bochs:2> c
(13075736) Breakpoint 10285624, in 0100:0000 (0x00001000)
ext at t=2945013
b) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100 ; b80001
bochs:3> s
ext at t=2945014
b) [0x00001003] 0100:0003 (unk. ctxt): mov ds, ax ; 8ed8
bochs:4>
ext at t=2945015
b) [0x00001005] 0100:0005 (unk. ctxt): mov es, ax ; 8ec0
bochs:5>
ext at t=2945016
b) [0x00001007] 0100:0007 (unk. ctxt): cli ; fa
bochs:6>
ext at t=2945017
b) [0x00001008] 0100:0008 (unk. ctxt): mov ss, ax ; 8ed0
bochs:7>
ext at t=2945018
b) [0x0000100a] 0100:000a (unk. ctxt): mov sp, 0xffff ; bcffff
bochs:8>
ext at t=2945019
b) [0x0000100d] 0100:000d (unk. ctxt): sti ; fb
bochs:9>
ext at t=2945020
b) [0x0000100e] 0100:000e (unk. ctxt): push dx ; 52
bochs:10>
ext at t=2945021
b) [0x0000100f] 0100:000f (unk. ctxt): push es ; 06
bochs:11>
ext at t=2945022
b) [0x00001010] 0100:0010 (unk. ctxt): xor ax, ax ; 31c0
bochs:12>
ext at t=2945023
b) [0x00001012] 0100:0012 (unk. ctxt): mov es, ax ; 8ec0
bochs:13>

kernel - Notepad
File Edit Format View Help
;
; Prototype SIMPLE KERNEL ver 0.01
; LAB-INFORMATIKA
; =====
[org 0x000]
[bits 16]

[SEGMENT .text]

;START #####
mov ax, 0x0100 ;lokasi memori untuk menempatkan kernel
mov ds, ax
mov es, ax

cli ;set interrupt OFF
mov ss, ax ;atur stack segment
mov sp, 0xFFFF ;atur stack pointer maksimum 64k
sti ;set interrupt ON

push dx
push es
xor ax, ax
mov es, ax
cli
mov word [es:0x21*4], _int0x21 ; setup interrupt service
mov [es:0x21*4+2], cs ; untuk menampilkan karakter di layar
sti
pop es
pop dx

mov si, strWelcomeMsg ; Tampilkan informasi proses
mov al, 0x01 ; request service 0x01
int 0x21 ; int 0x21

call _shell ; call the shell

int 0x19 ; reboot
;END #####

```

## TUGAS

### 1. Tabel pemetaan memori pada PC

| No | Blok Memori | Alokasi Pemakaian               |
|----|-------------|---------------------------------|
| 1  | F 0 0 0 0   | ROM BIOS, Diagnostic, BASIC     |
| 2  | E 0 0 0 0   | ROM program                     |
| 3  | D 0 0 0 0   | ROM program                     |
| 4  | C 0 0 0 0   | Perluasan BIOS untuk hardisk XT |
| 5  | B 0 0 0 0   | Monokrom Monitor                |
| 6  | A 0 0 0 0   | Monitor EGA, VGS, dll           |
| 7  | 9 0 0 0 0   | Daerah kerja pemakai s/d 640 KB |
| 8  | 8 0 0 0 0   | Daerah kerja pemakai s/d 576 KB |
| 9  | 7 0 0 0 0   | Daerah kerja pemakai s/d 512 KB |
| 10 | 6 0 0 0 0   | Daerah kerja pemakai s/d 448 KB |
| 11 | 5 0 0 0 0   | Daerah kerja pemakai s/d 384 KB |
| 12 | 4 0 0 0 0   | Daerah kerja pemakai s/d 320 KB |
| 13 | 3 0 0 0 0   | Daerah kerja pemakai s/d 256 KB |
| 14 | 2 0 0 0 0   | Daerah kerja pemakai s/d 192 KB |
| 15 | 1 0 0 0 0   | Daerah kerja pemakai s/d 128 KB |
| 16 | 0 0 0 0 0   | Daerah kerja pemakai s/d 64 KB  |

### 2. Perbedaan antara mode kerja 'Real-Mode' dan mode kerja 'Protect-Mode' pada PC IBM Compatible

#### a) Real-Mode

Real-Mode adalah sebuah modus di mana prosesor Intel x86 berjalan seolah-olah dirinya adalah sebuah prosesor Intel 8085 atau Intel 8088, meski ia merupakan prosesor Intel 80286 atau lebih tinggi. Karenanya, modus ini juga disebut sebagai modus 8086 (8086 Mode). Dalam modus ini, prosesor hanya dapat mengeksekusi instruksi 16-bit saja dengan menggunakan register internal yang berukuran 16-bit, serta hanya dapat mengakses hanya 1024 KB dari memori karena hanya menggunakan 20-bit jalur bus alamat. Semua program DOS berjalan pada modus ini.

Prosesor yang dirilis setelah 8085, semacam Intel 80286 juga dapat menjalankan instruksi 16-bit, tapi jauh lebih cepat dibandingkan 8085. Dengan kata lain, Intel 80286 benar-benar kompatibel dengan prosesor Intel 8086 yang didesain sebelumnya. Sehingga prosesor Intel 80286 pun dapat menjalankan program-program 16-bit yang didesain untuk 8085 (IBM PC), dengan tentunya kecepatan yang jauh lebih tinggi.

Dalam Real-mode, tidak ada proteksi ruang alamat memori, sehingga tidak dapat melakukan multi-tasking. Inilah sebabnya, mengapa program-program DOS bersifat single-tasking. Jika dalam modus real terdapat multi-tasking, maka kemungkinan besar antara dua program yang sedang berjalan, terjadi tabrakan (crash) antara satu dengan lainnya.

#### b) Protect-Mode

Modus terproteksi (protected mode) adalah sebuah modus di mana terdapat proteksi ruang alamat memori yang ditawarkan oleh mikroprosesor untuk digunakan oleh sistem operasi. Modus ini datang dengan mikroprosesor Intel 80286 atau yang lebih tinggi. Karena memiliki proteksi ruang alamat memori, maka dalam modus ini sistem operasi dapat melakukan multitasking.

Prosesor Intel 80286 memang dilengkapi kemampuan masuk ke dalam modus terproteksi, tapi tidak dapat keluar dari modus tersebut tanpa harus mengalami reset (warm boot atau

cold boot). Kesalahan ini telah diperbaiki oleh Intel dengan merilis prosesor Intel 80386 yang dapat masuk ke dalam modus terproteksi dan keluar darinya tanpa harus melakukan reset. Inilah sebabnya mengapa Windows 95/Windows 98 dilengkapi dengan modus Restart in MS-DOS Mode, meski sebenarnya sistem operasi tersebut merupakan sistem operasi yang berjalan dalam modus terproteksi.