

PRAKTIKUM SISTEM OPERASI
MENGENAL CARA ‘DEBUGGING’ PROGRAM BOOTSTRAP-LOADER



Oleh :
Dian Putri Mutiara Hapsari
L200210238

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2021

LANGKAH KERJA

A. Langkah 1

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\LABRPL-21>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>
```

Melakukan setpath di direktori kerja OS dan menuju ke lab3

B. Langkah 2

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

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

Memasukkan perintah "type s.bat"

C. Langkah 3

```
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          ; ea5be000f0
<bochs:1>
```



Melakukan debugging dengan memasukkan perintah 'S' lalu akan ditampilkan jendela PC Simulator

D. Langkah 4

```
Next at t=0
<0> [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b      ; ea5be000f0
<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>
```

Memasukkan perintah 'r' untuk melihat isi register CS dan IP

E. Langkah 5

```
<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>
```

Memasukkan perintah 's', kemudian setelahnya perintah 'r' dimasukkan untuk melihat isi direktori

F. Langkah 6

```
<bochs:4> vb 0:0x7C00
<bochs:5> c
<0> Breakpoint 2683464, in 0000:7c00 (0x00007c00)
Next at t=2082128
<0> [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6>
```

Membuat break point pada alamat 0000:7c000 dan memasukkan perintah "c" untuk melanjutkan proses ke titik pemberhentian.

G. Langkah 7

```
<bochs:4> vb 0:0x7C00
<bochs:5> c
<0> Breakpoint 2683464, 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> q
# In bx_win32_gui_c::exit(void)?

Bochs is exiting. Press ENTER when you're ready to close this window.
```

Mengulang proses penulisan perintah “s”

H. Langkah 8

```
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:fff0 <unk. ctxt>: jmp far f000:e05b ; ea5be000f0
```

Setelah memberhentikan proses, dilakukan pengulangan dan diawali dengan memasukkan perintah “S”

I. Langkah 9

```
<bochs:1> vb 0x0100:0x0000
<bochs:2> c
<0> Breakpoint 2683464, in 0100:0000 <0x00001000>
Next at t=2945013
<0> [0x00001000] 0100:0000 <unk. ctxt>: mov ax, 0x0100 ; b80001
```

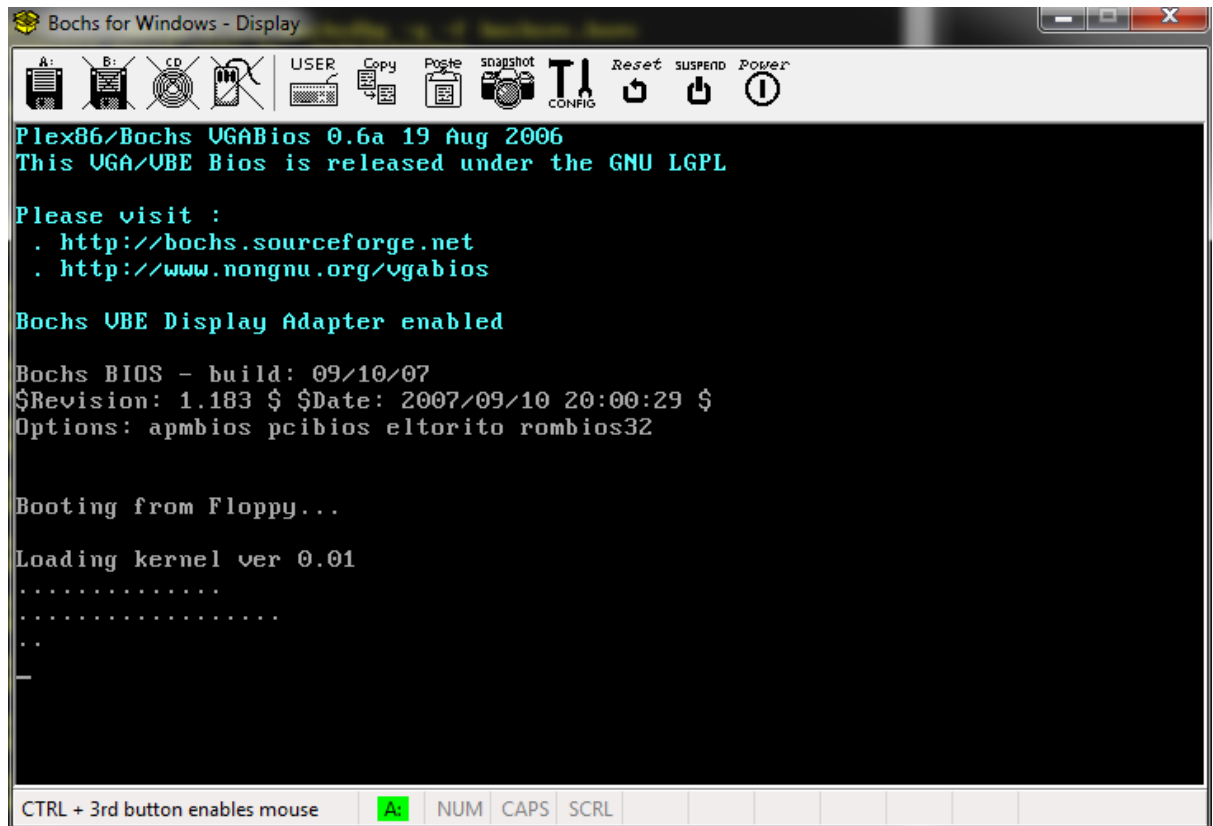
Membuat break point dan memasukkan perintah “c” agar melanjutkan proses ke titik pemberhentian

J. Langkah 10

```
<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
Next at t=2945023
<0> [0x00001012] 0100:0012 <unk. ctxt>: mov es, ax ; 8ec0
<bochs:13>
Next at t=2945024
<0> [0x00001014] 0100:0014 <unk. ctxt>: cli ; fa
<bochs:14>
Next at t=2945025
<0> [0x00001015] 0100:0015 <unk. ctxt>: mov word ptr es:0x84, 0x0030 ; 26c706840
03000
<bochs:15>
```

Pengulangan memasukkan perintah “S” sebanyak 10 kali dan setelahnya menekan tombol Enter secara langsung

K. Tampilan PC Simulator setelah langkah 10 selesai dijalankan



TUGAS

1. Buatlah tabel pemetaan memori pada PC selengkap mungkin.

Jawab :

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.

Jawab :

Real-Mode	Protected-Mode
Bersifat single-tasking	Bersifat multi-tasking
Tidak ada proteksi ruang alamat memori	Terdapat proteksi ruang alamat memori.
Hanya dapat menjalankan program-program 16-bit	Dapat menjalankan program-program 16-bit dan 32-bit
Menggunakan BIOS subroutines dan OS subroutines	Hanya menggunakan OS subroutines