

**PRAKTIKUM SISTEM OPERASI
MODUL 3
MENGENAL CARA ‘Debugging’ PROGRAM BOOTSRAP-LOADER**

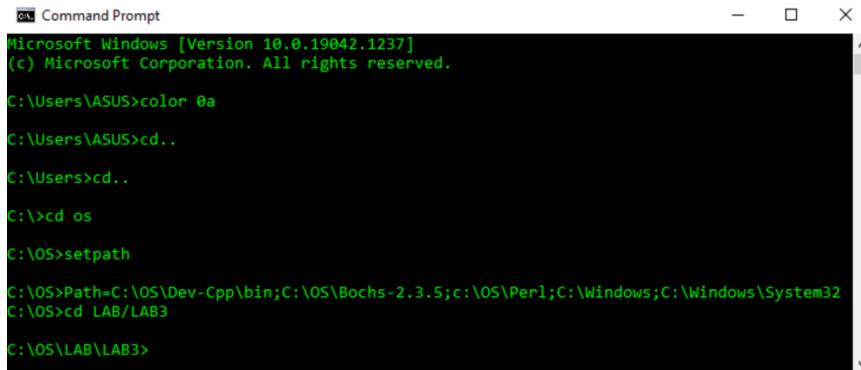


**DISUSUN OLEH:
ILHAM RIAN NOVANTO
L200200247**

**INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2021**

Langkah Kerja

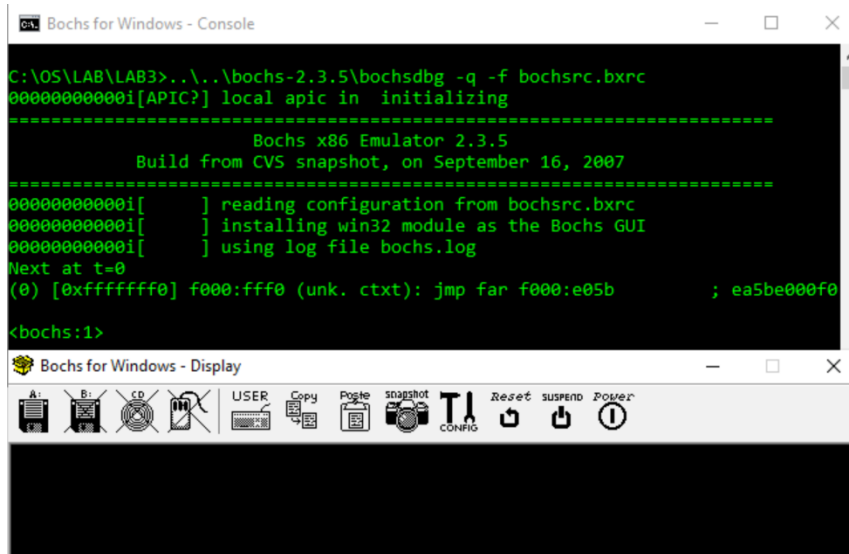
1. Membuka CMD lalu lanjutkan dengan 'CD OS', 'setpath' dan 'cd LAB/LAB3'.



```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>color 0a
C:\Users\ASUS>cd..
C:\Users>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>
```

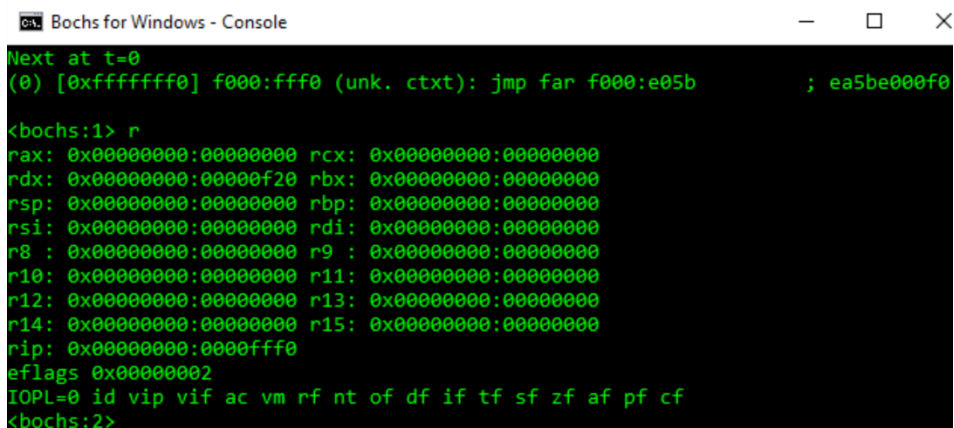
2. Melakukan proses 'debugging' masukan perintah 'S'.



```
Bochs for Windows - Console
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      ; ea5be000f0
<bochs:1>

Bochs for Windows - Display
[Icons: A1, B1, CD, USER, Copy, Paste, snapshot, CONFIG, Reset, suspend, Power]
```

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



```
Bochs for Windows - Console
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>
```

4. PC akan mengeksekusi program yang berada di lokasi pada 0000fff0 dengan perintah 's' kemudian lanjutkan dengan perintah 'r'.

```

Select Bochs for Windows - Console
<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>
```

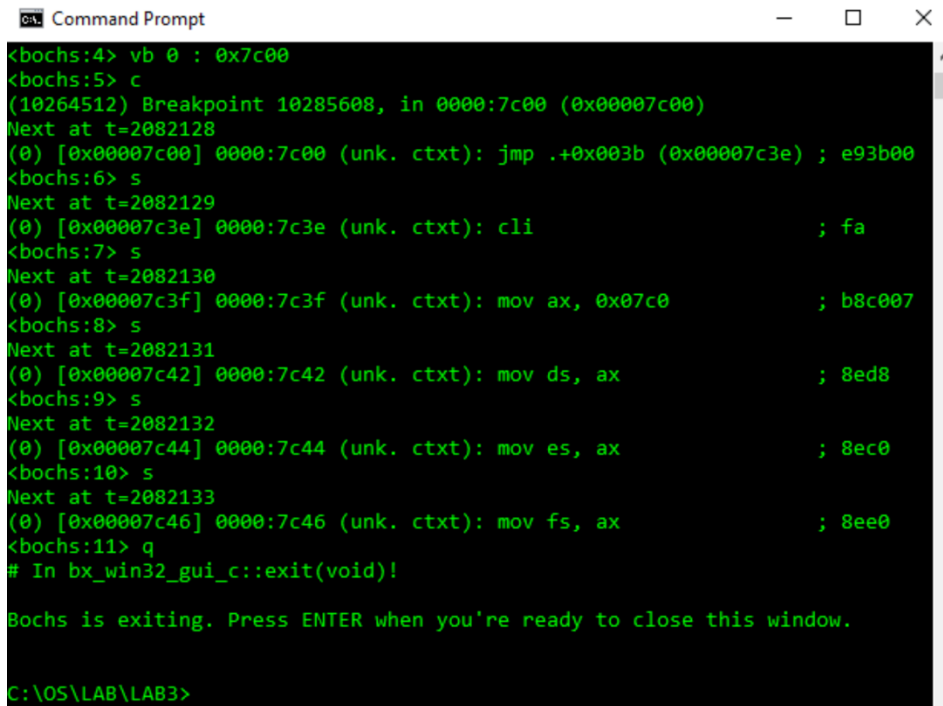
5. Masukkan perintah berikut 'vb 0:0x7C00' <ENTER> Maksud perintah ini adalah membuat titik pemberhentian (halte) pada alamat 0000:7C00.
Masukkan perintah 'c' Maksud perintah ini adalah teruskan (Continue) prosesnya sampai ke titik pemberhentian. Dalam sekejap PC sudah sampai pada pemberhentian yang kita buat di atas yaitu pada alamat 0000:7C00

C mulai memasuki tahapan 'BOOTSTRAPLOADER', untuk sampai pada tahap ini PC sudah menghabiskan clock sebanyak '2082128' (dapat dilihat di "Next at t=2082128").

```

Bochs for Windows - Console
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
<bochs:5> c
(10264512) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2084140
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6>
```

6. Lalu sekarang PC akan mulai menjalankan program 'boot.asm'. ketikan "s" secara berulang dan "q" Untuk menghentikan debugging.

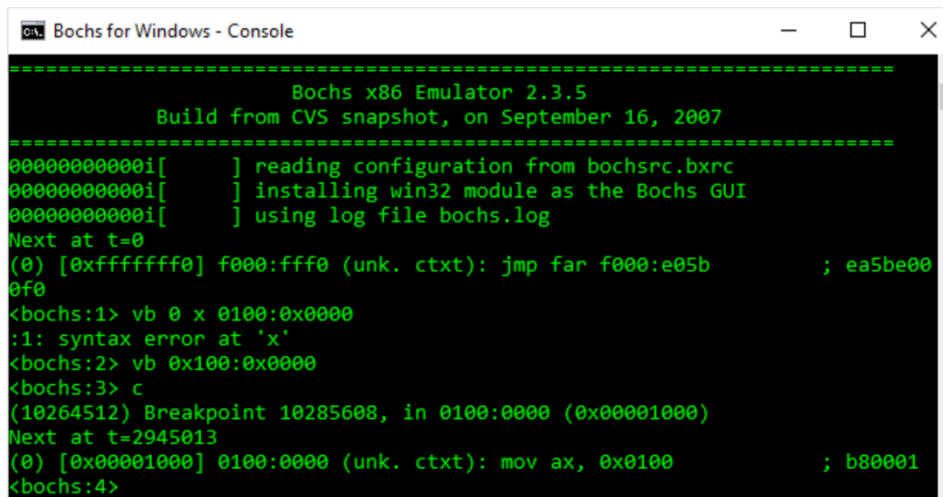


```
Command Prompt
<bochs:4> vb 0 : 0x7c00
<bochs:5> c
(10264512) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> s
Next at t=2082129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:7> s
Next 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.

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

7. Kemudian buatlah break-point, masukan perintah 'vb 0x0100:0x0000' untuk menghentikan langkah saat PC mulai mengeksekusi instruksi dari program 'kernel.bin'. Selanjutnya perintahkan PC untuk melanjutkan pekerjaan, 'c'.



```
Bochs for Windows - Console
=====
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 ; ea5be00
0f0
<bochs:1> vb 0 x 0100:0x0000
:1: syntax error at 'x'
<bochs:2> vb 0x100:0x0000
<bochs:3> c
(10264512) Breakpoint 10285608, in 0100:0000 (0x00001000)
Next at t=2945013
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100 ; b80001
<bochs:4>
```

8. Selanjutnya teruskan langkah PC Simulator step-by-step minimal sebanyak 10x, ketik 's'.

The screenshot shows two windows from the Bochs emulator. The 'Bochs for Windows - Console' window displays assembly instructions and memory addresses, such as 'Next at t=2945013', '(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100', and 'Next at t=2945014'. The 'Bochs for Windows - Display' window shows the Bochs UGA BIOS screen, which includes the text 'Plex86/Bochs UGABios 0.6a 19 Aug 2006', 'This UGA/UBE Bios is released under the GNU LGPL', and 'Bochs UBE Display Adapter enabled'. It also shows the BIOS version 'Bochs BIOS - build: 09/10/07' and options 'Options: apmbios pcibios eltorito rombios32'.

Tugas

1. Buatlah tabel pemetaan memori pada PC selengkap mungkin.

Processor address	Bank 0 (Boot)	Bank 1 (System)	Bank 2 (User1)	Bank 3 (User2)
0xFFFF	16 kB System	System Shared 8 kB	System Shared 8 kB	System Shared 8 kB
0xE000				
0xDFFF	Shared	Shared 8 kB (Bank 0,1)		
0xC000				
0xBFFF	Video Memory 8 kB			
0xA000				
	(no memory mapped-empty)	System Private 48 kB	User 56 kB	User 56 kB
0x3FFF	16 kB read only Memory (ROM)			
0x0000				

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 hardsik 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

2. Baca buku referensi, jelaskan 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. Jikadalam modus real terdapat multi-tasking, maka kemungkinan besar antara dua program yang sedang berjalan, terjaditabrakan (crash) antara satu dengan lainnya.

B. Protected 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.