

LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM SISTEM OPERASI
**MODUL 3 MENGENAL CARA DEBUGGING PROGRAM BOOTSTRAP-
LOADER**



Disusun Oleh :
AFIFAH NUR NABILA
L200210249
Kelas E

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2022/2023

Laporan Praktikum Modul 3

| | |
|---|---|
| NIM : L200210249 Nama : Afifah Nur Nabila Dosen Pengampu : Heru Setiya Nugraha, S.T, M.Kom Tanggal Praktikum : 27 September 2022 | Nilai praktek : Tanda tangan : |
|---|---|

Langkah Kerja

1. Masuk ke direktori kerja 'C:\OS\LAB\LAB3'

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Person>cd /

C:\>cd 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>
```

2. Proses debugging. Ketikkan 'type s.bat'

```
C:\OS\LAB\LAB3>type s.bat
...\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc

C:\OS\LAB\LAB3>
```

3. Masukkan perintah 'S'

```
C:\OS\LAB\LAB3>s

C:\OS\LAB\LAB3>...\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
00000000000i[APIC?] local apic in initializing
=====
                Bochs x86 Emulator 2.3.5
                Build from CVS snapshot, on September 16, 2007
=====
00000000000i[      ] reading configuration from bochsrc.bxrc
00000000000i[      ] installing win32 module as the Bochs GUI
00000000000i[      ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b      ; ea5be000f0
<bochs:1>
```

Layar pada PC-Simulator terlihat gelap



4. Melihat isi register CS dan IP dengan perintah 'r'

```
C:\OS\LAB\LAB3>s
C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsdbg -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:fff0 (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
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: 0x00000000:0000fff0
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:2>
```

5. Mengeksekusi perintah, ketikkan 's' kemudian perintah 'r'

```
Bochs for Windows - Console
Next at t=0
(0) [0xffffffff] 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
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: 0x00000000:0000fff0
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
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>
```

6. 'break point' dengan mengetikkan perintah 'vb 0:0x7C00' dan selanjutnya masukkan perintah 'c'

```
<bochs:4> vb 0:0x7C00
<bochs:5> c
(10264512) Breakpoint 10285624, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6>
```

PC Simulator menampilkan teks 'Booting from Floppy . . .'

```
Bochs for Windows - Display
Plex86/Bochs UGABios 0.6a 19 Aug 2006
This UGA/UBE Bios is released under the GNU LGPL

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

Bochs UBE 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 + 3rd button enables mouse
```

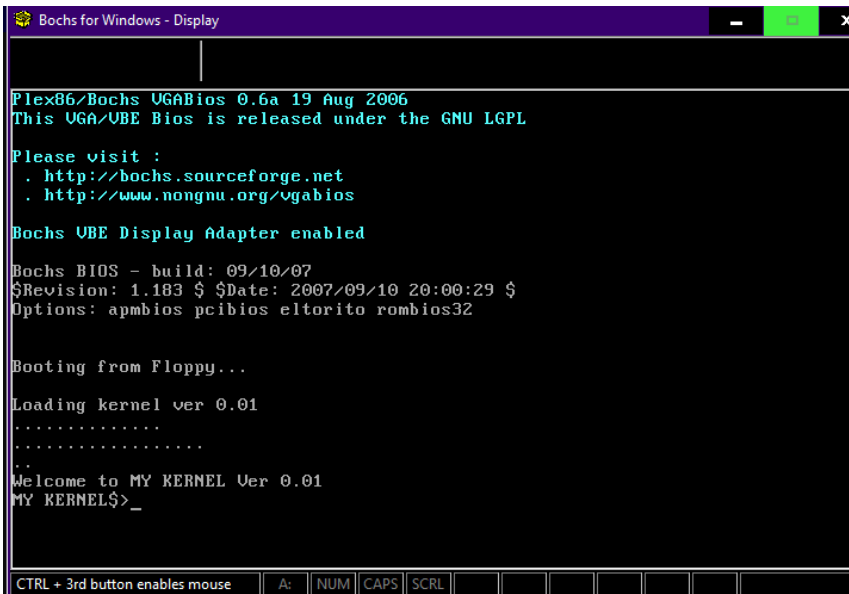
7. Jalankan PC langkah demi langkah dengan perintah 's', kemudian hentikan proses dengan perintah 'q'

```
<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> 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>
```

8. PC Simulator menjalankan program 'kernel.bin'



The screenshot shows a window titled "Bochs for Windows - Display". The text inside the window is as follows:

```
Plex86/Bochs VGABios 0.6a 19 Aug 2006
This UGA/UBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs UBE 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...

Loading kernel ver 0.01
.....
..
Welcome to MY KERNEL Ver 0.01
MY KERNEL$>_

CTRL + 3rd button enables mouse  A: NUM CAPS SCRL
```

9. Menghentikan PC Simulator saat menjalankan program 'kernel.bin'. Mulai dari awal dengan perintah 's' dan buat 'break point' dengan perintah 'vb 0x0100:0x0000'. Kemudian ketikkan perintah 'c'

```
C:\OS\LAB\LAB3>s
C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsdbg -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:fff0 (unk. ctxt): jmp far f000:e05b      ; ea5be00f0
<bochs:1> vb 0x0100:0x0000
<bochs:2> c
(10264512) Breakpoint 10285624, in 0100:0000 (0x00001000)
Next at t=2945013
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100      ; b80001
<bochs:3> s
```

10. Teruskan langkah PC Simulator dengan ketik 's' sebanyak 10 kali, step berikutnya dapat dilakukan dengan cara menekan tombol ENTER secara langsung

```
Bochs for Windows - Console
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> s
Next at t=2945015
(0) [0x00001005] 0100:0005 (unk. ctxt): mov es, ax          ; 8ec0
<bochs:5> s
Next at t=2945016
(0) [0x00001007] 0100:0007 (unk. ctxt): cli                  ; fa
<bochs:6> s
Next at t=2945017
(0) [0x00001008] 0100:0008 (unk. ctxt): mov ss, ax          ; 8ed0
<bochs:7> s
Next at t=2945018
(0) [0x0000100a] 0100:000a (unk. ctxt): mov sp, 0xffff      ; bcffff
<bochs:8> s
Next at t=2945019
(0) [0x0000100d] 0100:000d (unk. ctxt): sti                  ; fb
<bochs:9> s
Next at t=2945020
(0) [0x0000100e] 0100:000e (unk. ctxt): push dx             ; 52
<bochs:10> s
Next at t=2945021
(0) [0x0000100f] 0100:000f (unk. ctxt): push es             ; 06
<bochs:11> s
Next at t=2945022
(0) [0x00001010] 0100:0010 (unk. ctxt): xor ax, ax          ; 31c0
<bochs:12> s
```

TUGAS

1. Buatlah tabel pemetaan memori pada PC selengkap mungkin.

| Blok Memori | Alokasi Pemakaian |
|-------------|----------------------------------|
| F 0 0 0 0 | ROM BIOS, Diagnostic, BASIC |
| E 0 0 0 0 | ROM program |
| D 0 0 0 0 | ROM program |
| C 0 0 0 0 | Perluasan BIOS untuk harddisk XT |
| B 0 0 0 0 | Monokrom Monitor |
| A 0 0 0 0 | Monitor EGA, VGS, dll |
| 9 0 0 0 0 | Daerah kerja pemakai s/d 640 KB |
| 8 0 0 0 0 | Daerah kerja pemakai s/d 576 KB |
| 7 0 0 0 0 | Daerah kerja pemakai s/d 512 KB |
| 6 0 0 0 0 | Daerah kerja pemakai s/d 448 KB |
| 5 0 0 0 0 | Daerah kerja pemakai s/d 384 KB |
| 4 0 0 0 0 | Daerah kerja pemakai s/d 320 KB |
| 3 0 0 0 0 | Daerah kerja pemakai s/d 256 KB |
| 2 0 0 0 0 | Daerah kerja pemakai s/d 192 KB |
| 1 0 0 0 0 | Daerah kerja pemakai s/d 128 KB |
| 0 0 0 0 0 | Daerah kerja pemakai s/d 64 KB |

2. Baca buku referensi, jelaskan perbedaan antara mode kerja 'Real-Mode' dan mode kerja 'Protect-Mode' pada PC IBM Compatible.

⇒ Real-Mode

- Didasarkan pada prosesor 8085 & 8088
- Prosesor hanya dapat mengeksekusi instruksi 16-bit saja dengan menggunakan register internal yang berukuran 16-bit
- Hanya dapat mengakses 1024 KB dari memori karena hanya menggunakan 20-bit jalur bus alamat
- Menjalankan semua program DOS
- Tidak ada proteksi ruang alamat memori, sehingga tidak dapat melakukan multi-tasking

⇒ Protect-Mode

- Dimulai dengan chip 80286 di IBM AT, protect mode baru diperkenalkan
- Terdapat proteksi ruang alamat memori yang ditawarkan oleh mikroprosesor untuk digunakan oleh system operasi, sehingga dapat melakukan multi-tasking
- Tidak ada batas 1 MB dalam protect mode
- Memiliki mikroprosesor Intel 80286 atau yang lebih tinggi
- Semua sistem operasi utama saat ini menggunakan protect mode termasuk Windows 3.x, Window 9x, Windows NT, OS / 2 dan Linux.