

**LAPORAN DAN TUGAS
PRAKTIKUM SISTEM OPERASI
MODUL 3**



DISUSUN OLEH :
Tegar Wisnu Noviyanto
L200210247

**TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2022/2023**

LAPORAN LATIHAN PRAKTIK LANGKAH KERJA :

1. Start kemudian buka command prompt
2. Ketik cd os lalu enter
3. Ketik setpath lalu enter
4. Ketik cd lab/lab3 lalu enter
5. Ketik setpath lalu enter
6. Ketik s lalu enter untuk run debugging. Lalu muncul bochs kosong:

```
C:\>cd os\

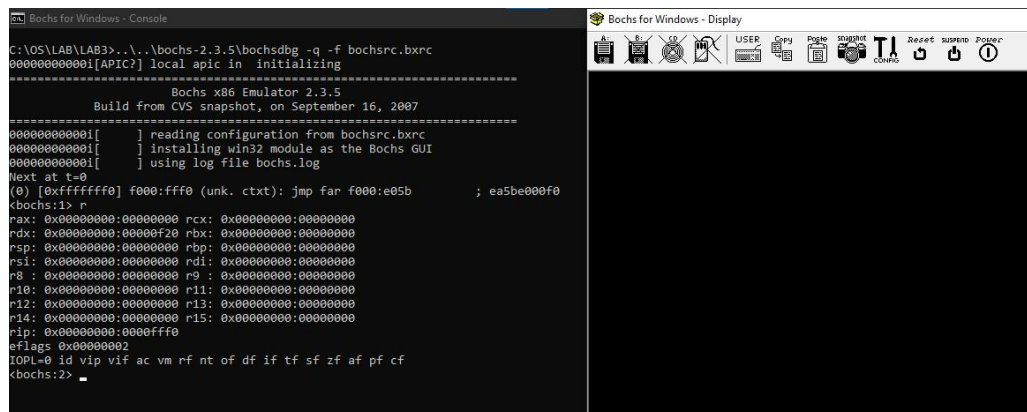
C:\OS>setpath\
'setpath\' is not recognized as an internal or external command,
operable program or batch file.

C:\OS>setpath

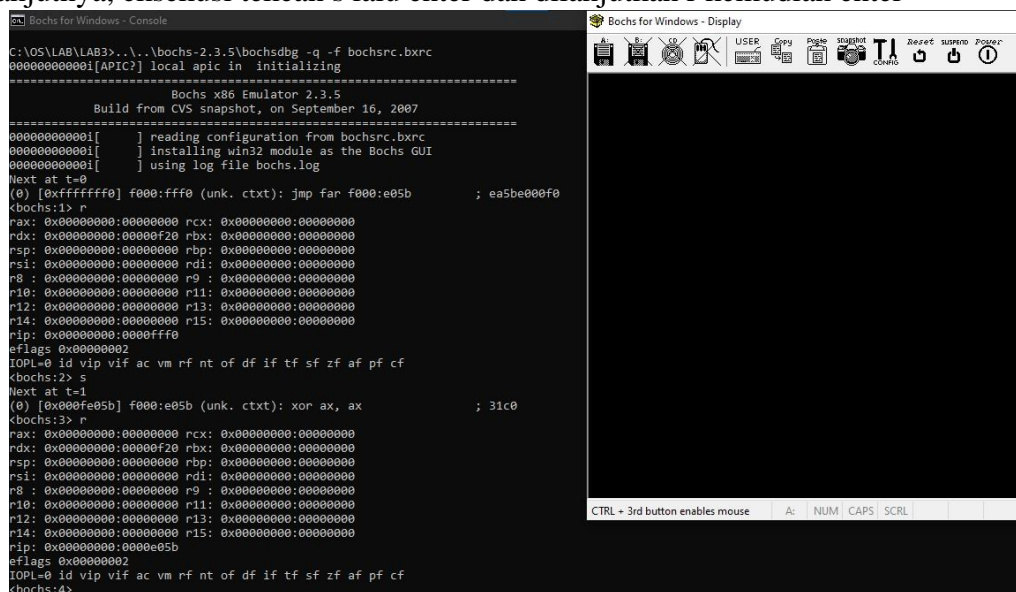
C:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;C:\Windows;C:\Windows\System32
C:\OS>cd LAB/LAB3
C:\OS\LAB\LAB3>s
```

7.

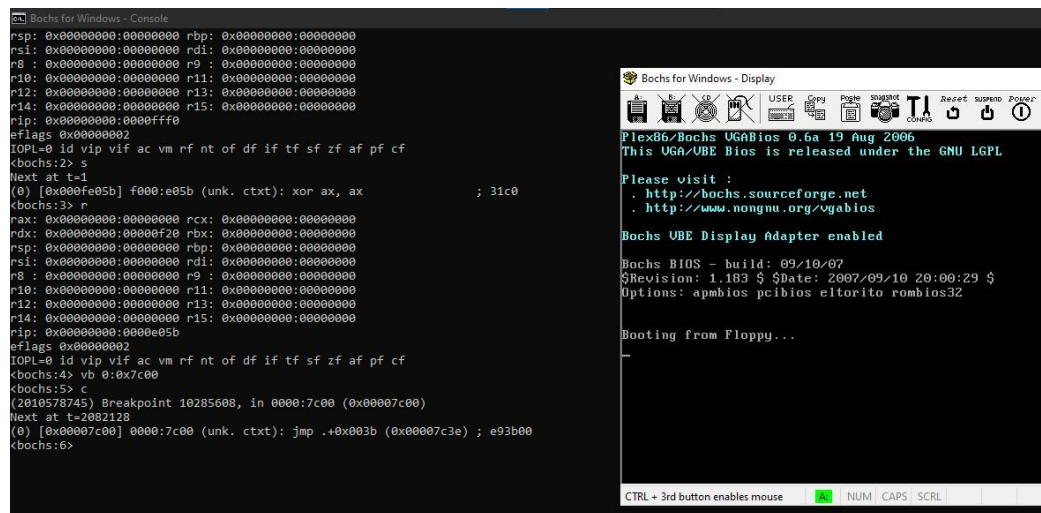
8. Ketikan r lalu enter.



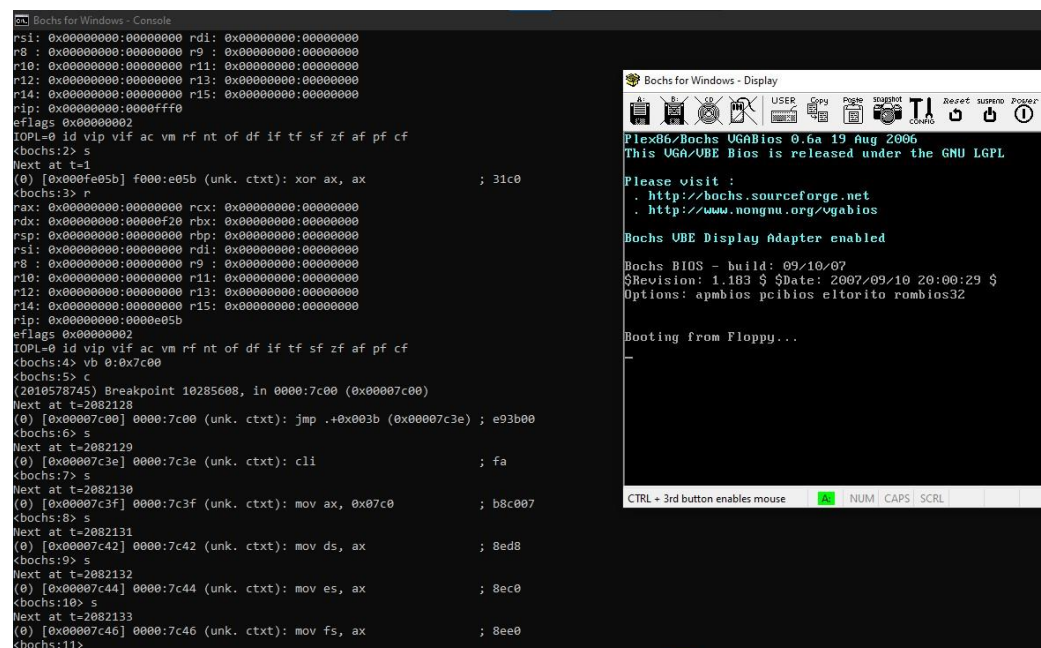
9. Selanjutnya, eksekusi tekean s lalu enter dan dilanjutkan r kemudian enter



10. Masukkan perintah `vb 0:0x7c00` lalu enter. Perintah tersebut membuat titik pemberhentian pada titik `vb 000:7c000` selanjutnya, ketik `c` lalu enter 11. Pada simulator bochs akan menampilkan gambar seperti dibawah ini:



12. Ketikkan `s` lalu enter untuk membandingkan 10 instruksi yang akan dieksekusi oleh PC dengan program yang ada di `boot.asm` lakukan perintah tersebut pada setiap Langkah bochs yang muncul



13. Mulai Kembali dari awal bila terjadi kesalahan pada titik ini. Ketikkan `cd\` lalu enter, `cd os` lalu enter, `setpath` lalu enter, `cd lab/lab3` lalu enter, ketik `s` lalu enter, selanjutnya masukkan perintah `vb 0x0100:0x000` untuk menghentikan PC saat mulai eksekusi instruksi dari program `kernel.bin` Setelah itu, ketik `c` lalu enter untuk melanjutkan seperti gambar dibawah ini:

Bochs for Windows - Console

```

(c) Microsoft Corporation. All rights reserved.

C:\Users\user>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>s

C:\OS\LAB\LAB3>.\..\bochs-2.3.5\bochsrc -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:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> vb 0x0100:0x000
<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>

```

Bochs for Windows - Display

```

Plex86/Bochs UGA BIOS 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
.....
..
..

```

CTRL + 3rd button enables mouse NUM | CAPS | SCRL

14.Selanjutnya teruskan Langkah PC simulator minimal 10 kali dengan menggunakan perintah s lalu enter. Perhatikan dan catat setiap perubahan yang ada dalam tampilan lalu bandingkan dengan code pada program kernel.asm

Bochs for Windows - Console

```

=====
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:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> vb 0x0100:0x000
<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
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>

```

Bochs for Windows - Display

```

Plex86/Bochs UGA BIOS 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
.....
..
..

```

CTRL + 3rd button enables mouse NUM | CAPS | SCRL

15.Perbandingan dengan source-code pada program kernel.asm

Bochs for Windows - Console

```
=====
000000000001[      ] reading configuration from bochsrc.bxrc
000000000001[      ] installing win32 module as the Bochs GUI
000000000001[      ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b      ; ea5be00ef0
<bochs:1> v0 0x0100:0x000
<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
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>
```

kernel.asm - Notepad

File Edit Format View Help

=====

[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 #####

Ln 1, Col 1100%Windows (CRLF)

Tugas Modul 3

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 hardisk 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.

Real-Mode:

- Didasarkan pada prosesor 8086 & 8088.
- PC IBM asli menyertakan prosesor 8088 yang dapat menjalankan instruksi 16 bit menggunakan register internal 16 bit dan dapat menangani 1 MB menggunakan 20 baris alamat.
- Mode instruksi 16 bit 8088 disebut mode real.
- Semua perangkat lunak yang berjalan dalam realmode harus menggunakan hanya instruksi 16 bit dan bekerja dalam arsitektur memori 20 bit yang didukungnya.
- Tidak ada multi tasking - tidak ada proteksi untuk mencegah satu program menimpa program lain
- Semua prosesor memiliki realmode dan sebenarnya komputer biasanya menyala dalam realmode
- Real Mode digunakan oleh aplikasi DOS dan DOS Standar.

Protect-Mode :

- Dimulai dengan chip 80286 di IBM AT, protect mode baru diperkenalkan. Ini adalah mode operasi yang jauh lebih kuat daripada real mode, dan digunakan di semua sistem operasi multitasking modern.
- Akses penuh ke semua memori sistem.
- Tidak ada batas 1 MB dalam protect mode
- Kemampuan untuk melakukan banyak tugas, artinya membuat sistem operasi mengelola eksekusi beberapa program secara bersamaan.
- Dukungan untuk memori virtual, yang memungkinkan sistem menggunakan hard disk untuk meniru memori sistem tambahan bila diperlukan.
- Akses lebih cepat (32-bit) ke memori dan driver 32-bit yang lebih cepat untuk melakukan transfer I / O
- Setiap program yang sedang berjalan memiliki lokasi memori yang ditetapkan, yang dilindungi dari konflik dengan program lain.
- Jika program mencoba menggunakan alamat memori yang tidak diizinkan, "kesalahan perlindungan" dihasilkan.
- Semua sistem operasi utama saat ini menggunakan protect mode termasuk Windows 3.x, Window 9x, Windows NT, OS / 2 dan Linux.
- Semua prosesor dari 286 on dapat menggunakan protect mode.