

Nama : Fariz Taufiqul Hafidz

NIM : L200210192

Kelas : D

MODUL 3

Mata Kuliah : Praktikum Sistem Operasi

1. Command Prompt

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

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. File sudah siap untuk proses “debugging”.

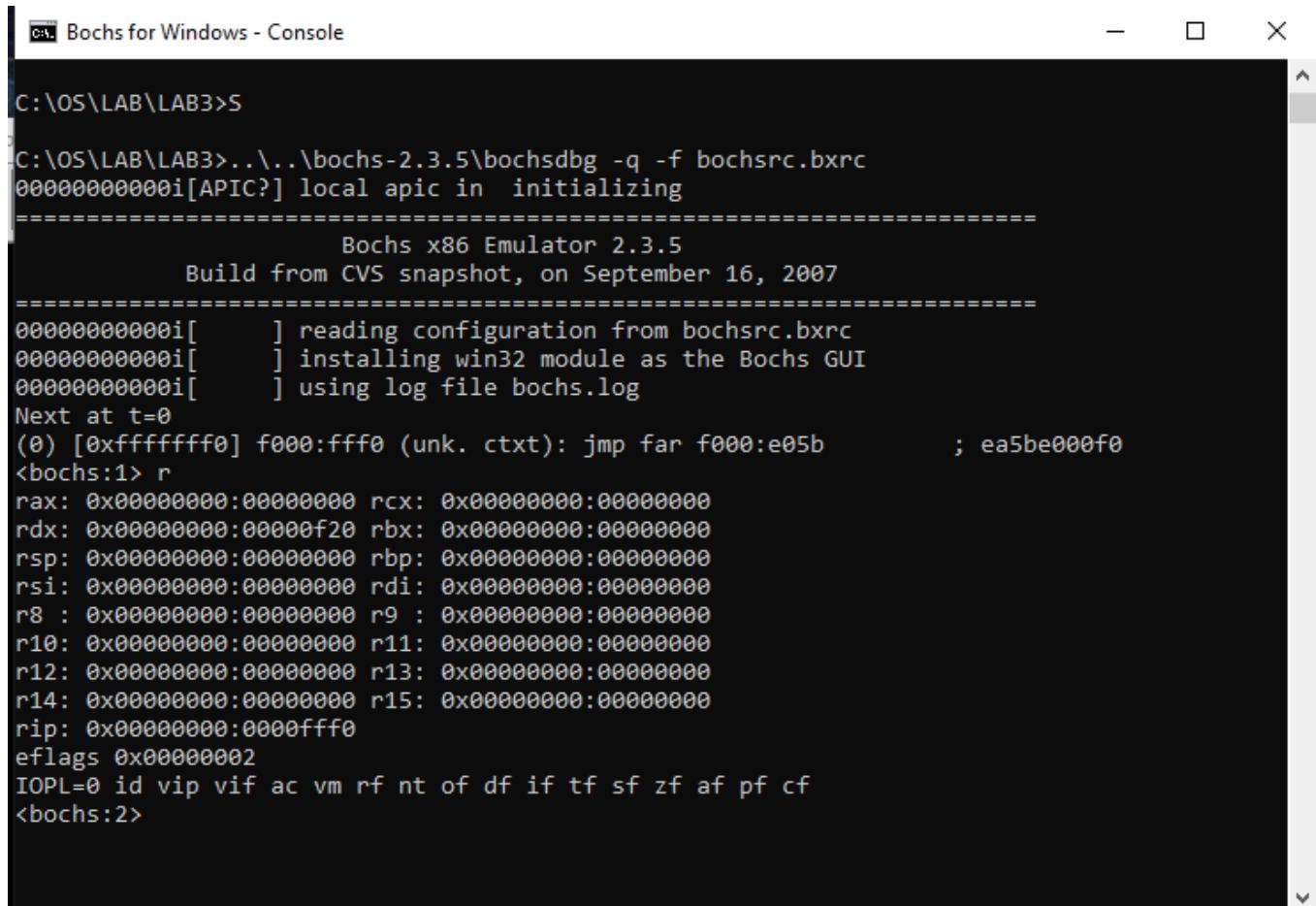
3. Mulai melakukan “debugging” :

Bochs for Windows - Console

```
c:\OS\LAB\LAB3>S

c:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsrc -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) [0xfffffffff0] f000:ffff0 (unk. ctxt): jmp far f000:e05b      ; ea5be000f0
<bochs:1>
```

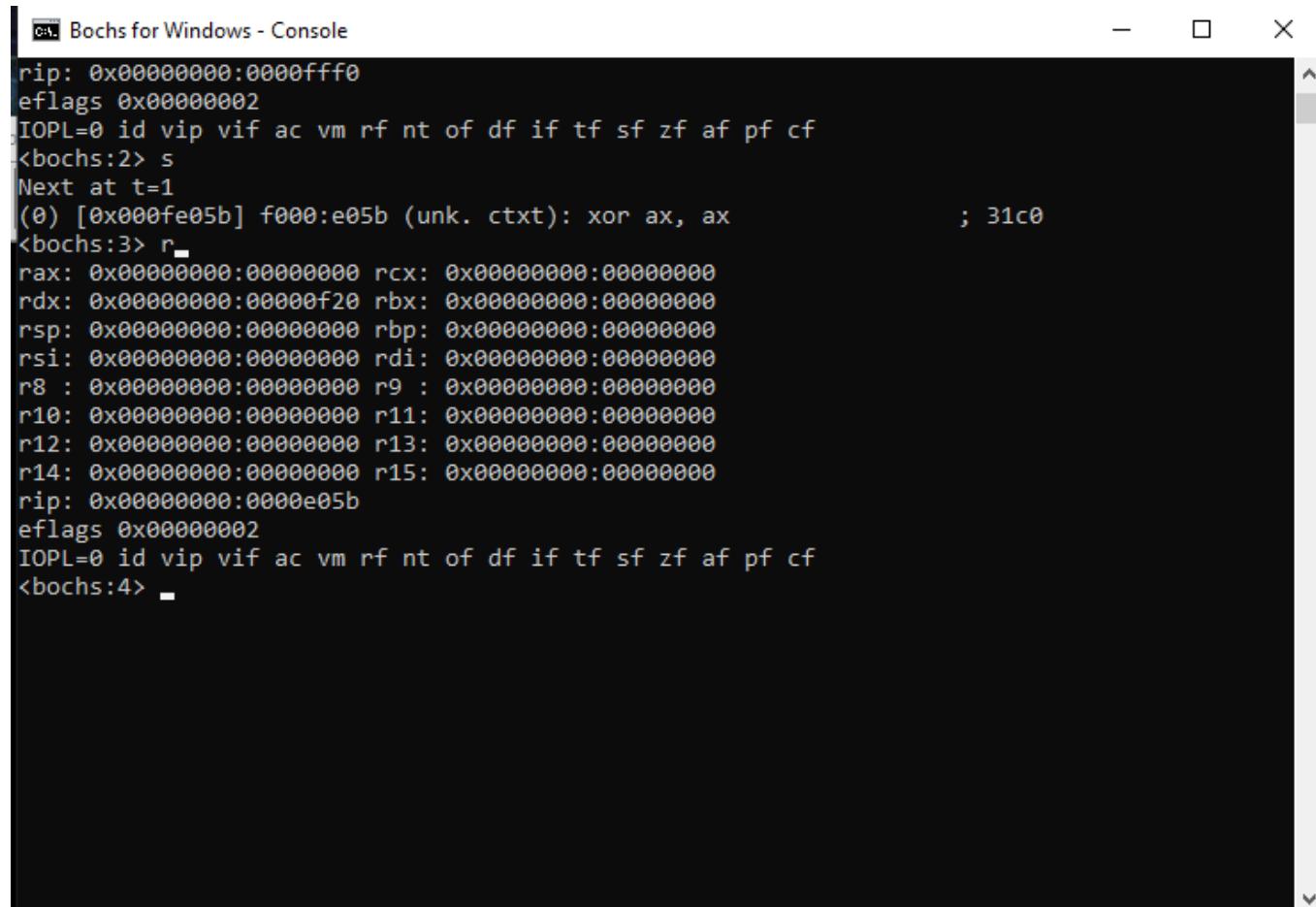
4. Melihat isi register CS dan IP :



```
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) [0xffffffff0] f000:ffff0 (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: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:2>
```

5. Ketik “s” dan “r” :



```
rip: 0x00000000:0000ffff
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. Melakukan break point pada 0000:7C00 dengan masukan perintah berikut “vb 0:0x7C00” dan melanjutkan program dengan masukkan perintah “c” :

Bochs for Windows - Console

```
rip: 0x00000000:0000ffff
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: 0x00000000:00000f20 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=2084619
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> -
```

Bochs for Windows - Display

Plex86/Bochs VGA/Bios 0.6a 19 Aug 2006
This VGA/VBE Bios is released under the GNU LGPL

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

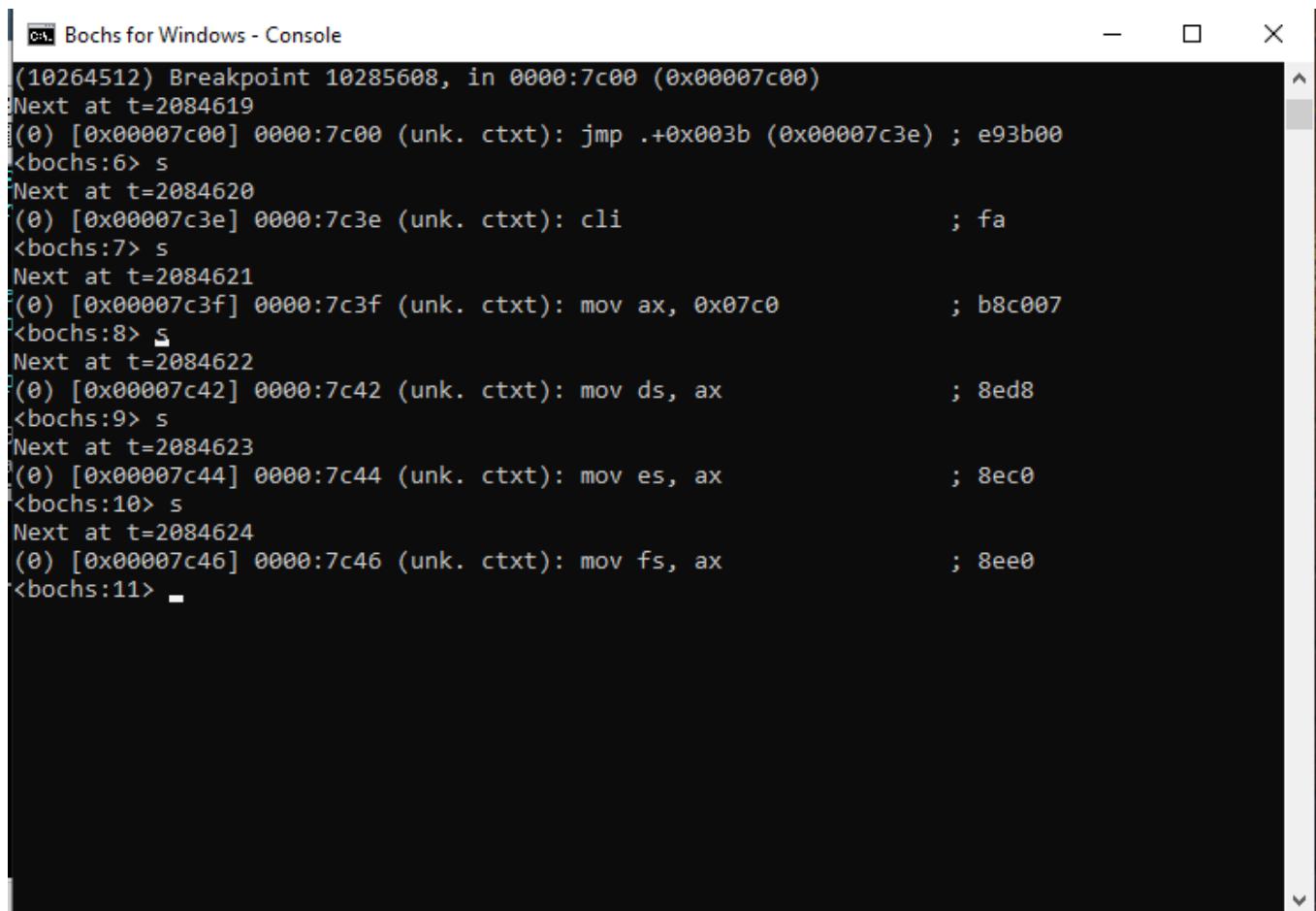
Bochs VBE Display Adapter enabled

Bochs BIOS - build: 09/10/07
\$Revision: 1.183 \$ \$Date: 2007/09/10 20:00:29 \$
Options: apmbios pcibios eltorito rombios32

Booting from Floppy...

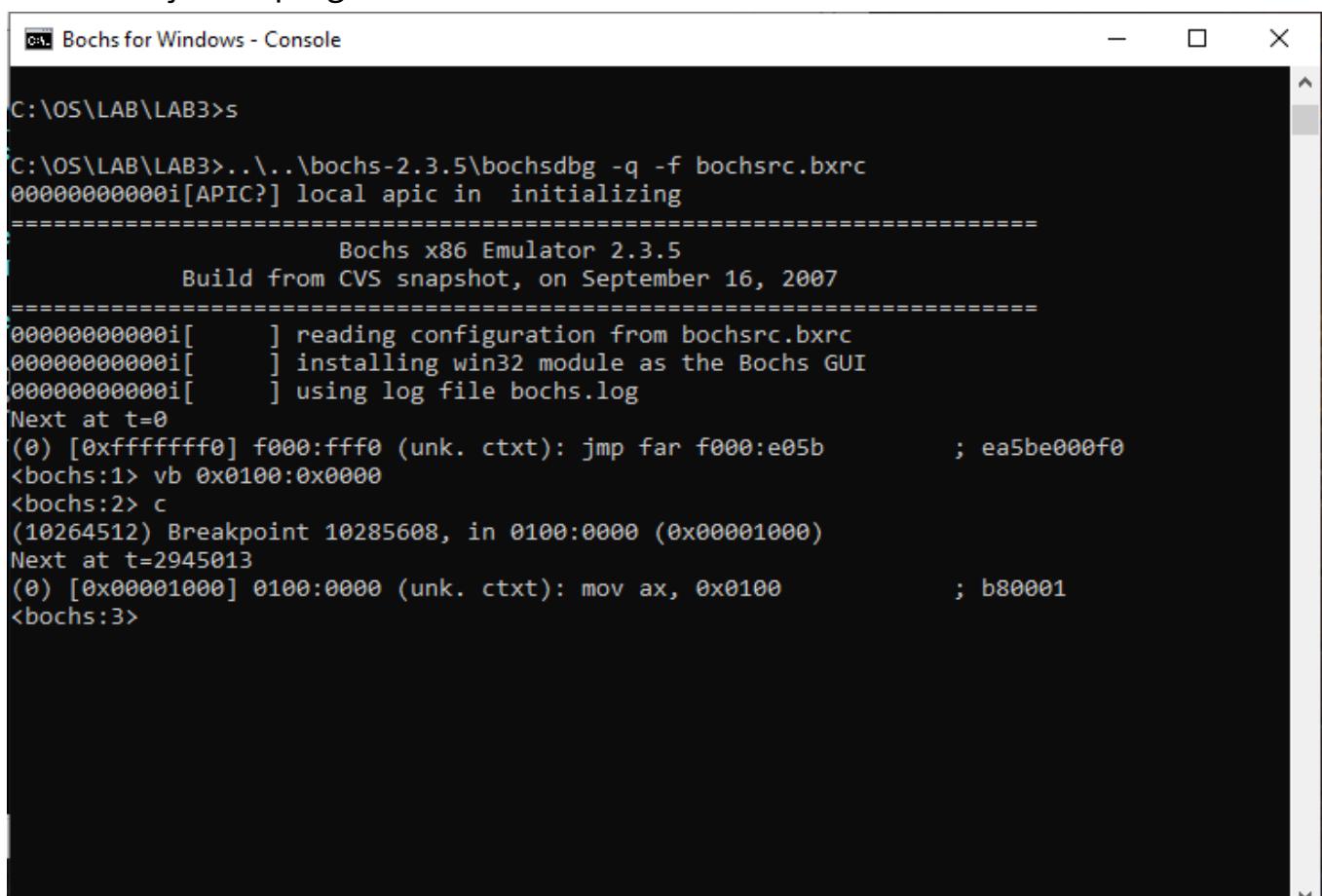
CTRL + 3rd button enables mouse A: NUM CAPS SCRL

7. Boot “floppya.img” sebagai disk-boot dan menjalankan program “boot.asm” :



The screenshot shows a terminal window titled "Bochs for Windows - Console". The window displays assembly code being executed. The code includes instructions like "jmp .+0x003b", "cli ; fa", "mov ax, 0x07c0 ; b8c007", "mov ds, ax ; 8ed8", "mov es, ax ; 8ec0", and "mov fs, ax ; 8ee0". The assembly code is annotated with comments such as "(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00" and "<bochs:6> s". The terminal window has standard window controls (minimize, maximize, close) at the top right.

8. Menjalankan program “kernel.bin” pada alamat “0100:0000”.
9. Ketik “s” kemudian membuat break-point, masukkan perintah “vb 0x0100:0x0000” dan melanjutkan program ketik “c” :



The screenshot shows a terminal window titled "Bochs for Windows - Console". The window displays the boot process of a kernel. It starts with "C:\OS\LAB\LAB3>s" and then shows the Bochs initialization log: "Bochs x86 Emulator 2.3.5 Build from CVS snapshot, on September 16, 2007". It then shows the kernel loading process: "[APIC?] local apic in initializing", "[reading configuration from bochssrc.bxrc]", "[installing win32 module as the Bochs GUI]", and "[using log file bochs.log]". The assembly code execution continues with "jmp far f000:e05b ; ea5be000f0", "vb 0x0100:0x0000", and "c". A breakpoint is set at address 0x0100:0000 with the message "(10264512) Breakpoint 10285608, in 0100:0000 (0x00001000)". The assembly code is annotated with comments like "(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100 ; b80001". The terminal window has standard window controls at the top right.

10.Ketik 's' sebanyak 10 kali :

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