

Nama : Aji Prastyo

NIM : L200170082

Kelas : E

1. Masuk ke direktori C:/OS, lakukan setpath dan masuk ke direktori lab/lab3

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\User>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>
```

2. ketikkan type s.bat

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:/OS

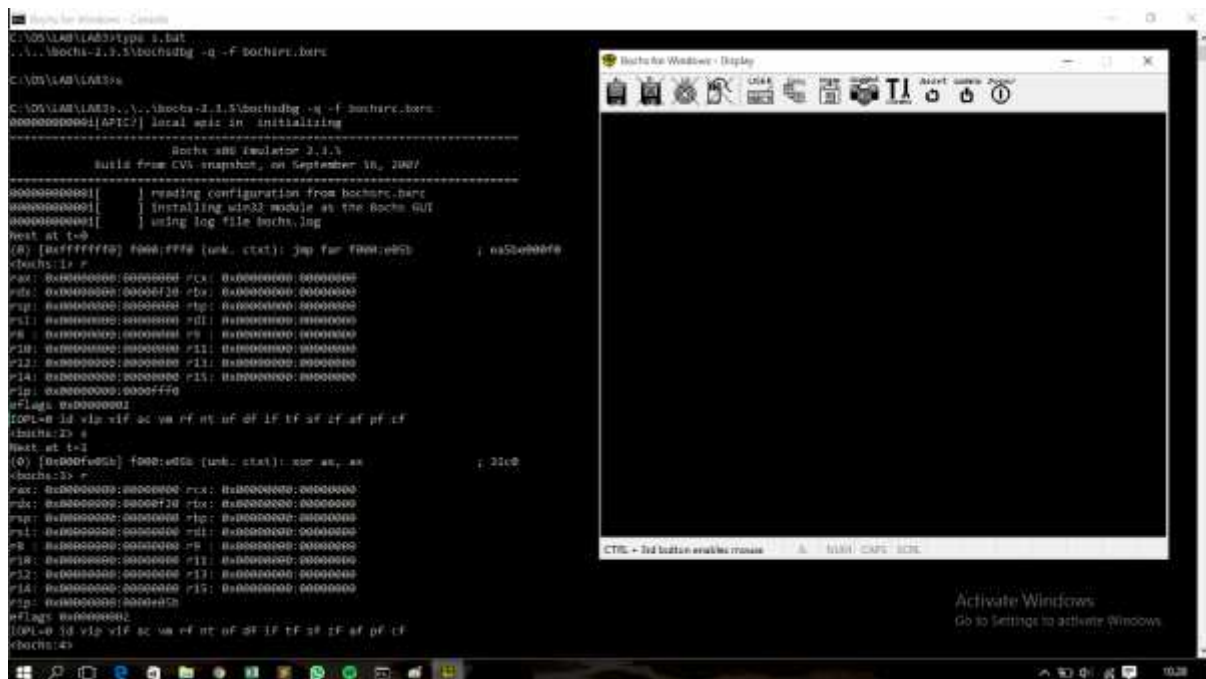
C:\OS>setpath

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

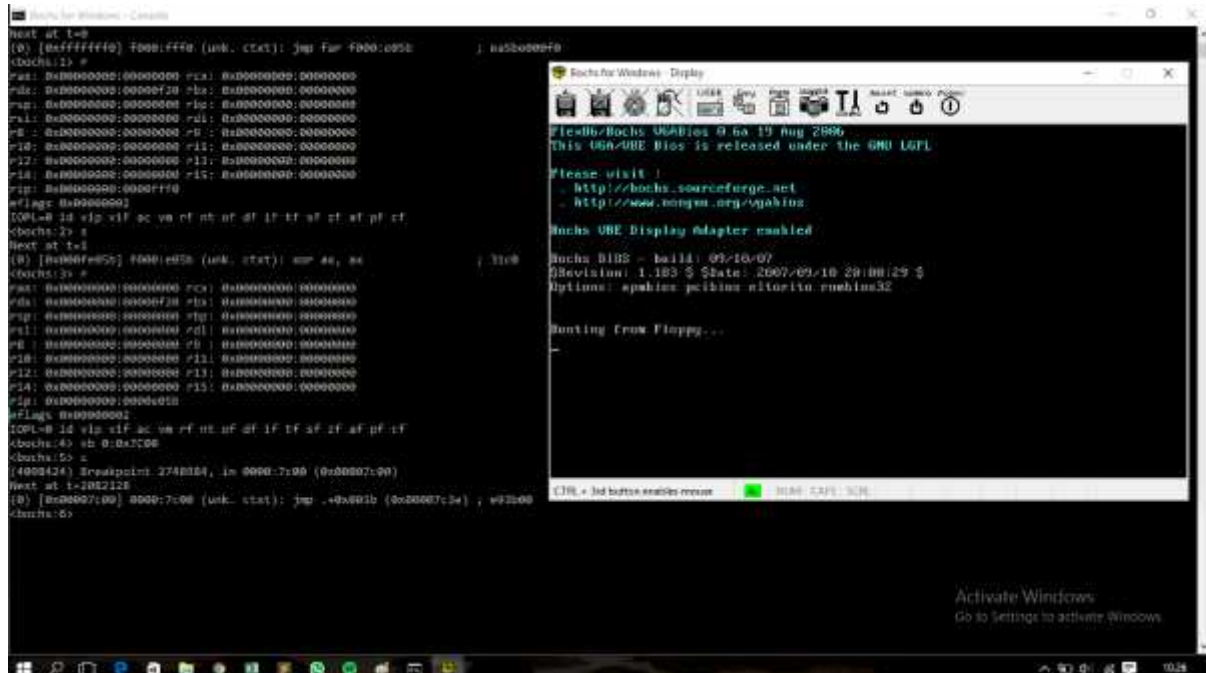
C:\OS\LAB\LAB3>type s.bat
C:\OS\LAB\LAB3>
C:\OS\LAB\LAB3>
```

[illegible]

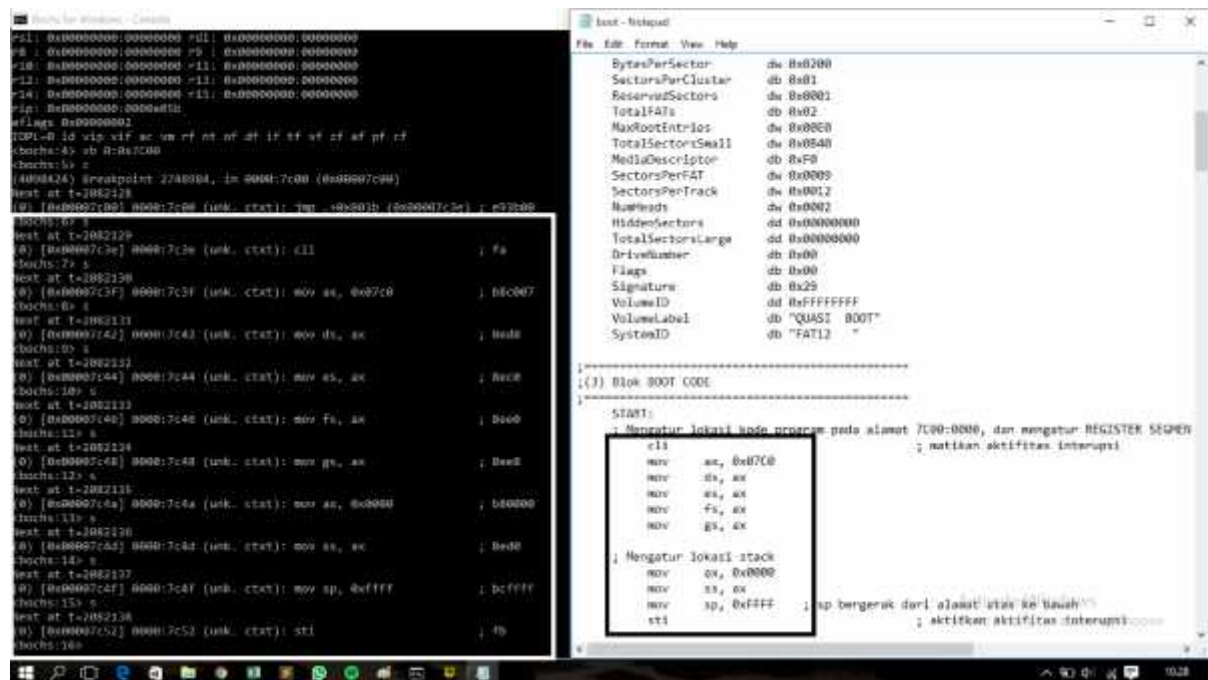
## 6. Ketikkan 's'



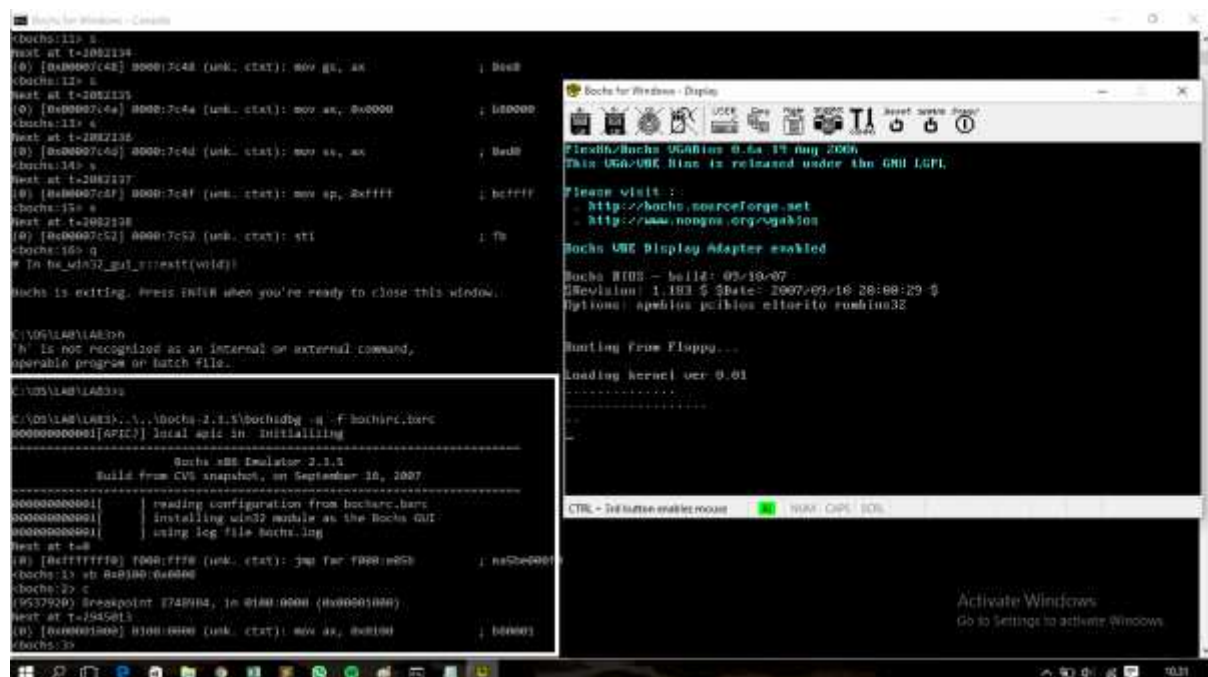
## 7. Kemudian masukkan perintah 'vb 0:0x7C00' untuk membuat pemberhentian di alamat tersebut.



8. Ketikkan 'c' untuk continue / melanjutkan. Lalu ketikkan 's' berulang sebanyak 10 kali, dan lalukan pengecekan dengan file boot.asm



9. Ketikkan 'q' untuk menghentikan debugging. Kemudian lakukan debugging lagi dengan cara ketikkan 's', kemudian ketikkan 'vb 0x0100:0x0000' untuk menghentikan langkah saat PC mulai mengeksekusi instruksi dari program 'kernel.bin', lalu ketikkan 'c'



The image shows a Windows desktop with two windows. The left window is 'Bochs x86 emulator 2.3.5' running a virtual machine. The right window is a terminal titled 'kernel - Tsp' showing assembly code for a kernel. A red box highlights a section of the assembly code in the terminal.

**Bochs x86 emulator 2.3.5**

```

Bochs from CVS snapshot, on September 18, 2007
=====
000000000000[    ] reading configuration from bochs.conf
000000000000[    ] installing win32 module as the Bochs GUI
000000000000[    ] using log file bochs.log
test at t=0
(0) [0000000000] 0000:ffff (unk. stat): jmp far f000-e000    ; ea2000f0
bochs:1> vs 0x0100-0x0000
bochs:2> s
(0x17520) breakpoint 2748064, in 0100:0000 (0x0000)mem
=====
(0) [0x00001000] 0100:0000 (unk. stat): mov ax, 0x0100      ; 000001
bochs:3> s
test at t=2045014
(0) [0x00001000] 0100:0001 (unk. stat): mov dx, ax          ; 0000
bochs:4> s
test at t=2045015
(0) [0x00001005] 0100:0005 (unk. stat): mov ax, ax          ; 0000
bochs:5> s
test at t=2045016
(0) [0x00001007] 0100:0007 (unk. stat): cll                 ; fa
bochs:6> s
test at t=2045017
(0) [0x00001008] 0100:0008 (unk. stat): mov ax, ax          ; 0000
bochs:7> s
test at t=2045018
(0) [0x0000100a] 0100:000a (unk. stat): mov ap, 0xffff      ; bcffff
bochs:8> s
test at t=2045019
(0) [0x0000100d] 0100:000d (unk. stat): shl                 ; fb
bochs:9> s
test at t=2045020
(0) [0x0000100e] 0100:000e (unk. stat): push dx             ; 02
bochs:10> s
test at t=2045021
(0) [0x0000100f] 0100:000f (unk. stat): push ax             ; 00
bochs:11> s
test at t=2045022
(0) [0x00001010] 0100:0010 (unk. stat): xor ax, ax           ; 3ic0
bochs:12> s
test at t=2045023
(0) [0x00001012] 0100:0012 (unk. stat): mov ax, ax           ; 0000

```

**kernel - Tsp**

```

File Edit Format View Help
;
; Prototype: SIMPLE_KERNEL ver=0.01
; LAB-INFORMATIKA
;=====
[org 0x000]
[bits 16]

[SEGMENT .text]

;=====
;START
mov ax, 0x0100                ;lokasi memori untuk menempatkan kernel
mov dx, ax
mov ax, ax

cli                            ;set Interrupt OFF
mov bx, ax                    ;atur stack segment
mov sp, 0xffff                ;atur stack pointer maksimum 64k
sti                            ;set Interrupt ON

push dx
push es
xor ax, ax
mov ax, ax

cli
mov word [es:0x21*0], _intdx21 ; setup interrupt service
mov [es:0x21*4+2], cx          ; untuk menampilkan karakter di layar
sti
pop es
pop dx

mov zi, strWelcomeMsg          ; Tampilkan informasi proses
mov al, 0x01                    ; request service 0x01
int 0x21                        ; int 0x21

call _shell                     ; call the shell

```

### 1. Tabel pemetaan memori pada PC

No.	Blok Memori	Alokasi Pemakaian
1	F 0 0 0 0	ROM BIOS, Diagnostic, BASIC
2	E 0 0 0 0	ROM program
3	D 0 0 0 0	ROM program
4	C 0 0 0 0	Perluasan BIOS untuk hardisk XT
5	B 0 0 0 0	Monokrom Monitor
6	A 0 0 0 0	Monitor EGA, VGS, dll
7	9 0 0 0 0	Daerah kerja pemakai s/d 640 KB
8	8 0 0 0 0	Daerah kerja pemakai s/d 576 KB
9	7 0 0 0 0	Daerah kerja pemakai s/d 512 KB
10	6 0 0 0 0	Daerah kerja pemakai s/d 448 KB
11	5 0 0 0 0	Daerah kerja pemakai s/d 384 KB
12	4 0 0 0 0	Daerah kerja pemakai s/d 320 KB
13	3 0 0 0 0	Daerah kerja pemakai s/d 256 KB
14	2 0 0 0 0	Daerah kerja pemakai s/d 192 KB
15	1 0 0 0 0	Daerah kerja pemakai s/d 128 KB
16	0 0 0 0 0	Daerah kerja pemakai s/d 64 KB

## 2. Perbedaan mode kerja “Real Mode” dan “Protected Mode”

- 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.
- 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.