LAPORAN PRAKTIKUM SISTEM OPERASI
MODUL 3 : MENGENAL CARA 'DEBUGGING' PROGRAM BOOTSTRAP-LOADER

DISUSUN OLEH :

NAMA : BIMA TRIADMAJA

NIM : L200210137

KELAS : C

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2022/2023

Berikut adalah screenshot dari percobaan yang telah saya lakukan :

1. Menuju direktori kerja OS melalui Command Prompt, selanjutnya ‘setpath’, lalu masuk ke direktoriLAB3 serta melakukan pengecekan.
2. File sudah disiapkan, selanjutnya adalah melakukan proses ‘debugging’ dengan ketik ‘type s.bat’.
3. Mulai melakukan debugging dengan memasukkan perintah ‘S’.

```
Bochs for Windows - Console
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Infinity>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>type s.bat
..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc

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) [0xfffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b        ; ea5be000f0
<bochs:1> r
```

4. Selanjutnya memasukkan perintah ‘r’ dan akan ditampilkan seperti berikut :

```
<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: 0x00000000:00000f20 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
```

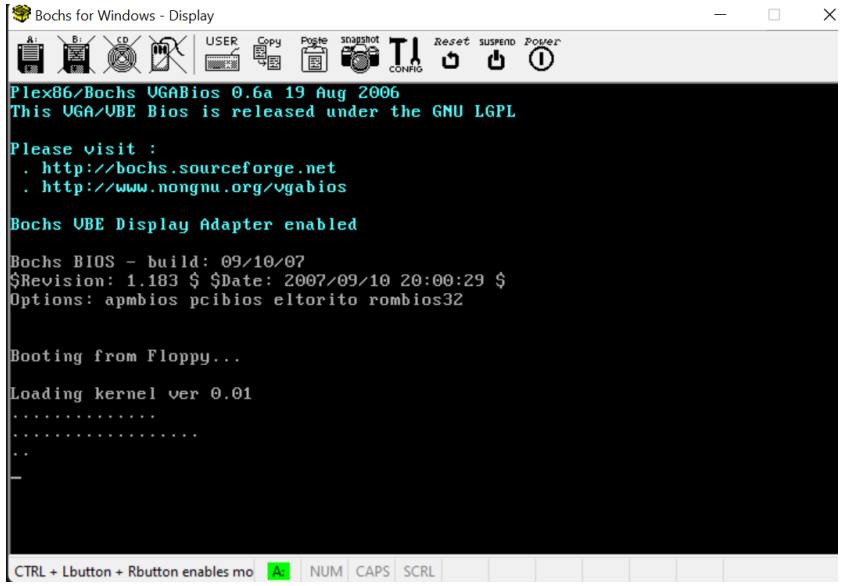
5. Selanjutnya kita menyuruh PC untuk mengeksekusi perintah dengan mengetik ‘s’ kemudian dilanjut perintah ‘r’, maka akan ditampilkan seperti berikut :

```
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 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
```

6. Memberikan sebuah sinyal berhenti yang disebut ‘break point’ dengan memasukkan perintah ‘vb 0:0x7C00’ kemudian dilanjutkan memasukkan perintah ‘c’.

```
(0) [0xfffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b         ; ea5be000f0
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 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> s
Next at t=2
(0) [0x000fe05d] f000:e05d (unk. ctxt): out 0x0d, al              ; e60d
<bochs:4> s
Next at t=3
(0) [0x000fe05f] f000:e05f (unk. ctxt): out 0xda, al              ; e6da
<bochs:5> vb 0:0x7C00
<bochs:6> c
(2002577225) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:7> s
```

Pada PC Simulator mulai ditampilkan seperti ini :



7. Sekarang PC mulai memasuki tahapan ‘BOOTSTRAP-LOADER’, langkah selanjutnya menjalankan PC langkah demi langkah (debugging) dengan perintah ‘s’, dan jika terjadi kesalahan dapat diberhentikan dengan mengetik ‘q’.

```
(2002577225) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:7> s
Next at t=2082129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli                       ; fa
<bochs:8> s
Next at t=2082130
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0            ; b8c007
<bochs:9> s
Next at t=2082131
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax                ; 8ed8
<bochs:10> q
```

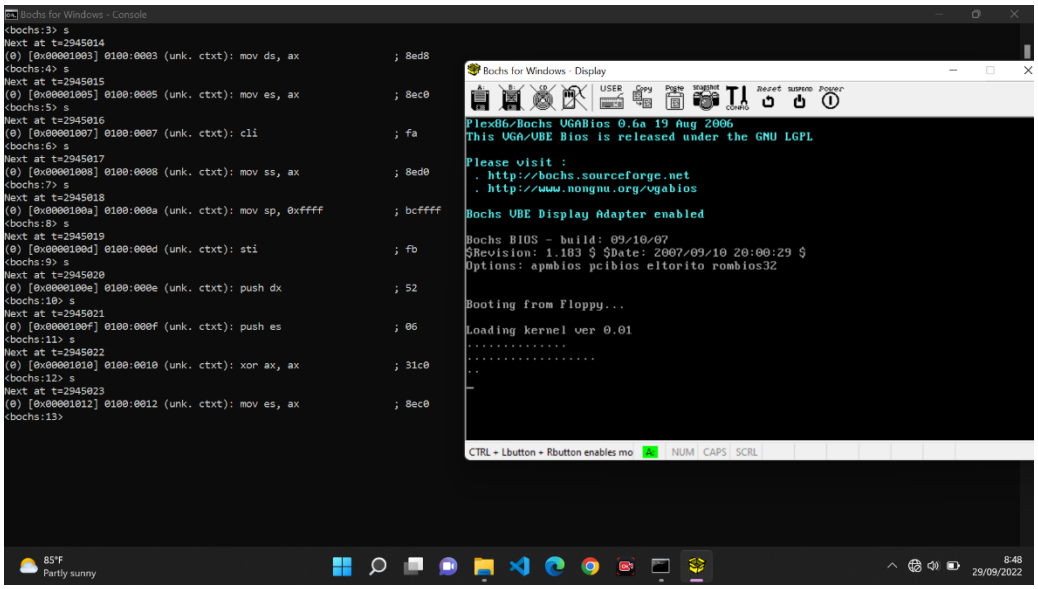
8. Selanjutnya memerintahkan PC Simulator untuk melanjutkan pekerjaannya, yaitu dengan menambahkan ‘break point’ maksimal 7.

9. Menghentikan PC Simulator pada saat ‘debugging’ sebelumnya dengan perintah ‘q’ kemudian memasukkan perintah ‘vb 0x0100:0x0000’, maka akan ditampilkan seperti berikut :

```
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) [0xfffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b         ; ea5be000f0
<bochs:1> vb 0x0100:0x0000
<bochs:2> c
(10264512) Breakpoint 10285608, in 0100:0000 (0x00001000)
Next at t=2945013
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100            ; b80001
<bochs:3> s
```

10. Selanjutnya meneruskan langkah PC Simulator step by step dengan mengetikkan ‘s’ minimal sebanyak 10x.



Tugas

1. Buatlah tabel pemetaan memori pada PC selengkap mungkin.
   Jawab :
   Saya akan memberikan 2 contoh :
   - ❖ Contoh 1 :

Tabel Peta Memori Pada IBM PC

Blok Memori	AlokasiPemakaian
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 untukhardisk XT
B 0 0 0 0	Monokrom Monitor
A 0 0 0 0	Monitor EGA, VGS, dll
9 0 0 0 0	Daerah kerjapemakai s/d 640 KB
8 0 0 0 0	Daerah kerjapemakai s/d 576 KB
7 0 0 0 0	Daerah kerjapemakai s/d 512 KB
6 0 0 0 0	Daerah kerjapemakai s/d 448 KB
5 0 0 0 0	Daerah kerjapemakai s/d 384 KB
4 0 0 0 0	Daerah kerjapemakai s/d 320 KB
3 0 0 0 0	Daerah kerjapemakai s/d 256 KB
2 0 0 0 0	Daerah kerjapemakai s/d 192 KB
1 0 0 0 0	Daerah kerjapemakai s/d 128 KB
0 0 0 0 0	Daerah kerjapemakai s/d 64 KB

   - ❖ Contoh 2 :

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Daerah Memori
32.768	16.384	8.192	4.096	2.048	1.024	512	256	128	64	32	16	8	4	2	1	(Alamat)
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 H awal EPROM
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH akhir EPROM
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H awal RAM
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	17FFH akhir RAM

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

- Protect Mode

  Protect Mode (modus terproteksi) 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.