

PRAKTIKUM SISTEM OPERASI
MENGENAL CARA ‘DEBUGGING’ PROGRAM
BOOTSTRAP-LOADER



Penyusun:

Nama : Della Fitria Lestari
NIM : L200219268
Kelas : E
Mata Kuliah : Praktikum Sistem Operasi
Program Studi : Informatika

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA

2021

Tugas

1. Tabel pemetaan memori pada PC:
- Real mode address space (< 1 MiB)

start	end	size	description	type	
Real mode address space (the first MiB)					
0x00000000	0x000003FF	1 KiB	Real Mode IVT (Interrupt Vector Table)	unusable in real mode	640 KiB RAM ("Low memory")
0x00000400	0x000004FF	256 bytes	BDA (BIOS data area)		
0x00000500	0x00007BFF	almost 30 KiB	Conventional memory		
0x00007C00	0x00007DFF	512 bytes	Your OS BootSector	usable memory	
0x00007E00	0x0007FFFF	480.5 KiB	Conventional memory		
0x00080000	0x0009FFFF	128 KiB	EBDA (Extended BIOS Data Area)		
0x000A0000	0x000BFFFF	128 KiB	Video display memory	hardware mapped	384 KiB System / Reserved ("Upper Memory")
0x000C0000	0x000C7FFF	32 KiB (typically)	Video BIOS	ROM and hardware mapped / Shadow RAM	
0x000C8000	0x000EFFFF	160 KiB (typically)	BIOS Expansions		
0x000F0000	0x000FFFFF	64 KiB	Motherboard BIOS		

- BIOS Data Area (BDA)

address (size)	description
0x0400 (4 words)	IO ports for COM1-COM4 serial (each address is 1 word, zero if none)
0x0408 (3 words)	IO ports for LPT1-LPT3 parallel (each address is 1 word, zero if none)
0x040E (word)	EBDA base address >> 4 (usually!)
0x0410 (word)	packed bit flags for detected hardware
0x0413 (word)	Number of kilobytes before EBDA / unusable memory
0x0417 (word)	keyboard state flags
0x041E (32 bytes)	keyboard buffer
0x0449 (byte)	Display Mode
0x044A (word)	number of columns in text mode
0x0463 (2 bytes, taken as a word)	base IO port for video
0x046C (word)	# of IRQ0 timer ticks since boot
0x0475 (byte)	# of hard disk drives detected
0x0480 (word)	keyboard buffer start
0x0482 (word)	keyboard buffer end
0x0497 (byte)	last keyboard LED/Shift key state

- Extended Memory (> 1 MiB)

start	end	size	region/exception	description
High Memory				
0x00100000	0x00EFFFFFFF	0x00E00000 (14 MiB)	RAM -- free for use (if it exists)	Extended memory 1, 2
0x00F00000	0x00FFFFFFF	0x00100000 (1 MiB)	Possible memory mapped hardware	ISA Memory Hole 15-16MB 3
0x01000000	????????	???????? (whatever exists)	RAM -- free for use	More Extended memory 1
0xC0000000 (sometimes, depends on motherboard and devices)	0xFFFFFFFF	0x40000000 (1 GiB)	various (typically reserved for memory mapped devices)	Memory mapped PCI devices, PnP NVRAM?, IO APIC/s, local APIC/s, BIOS, ...
0x0000000010000000 0 (possible memory above 4 GiB)	??????????????	?????????????? (whatever exists)	RAM -- free for use (PAE/64bit)	More Extended memory 1
????????????????	??????????????	??????????????	Possible memory mapped hardware	Potentially usable for memory

				mapped PCI devices in modern hardware (but typically not, due to backward compatibility)
--	--	--	--	--

- Ringkasan *Direct Mapping*

Item	Keterangan
Panjang alamat	$(s + W)$ bits
Jumlah unit yang dialamati	$2s + W$ words or bytes
Ukuran bloks sama dengan ukuran Line	$2W$ words or bytes
Jumlah blok memori utama	$2s + w/2W = 2s$
Jumlah line di chace	$M = 2r$
Besarnya tag	$(s - r)$ bits

- Ringkasan *Associative Mapping*

Item	Keterangan
Panjang alamat	$(s + W)$ bits
Jumlah unit yang dialamati	$2s + W$ words or bytes
Ukuran bloks sama dengan ukuran Line	$2W$ words or bytes
Jumlah blok memori utama	$2s + w/2W = 2s$
Jumlah line di chace	Undetermined
Besarnya tag	S bits

- Ringkasan *Set Associative Mapping*

Item	Keterangan
Panjang alamat	$(s + W)$ bits
Jumlah unit yang dialamati	$2s + W$ words or bytes
Ukuran bloks sama dengan ukuran Line	$2W$ words or bytes
Jumlah blok memori utama	$2s + w/2W = 2s$
Jumlah line di chace	Undetermined
Jumlah line dalam set	k
Jumlah set	$V = 2d$
Jumlah line di chace	$K_v = k \cdot 2d$
Besarnya tag	$(s - d)$ bits

2. Perbedaan antara mode kerja ‘Real-Mode’ dan mode kerja ‘Protect-Mode’ pada PC IBM Compatible:

• **Real-Mode**

- Merupakan suatu modus dengan prosesor Intel x86 yang berjalan seakan dirinya adalah sebuah prosesor Intel 8085 atau Intel 8088, walaupun ia merupakan prosesor Intel 80286 atau lebih tinggi. Karena itu, modus ini disebut juga sebagai modus 8086 (8086 Mode).
- Prosesor hanya bisa 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.
- Menggunakan BIOS subroutines dan juga dengan OS subroutines.

• **Protected-Mode**

- Merupakan 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.
- Hanya menggunakan BIOS OS subroutines.

Langkah Kerja

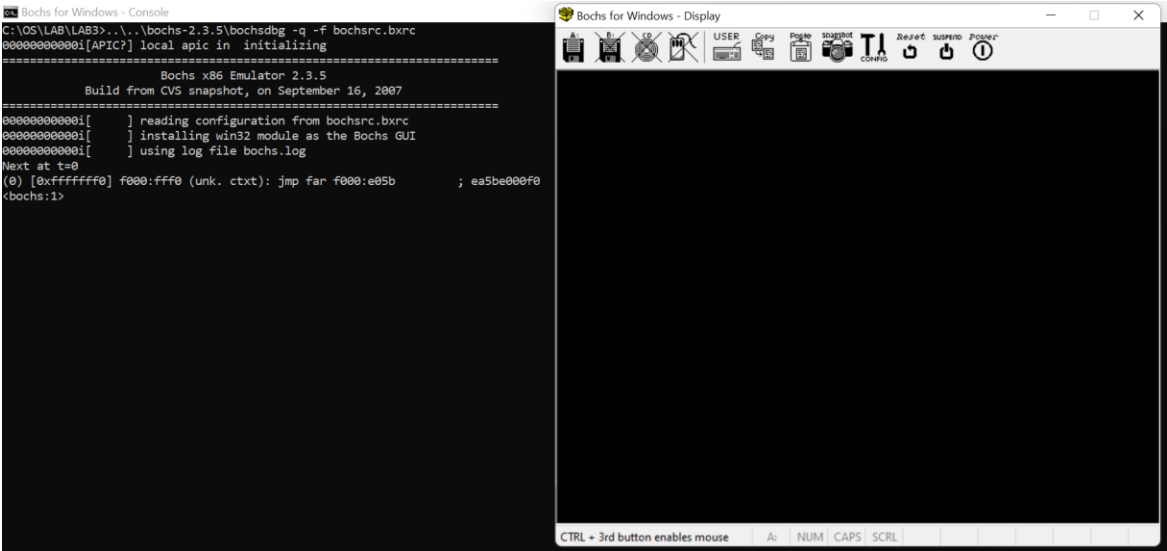
1. Masuk ke CMD kemudian lanjutkan dengan ‘CD OS’, ‘setpath’ dan ‘cd LAB/LAB3’



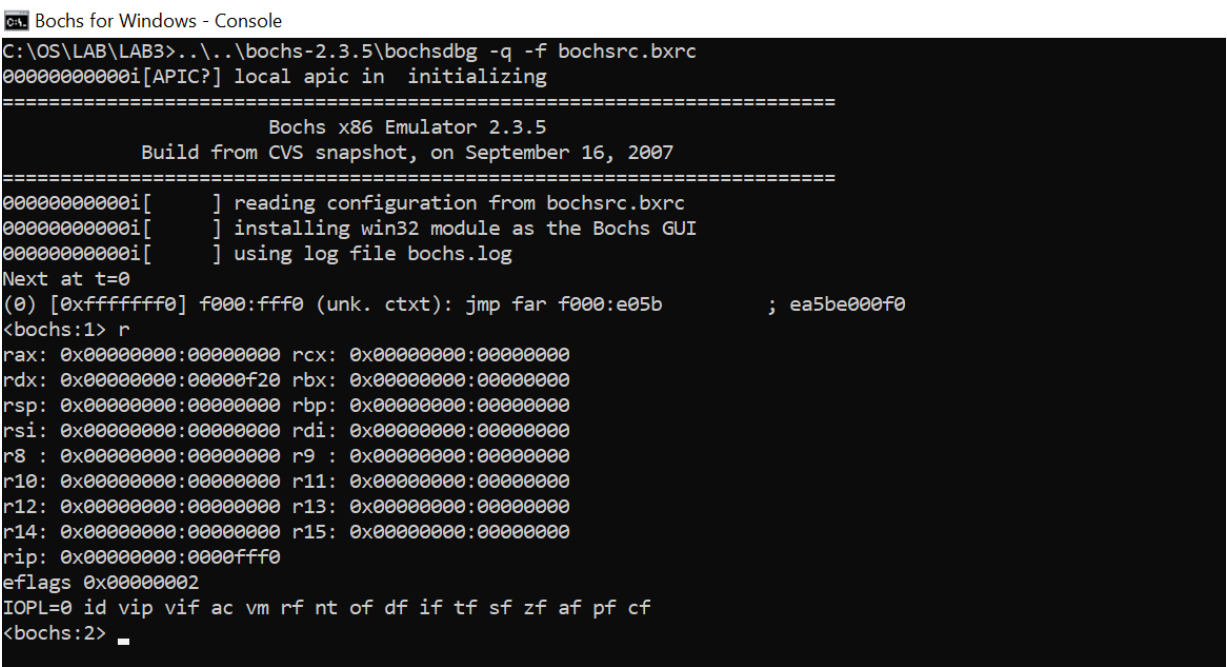
2. Mengetik ‘type s.bat’



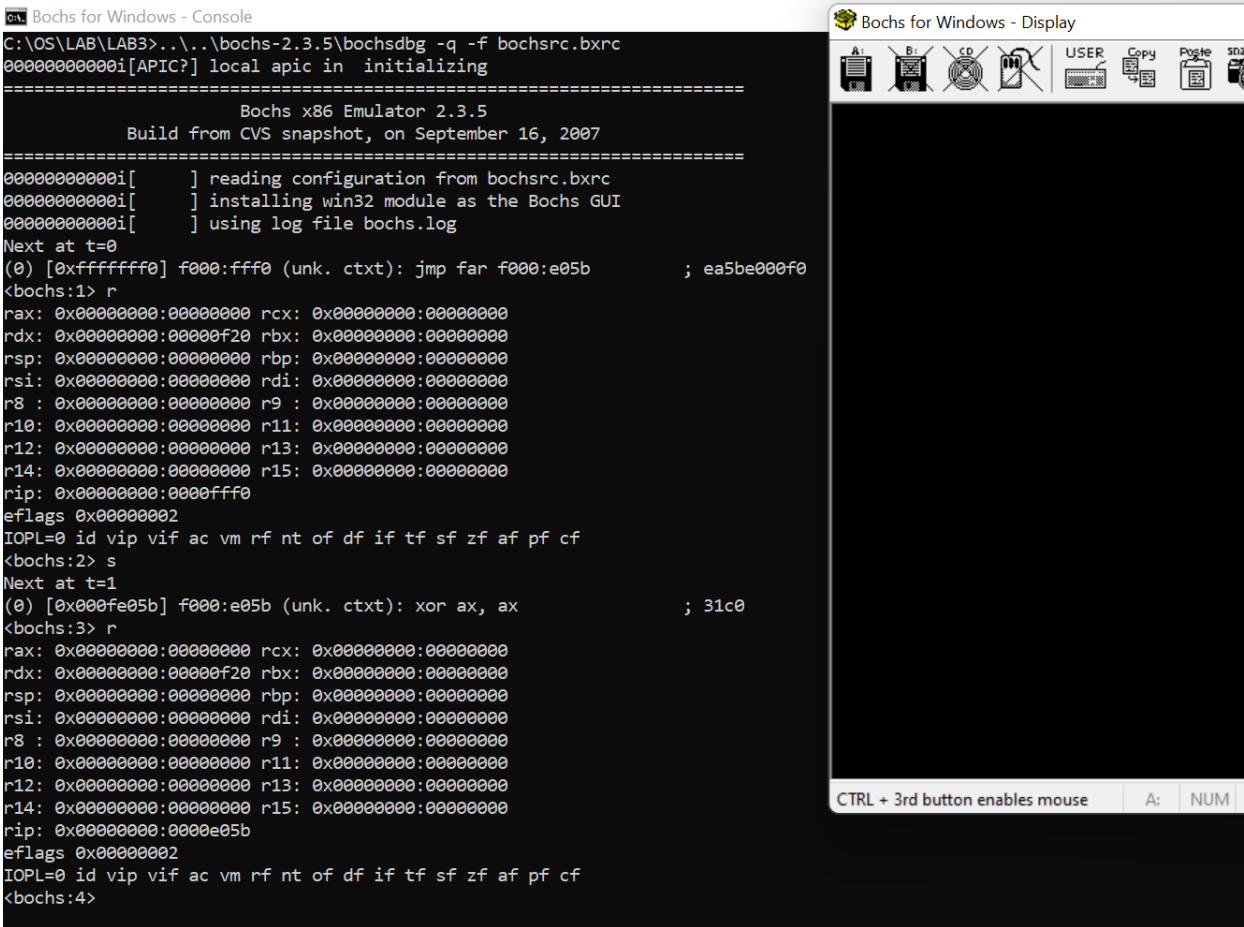
3. Mulai melakukan ‘debugging’ dengan memasukkan perintah ‘S’



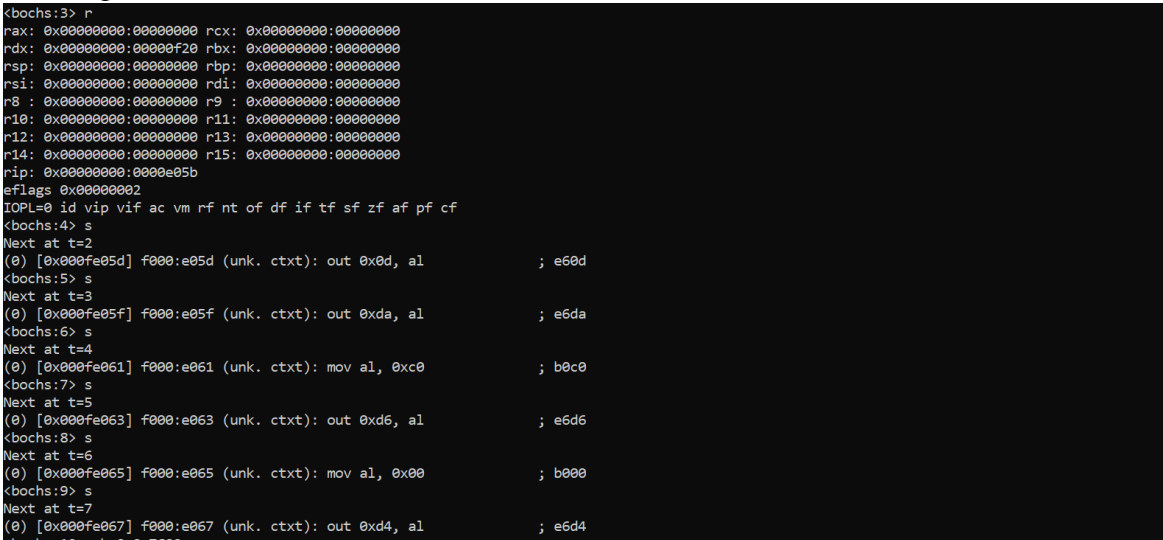
4. Melihat isi register CS dan IP dengan perintah ‘r’



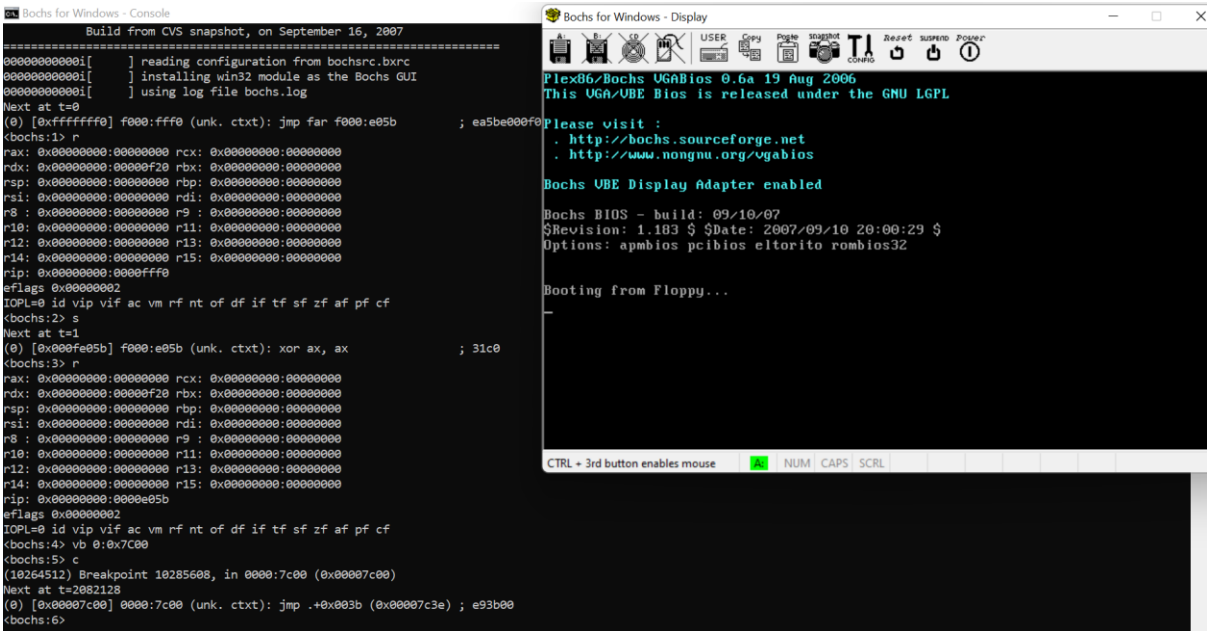
5. PC mengeksekusi perintah tersebut, dengan perintah ‘s’ dan dilanjutkan ‘r’



6. Mengetahui tahapan detail yang dilakukan PC dengan menjalankan perintah ‘s’ secara berulang,



7. Kemudian memasukkan break point dengan memasukkan perintah ‘vb 0:0x7C00’ dan memasukkan perintah ‘c’



8. Menjalankan PC langkah demi langkah (debugging) dengan perintah ‘s’

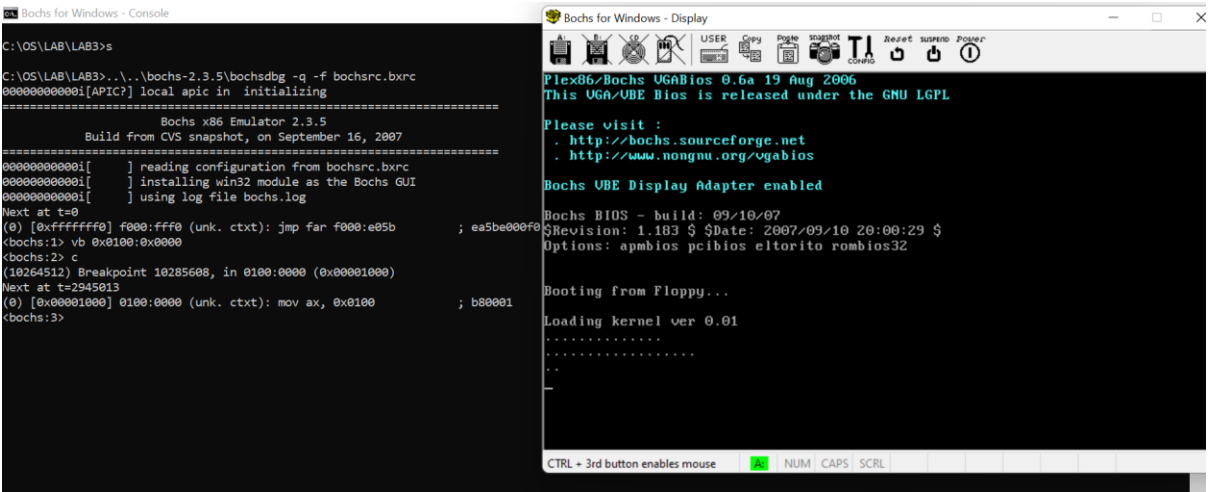
```
Bochs for Windows - Console
<bochs:4> vb 0:0x7C00
<bochs:5> c
(10264512) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> s
Next at t=2082129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:7> 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> h
h|help - show list of debugger commands
h|help command - show short command description
-- Debugger control --
help, q|quit|exit, set, instrument, show, trace, trace-reg,
record, playback, load-symbols, slist
-- Execution control --
c|cont|continue, s|step|stepi, p|n|next, modebp
-- Breakpoint management --
vb|vbreak, lb|lbreak, pb|pbreak|b|break, sb, sba, blist,
bpe, bpd, d|del|delete
-- CPU and memory contents --
x, xp, u|disasm|disassemble, r|reg|regs|registers, setpmem, crc, info,
set, dump_cpu, set_cpu, ptime, print-stack, watch, unwatch, ?|calc
<bochs:12> _
```

9. Menghentikan proses debugging dengan perintah ‘q’ dan mulai dari awal dengan ketik ‘s’ serta buat break point dengan perintah ‘vb 0x0100:0x0000’ kemudian selanjutnya mengetikkan perintah ‘c’

```
(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> q
# In bx_win32_gui_c::exit(void)!

Bochs is exiting. Press ENTER when you're ready to close this window.

C:\OS\LAB\LAB3> _
```



10. Meneruskan langkah PC simulator minimal sebanyak 10x, ketik ‘s’<ENTER> kemudian step berikutnya dilakukan dengan menekan tombol <ENTER> secara langsung.

