

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



DISUSUN OLEH :

Imrokatun Umi Fadhillah P

L200210267

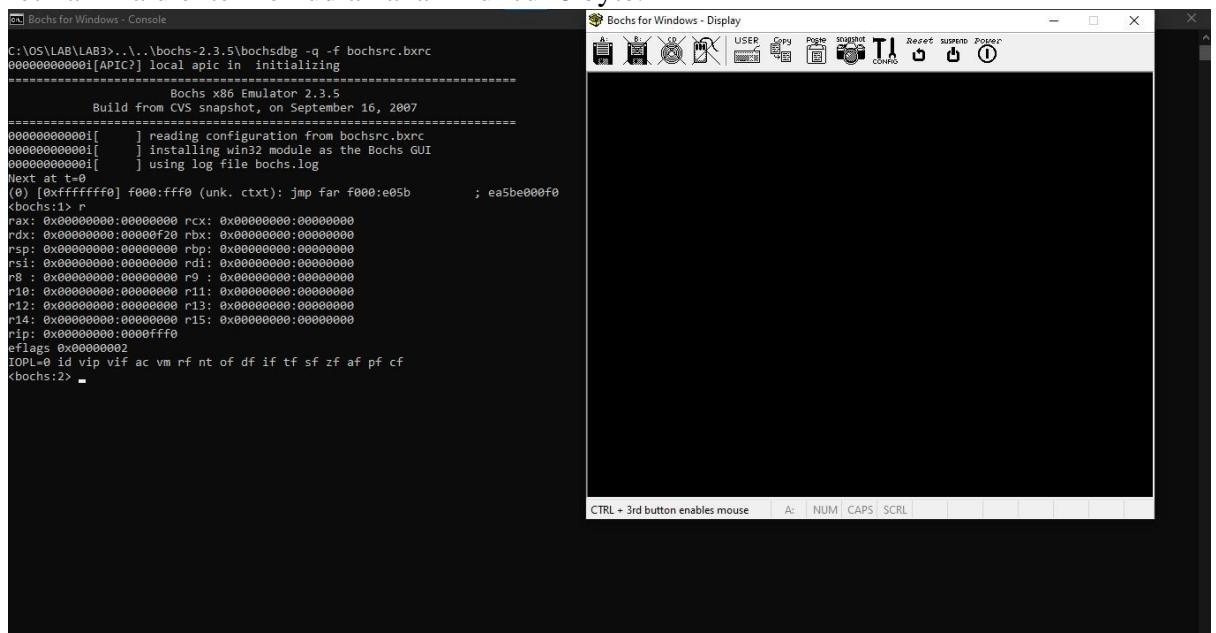
**TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2022/2023**

LAPORAN LATIHAN PRAKTIK LANGKAH KERJA :

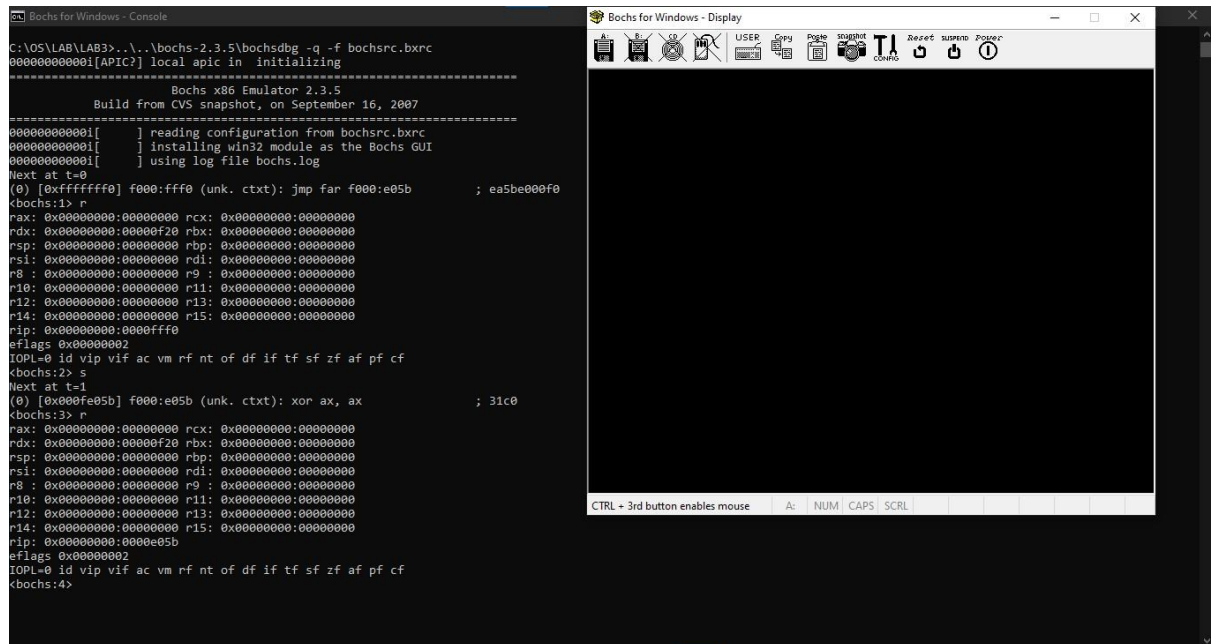
1. Start kemudian buka cmd
2. Ketikan `cd ../..` lalu enter
3. Ketikan `cd os` lalu enter
4. Ketikan `setpath` lalu enter
5. Ketikan `cd lab/lab3` lalu enter
6. Ketikan `setpath` lalu enter
7. Ketikan `s` lalu enter untuk melakukan debugging. Kemudian akan muncul bochs yang terlihat kosong seperti dibawah ini:

```
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
```

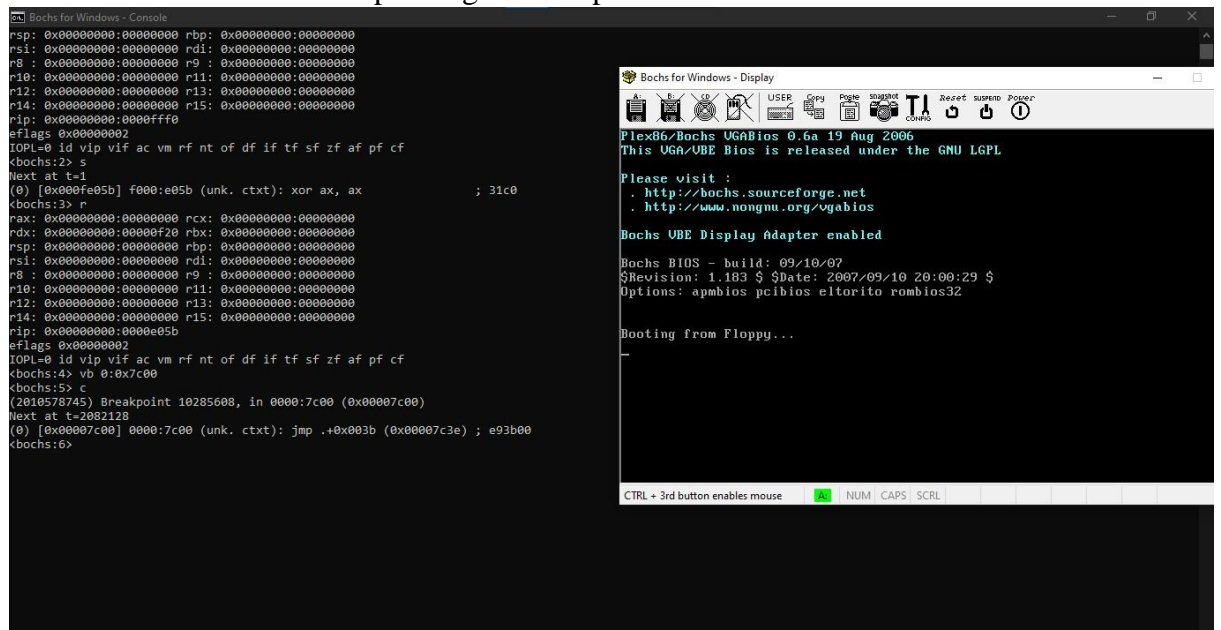
8. Ketikan `r` lalu enter kemudian akan muncul 8 byte.



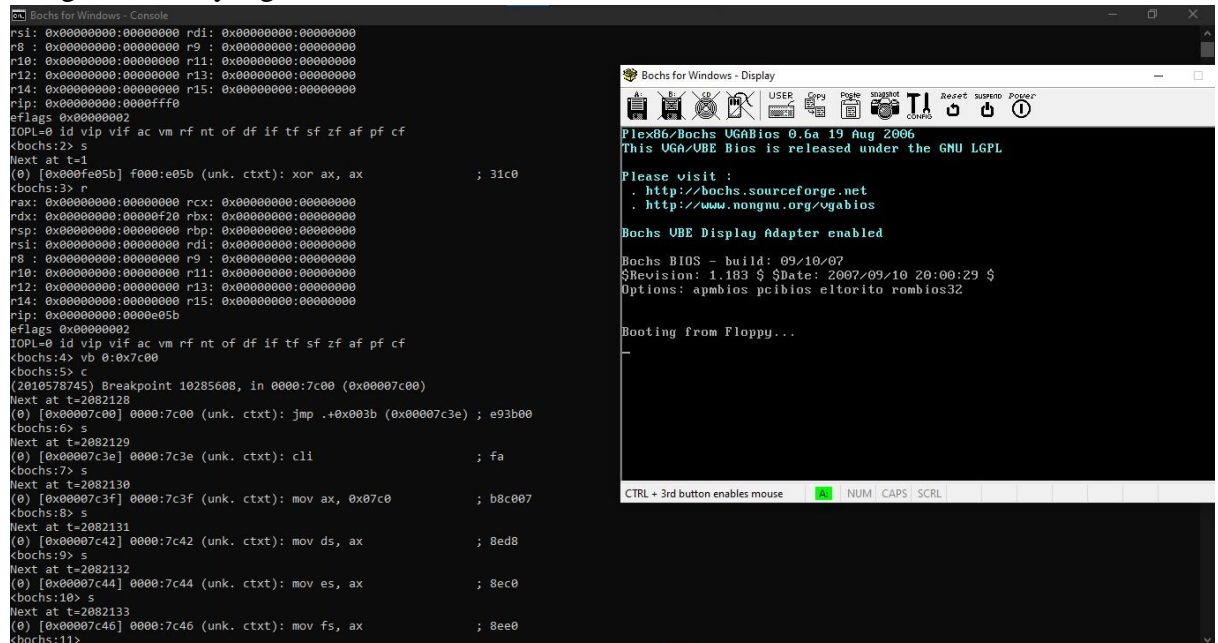
9. Selanjutnya, kita eksekusi dengan mengetikkan s lalu enter dan dilanjutkan r kemudian enter



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:



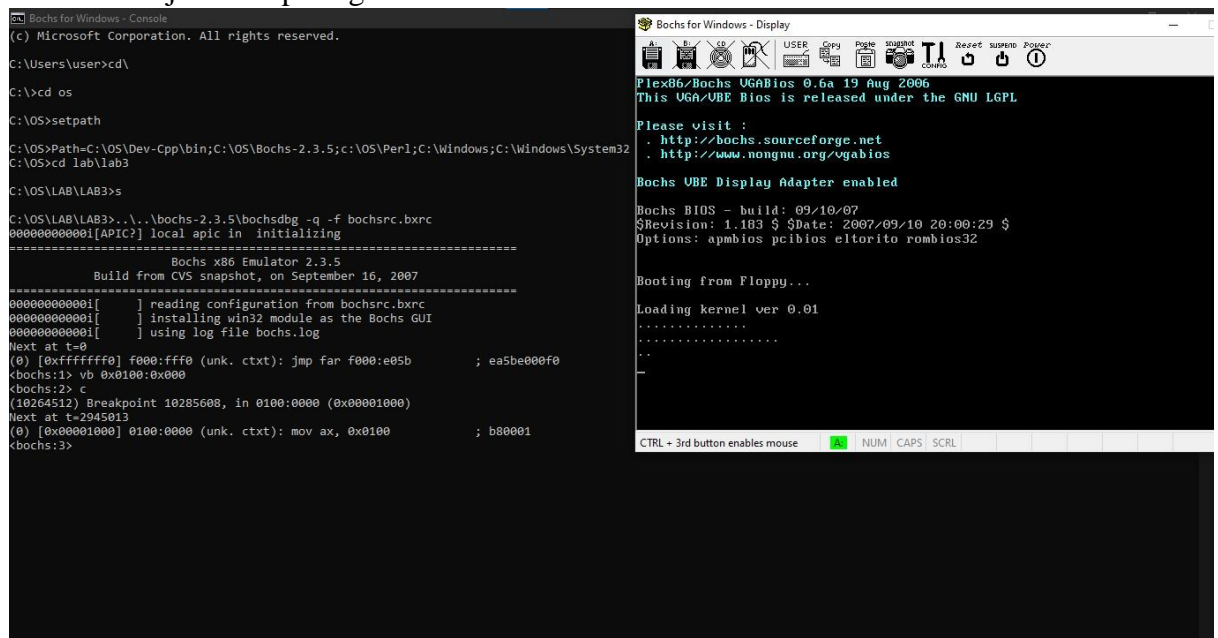
12. Ketikkan `s` lalu enter untuk membandingkan 10 instruksi yang akan dieksekusi oleh PC dengan program yang ada di `boot.asm` lakukan perintah tersebut pada setiap Langkah bochs yang muncul



The screenshot shows two windows from the Bochs emulator. The 'Bochs for Windows - Console' window displays assembly instructions and register values. The 'Bochs for Windows - Display' window shows the BIOS boot process, including the 'Flex86/Bochs UGA/Bios 0.6a 19 Aug 2006' message and the 'Bochs UBE Display Adapter enabled' message. The console window shows the following instructions and register values:

```
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> 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
(2010578745) 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>
```

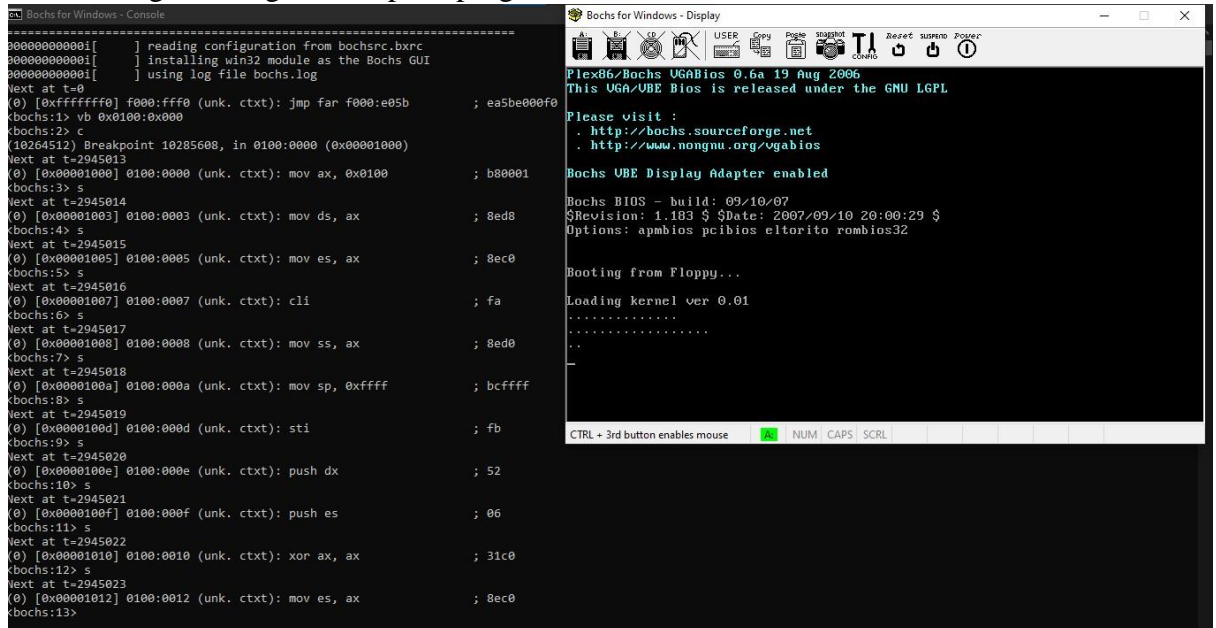
13. Kemudian jika terjadi kesalahan hetikan PC simulator dan mulai Kembali dari awal. Ketikkan `cd\` lalu enter, `cd os` lalu enter, `setpath` lalu enter, `cd lab/lab3` lalu enter, ketik `s` lalu enter, selanjutnya masukkan perintah `vb 0x0100:0x000` untuk menghentikan PC saat mulai eksekusi instruksi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



The screenshot shows two windows from the Bochs emulator. The 'Bochs for Windows - Console' window displays the boot process, including the 'Build from CVS snapshot, on September 16, 2007' message and the 'Loading kernel ver 0.01' message. The 'Bochs for Windows - Display' window shows the BIOS boot process, including the 'Flex86/Bochs UGA/Bios 0.6a 19 Aug 2006' message and the 'Bochs UBE Display Adapter enabled' message. The console window shows the following instructions and register values:

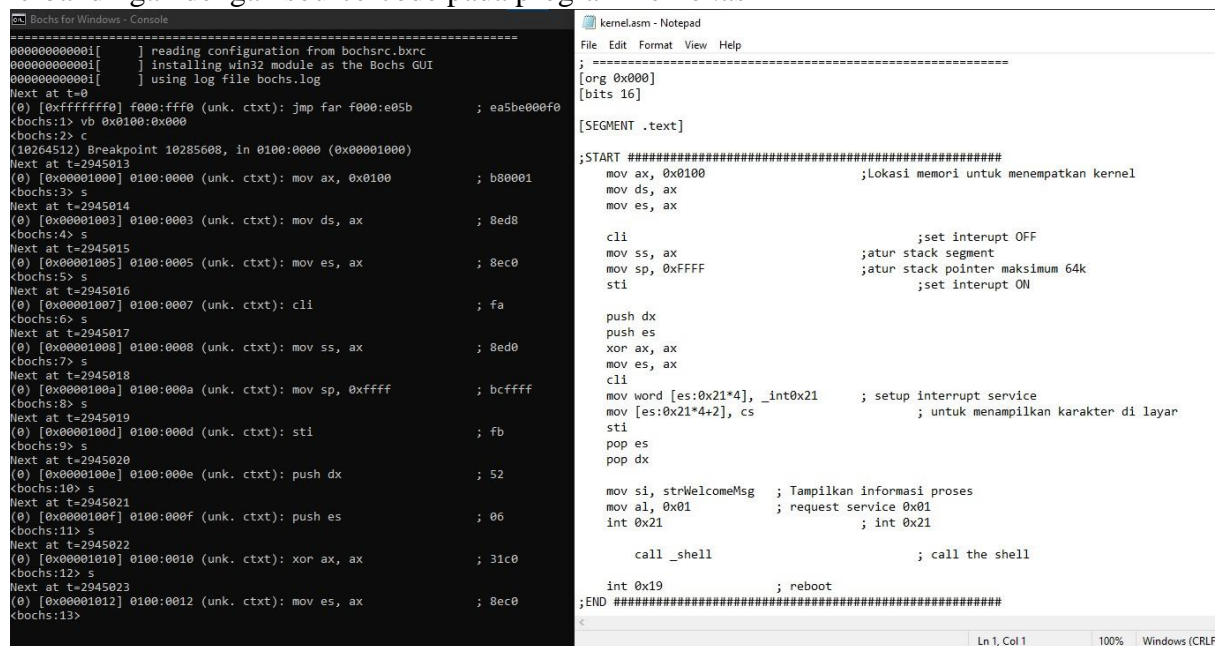
```
(c) Microsoft Corporation. All rights reserved.
C:\Users\User>cd\
C:\>cd os
C:\OS>setpath
C:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;C:\Windows;C:\Windows\System32
C:\OS>cd lab\lab3
C:\OS\LAB\LAB3>s
C:\OS\LAB\LAB3>...\bochs-2.3.5\bochsrc -q -f bochsrc.bxrc
0000000000i[APIC?] local apic in initializing
=====
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) [0xfffffff0] f000:fff0 (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>
```

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



The screenshot shows two windows from the Bochs emulator. The 'Bochs for Windows - Console' window displays the BIOS boot process, including reading configuration, installing the win32 module, and executing assembly code. The 'Bochs for Windows - Display' window shows the BIOS boot screen with text: 'Flex86/Bochs UGABios 0.6a 19 Aug 2006', 'This UGA/UE Bios is released under the GNU LGPL', 'Please visit: http://bochs.sourceforge.net, http://www.nongnu.org/ugabios', '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...', and 'Loading kernel ver 0.01'.

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



The screenshot shows two windows. The 'Bochs for Windows - Console' window displays the BIOS boot process, including reading configuration, installing the win32 module, and executing assembly code. The 'kernel.asm - Notepad' window displays the assembly code for the kernel, including comments and instructions. The code includes a segment definition, a start label, and various instructions for setting up the interrupt service, stack, and displaying a message.

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.