

LAPORAN PRAKTIKUM SISTEM OPERASI

```
Bochs for Windows - Console
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Hp>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

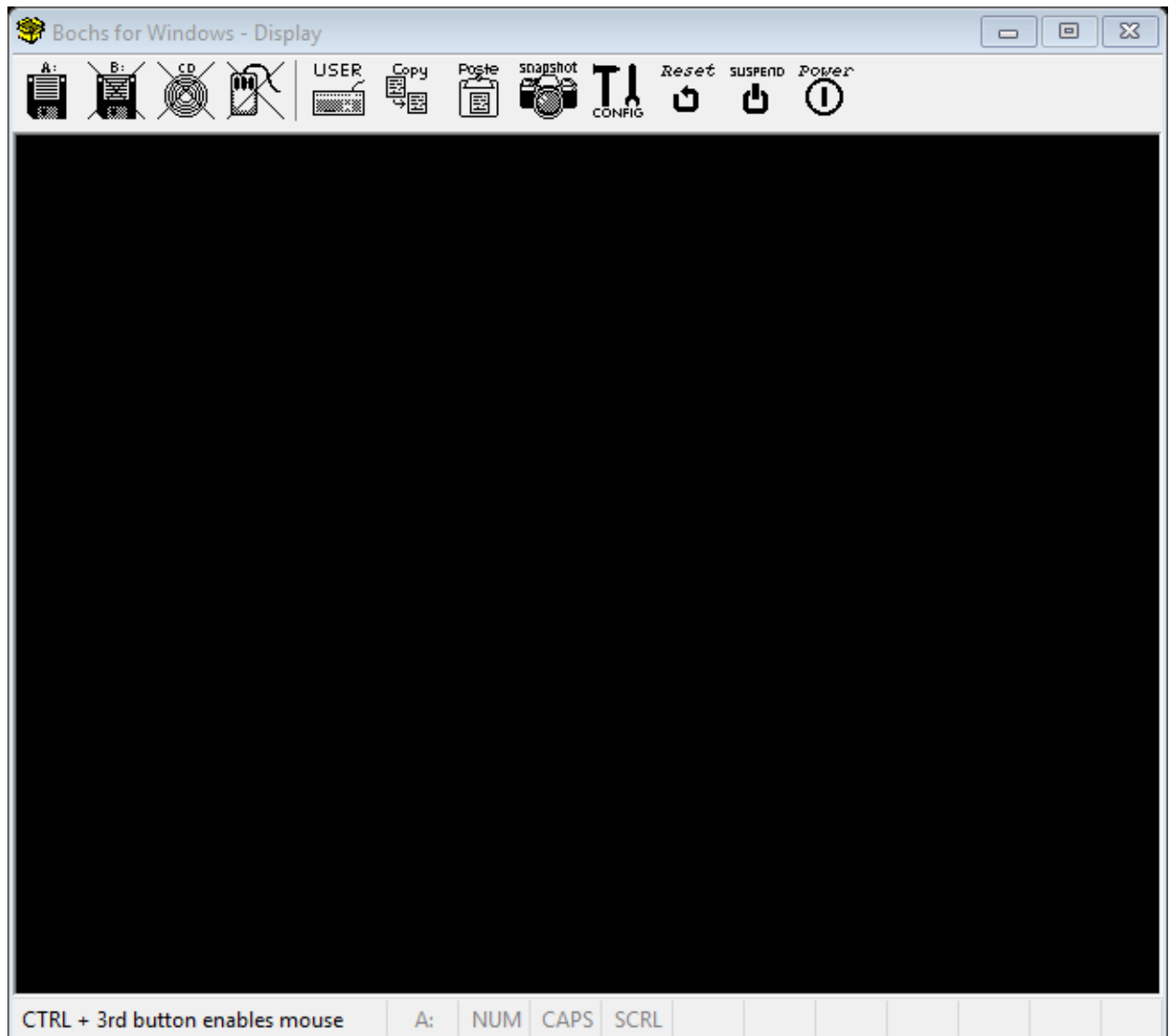
c:\OS\LAB\LAB3>type s.bat
..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc

c:\OS\LAB\LAB3>s

c:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
000000000000[APIC?] local apic in  initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000000[      ] reading configuration from bochsrc.bxrc
000000000000[      ] installing win32 module as the Bochs GUI
000000000000[      ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b      ; ea5be000f0
```

Seperti dengan praktikum lainnya yaitu diawali dengan setpath lalu berpindah ke direktori folder LAB3.

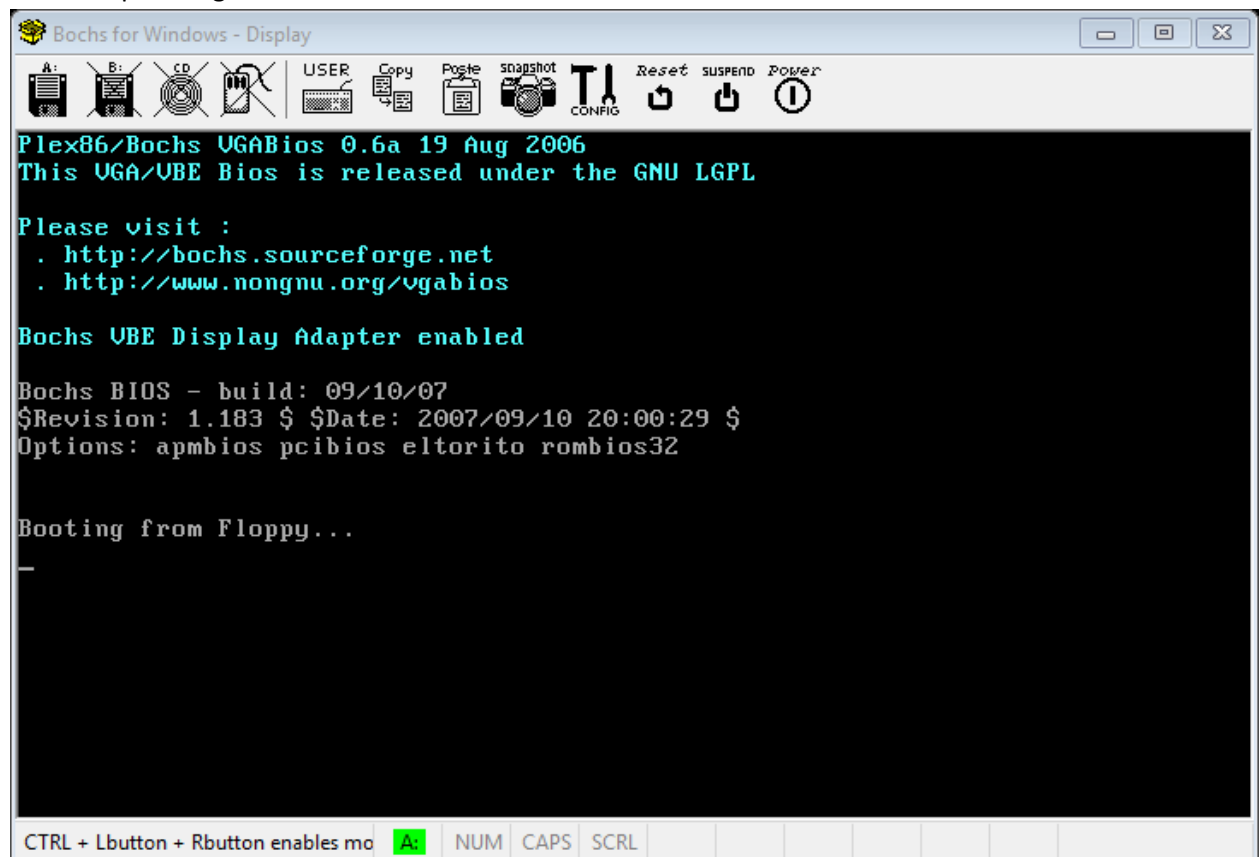
Lalu melakukan proses debugging. Program bochs yang diaktifkan adalah program versi debug yaitu Bochsdbg dengan mengetikkan *type s.bat*. perintah s digunakan untuk membuka PC-Simulator dan akan terlihat seperti dibawah:



Perintah *r* digunakan untuk melihat isi CS dan IP. sekarang kita suruh PC untuk mengeksekusi dengan perintah *s* . lalu lakukan perintah *r* lagi. Kita biarkan PC melanjutkan pekerjaannya, kemudian kita suruh berhenti saat PC mulai memasuki tahap BOOT yang dimulai pada alamat 0000:7c00 .

```
Bochs for Windows - Console
<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: 0x00000000:0000ffff
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) [0x000fe5b] f00: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
(2006900537) Breakpoint 10285624, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
```

Maksud perintah *c* adalah teruskan proses sampai ke titik pemberhentian. Lalu masuk ke PC maka akan tampil sebagai berikut.

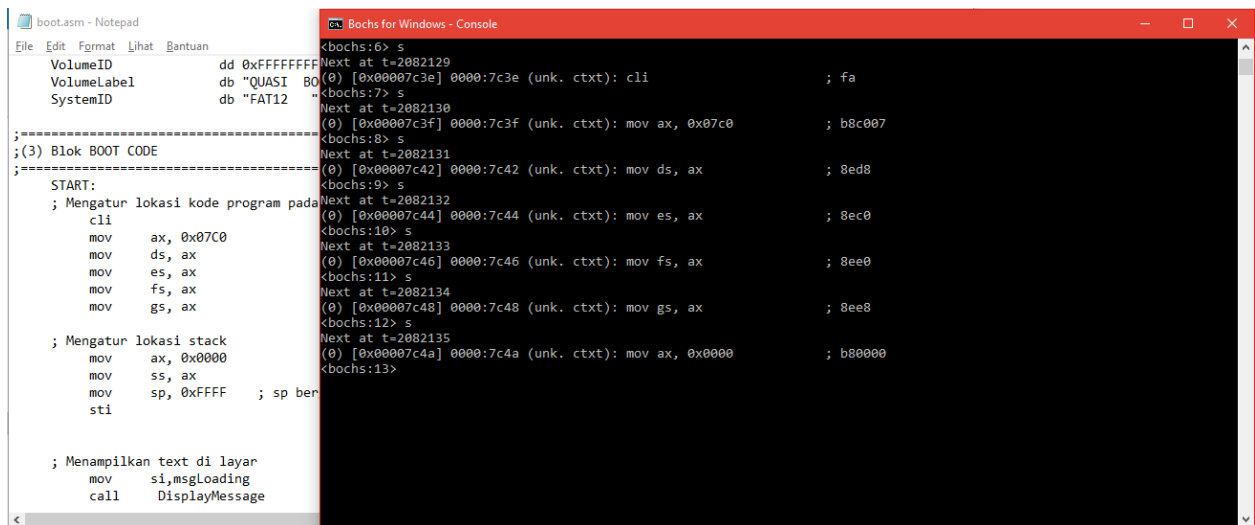


```

<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> s
Next at t=2082134
(0) [0x00007c48] 0000:7c48 (unk. ctxt): mov gs, ax          ; 8ee8
<bochs:12> s
Next at t=2082135
(0) [0x00007c4a] 0000:7c4a (unk. ctxt): mov ax, 0x0000      ; b80000

```

Lakukan perintah s langkah demi langkah sebanyak maksimal 7 kali lalu bandingkan dengan boot.asm



Lalu hapus semua & lakukan semua langkah diatas dengan sama lalu bandingkan dengan kernel.asm

