PRAKTIKUM SISTEM OPERASI

MODUL 1 : PENGENALAN SISTEM PENGEMBANGAN OS DENGAN PC SIMULATOR ‘BOCHS’

Disusun oleh:

AFIFAH GHAISANI IMANA

L200190198

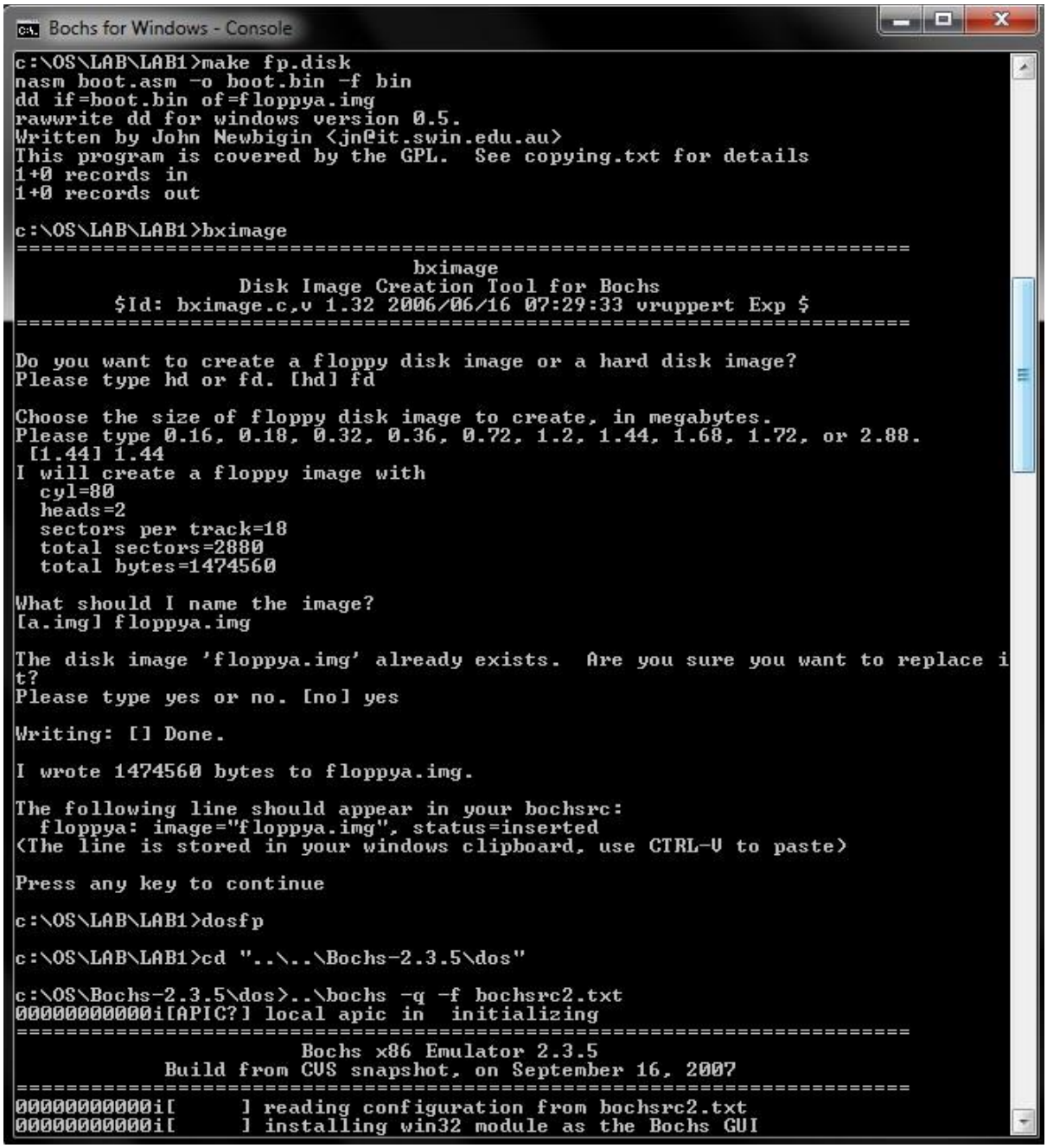
PROGRAM STUDI TEKNIK INFORMATIKA

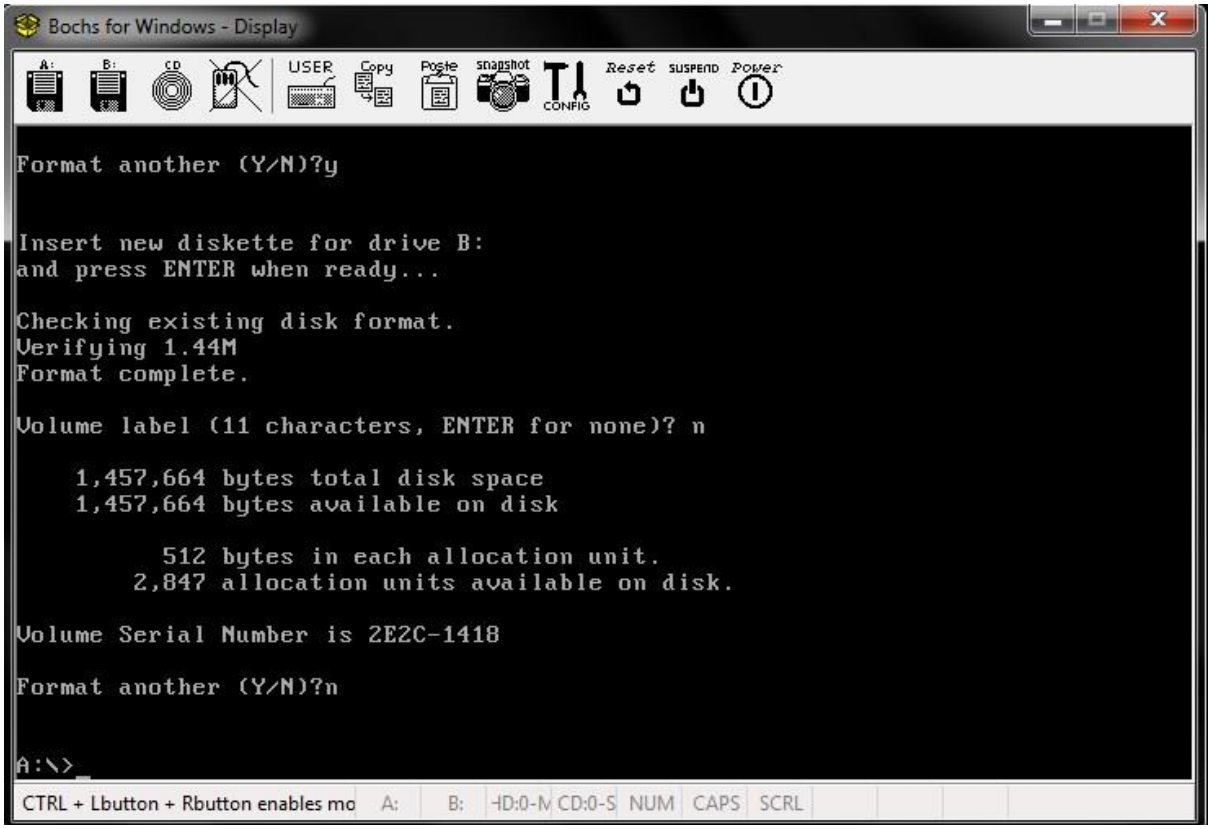
FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

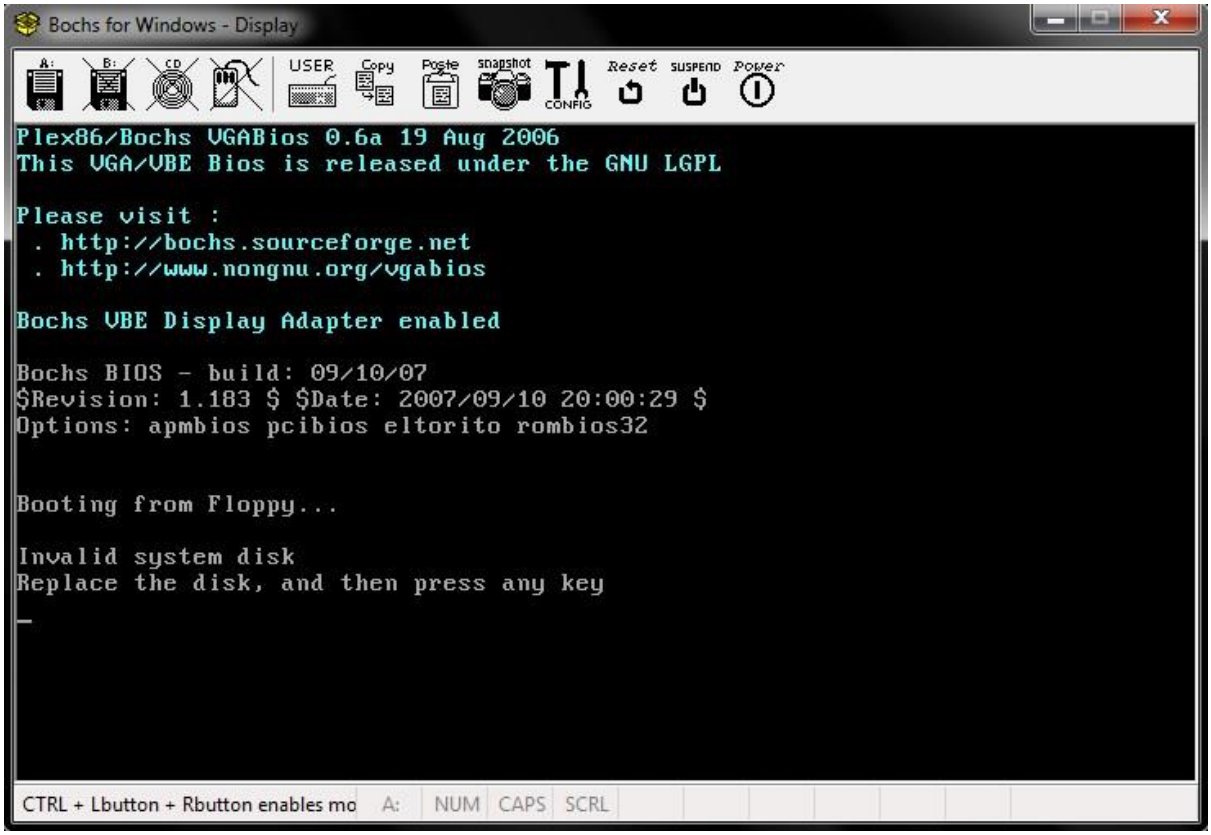
TAHUN 2019/2020

LANGKAH-LANGKAH :

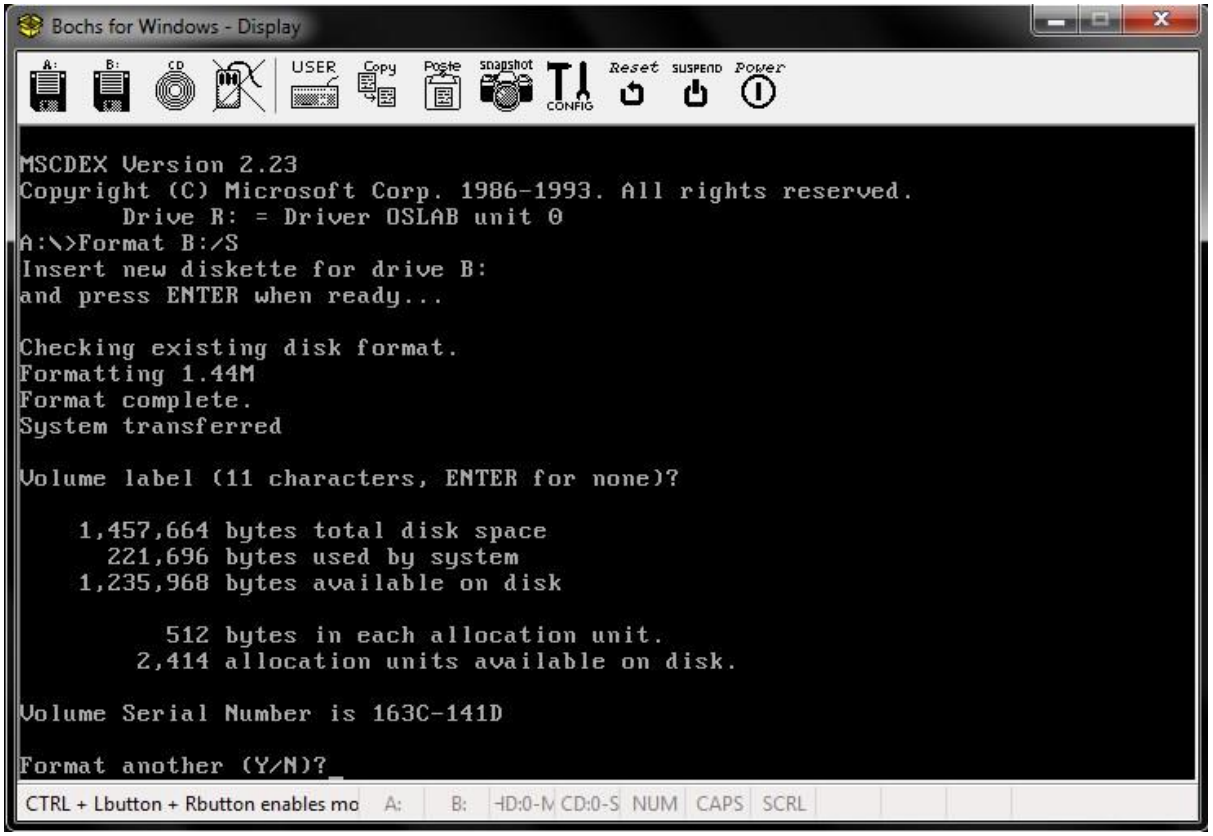


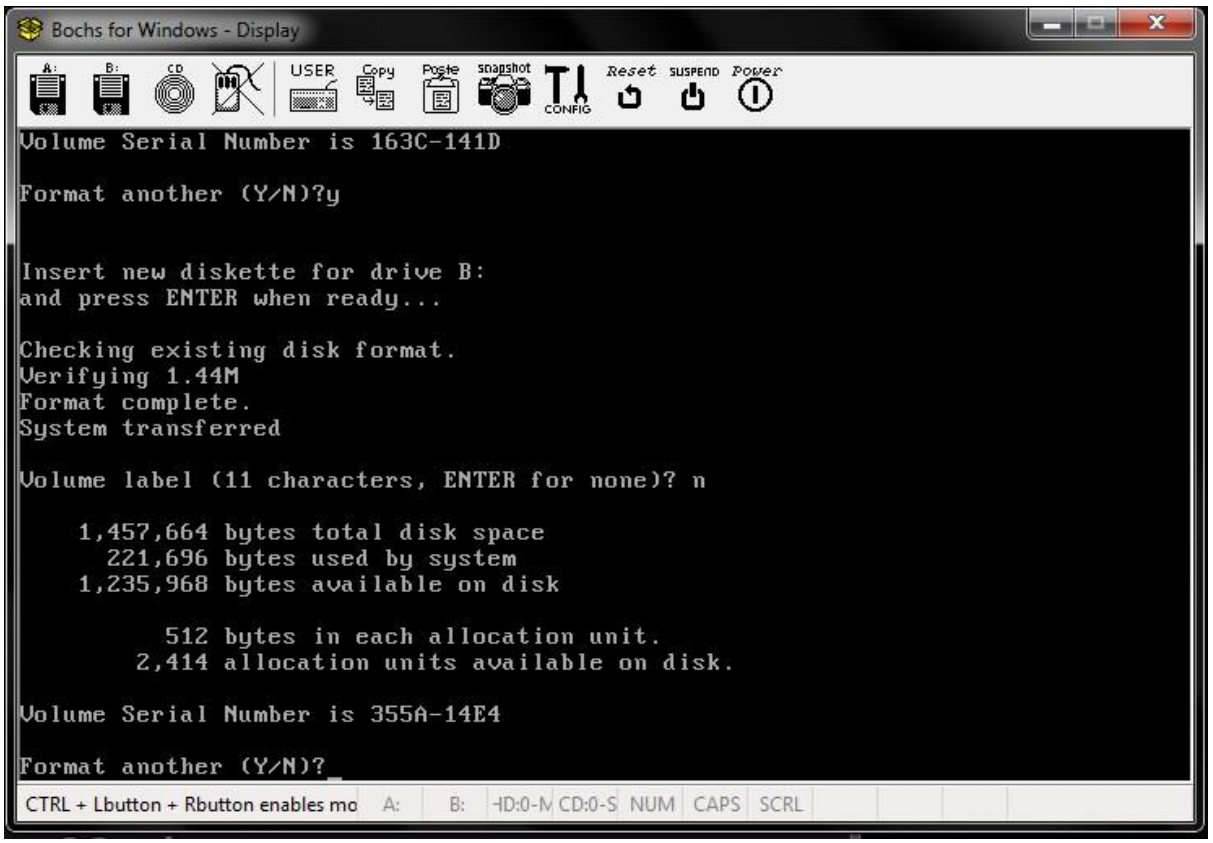

Bochs for Windows - Display
A:
B:
CD
USER
Copy
Paste
snapshot
CONFIG
Reset
SUSPEND
Power
Format another (Y/N)?y
Insert new diskette for drive B:
and press ENTER when ready...
Checking existing disk format.
Verifying 1.44M
Format complete.
Volume label (11 characters, ENTER for none)? n
1,457,664 bytes total disk space
1,457,664 bytes available on disk
512 bytes in each allocation unit.
2,847 allocation units available on disk.
Volume Serial Number is 2E2C-1418
Format another (Y/N)?n
A:\>_
CTRL + Lbutton + Rbutton enables mo
A:
B:
HD:0-M
CD:0-S
NUM
CAPS
SCRL

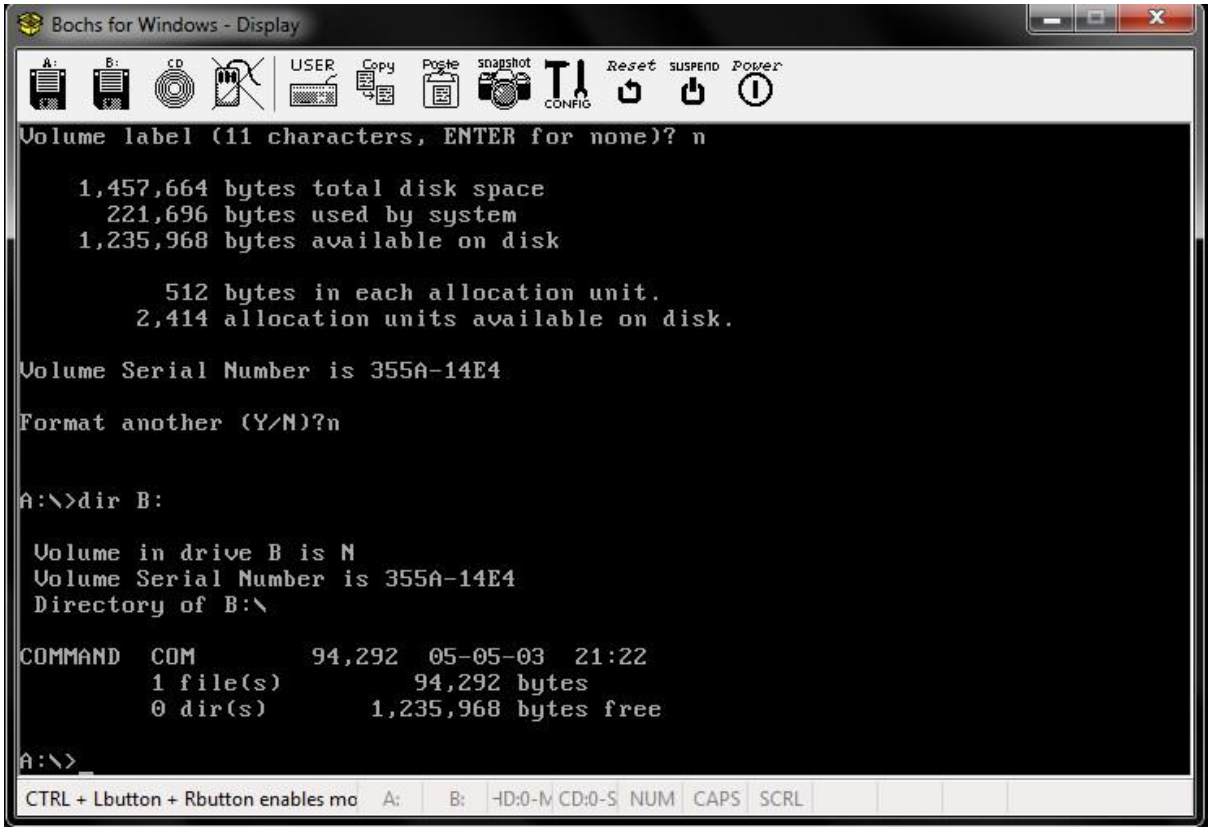

C:\Windows\system32\cmd.exe
c:\OS\LAB\LAB1>cd "..\..\Bochs-2.3.5\dos"
c:\OS\Bochs-2.3.5\dos>..\bochs -q -f bochsrc2.txt
00000000000i[APIC?] local apic in initializing
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
00000000000i[ ] reading configuration from bochsrc2.txt
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
In bx_win32_gui_c::exit(void)!
Bochs is exiting with the following message:
[WGUI] POWER button turned off.
c:\OS\Bochs-2.3.5\dos>cd "C:\os\lab\lab1"
C:\OS\LAB\LAB1>tdump boot.bin
Turbo Dump Version 5.0.16.12 Copyright (c) 1988, 2000 Inprise Corporation
Display of File BOOT.BIN
000000: E9 3B 00 51 55 41 53 49 2D 4F 53 00 02 01 01 00 .;.QUASI-OS.....
000010: 02 E0 00 40 0B F0 09 00 12 00 02 00 00 00 00 00 ...@............
000020: 00 00 00 00 00 00 29 FF FF FF FF 51 55 41 53 49 ......)....QUASI
000030: 20 20 42 4F 4F 54 46 41 54 31 32 20 20 20 FA B8   BOOTFAT12   ..
000040: C0 07 8E D8 8E C0 8E E0 8E E8 B8 00 00 8E D0 BC ....
000050: FF FF FB BE AE 01 E8 CB 00 31 C9 31 D2 B8 20 00 ....1.1.. .
000060: F7 26 11 00 F7 36 0B 00 91 A0 10 00 F7 26 16 00 .&...6.......&..
000070: 03 06 0E 00 A3 9F 01 01 0E 9F 01 BB 00 02 E8 B3 ....
000080: 00 8B 0E 11 00 BF 00 02 51 B9 0B 00 BE A3 01 57 ....Q......W
000090: F3 A6 5F 74 0A 59 81 C7 20 00 E2 EC E9 79 00 BE .._t.Y.. ....y..
0000A0: CB 01 E8 7F 00 8B 55 1A 89 16 A1 01 31 C0 A0 10 ......U.....1...
0000B0: 00 F7 26 16 00 89 C1 A1 0E 00 BB 00 02 E8 74 00 ..&...........t.
0000C0: BE CB 01 E8 5E 00 B8 00 01 8E C0 BB 00 00 53 A1 ....^.........S.
0000D0: A1 01 5B E8 9C 00 31 C9 8A 0E 0D 00 E8 55 00 53 ..[...1......U.S
0000E0: A1 A1 01 89 C1 89 C2 D1 EA 01 D1 BB 00 02 01 CB ....
0000F0: 8B 17 A9 01 00 75 07 81 E2 FF 0F E9 03 00 C1 EA .....u..........
000100: 04 89 16 A1 01 81 FA F0 0F 72 C4 BE CB 01 E8 13 ....r......
000110: 00 68 00 01 68 00 00 CB BE D0 01 E8 06 00 B4 00 .h..h...........
000120: CD 16 CD 19 AC 08 C0 74 0A B4 0E B7 00 B3 07 CD ....t........
000130: 10 EB F1 C3 BF 05 00 50 53 51 E8 45 00 B4 02 B0 .......PSQ.E....
000140: 01 8A 2E 9E 01 8A 0E 9C 01 8A 36 9D 01 8A 16 24 ....6....$
000150: 00 CD 13 73 0C 31 C0 CD 13 4F 59 5B 58 75 D8 CD ...s.1...OY[Xu..
000160: 18 BE CE 01 E8 BD FF 59 5B 58 03 1E 0B 00 40 E2 ....Y[X....@.
000170: C3 C3 2D 02 00 31 C9 8A 0E 0D 00 F7 E1 03 06 9F ..-..1..........
000180: 01 C3 31 D2 F7 36 18 00 FE C2 88 16 9C 01 31 D2 ..1..6........1.
000190: F7 36 1A 00 88 16 9D 01 A2 9E 01 C3 00 00 00 00 .6..............
0001A0: 00 00 00 4B 45 52 4E 45 4C 20 20 42 49 4E 0D 0A ...KERNEL BIN..
0001B0: 4C 6F 61 64 69 6E 67 20 6B 65 72 6E 65 6C 20 76 Loading kernel v
0001C0: 65 72 20 30 2E 30 31 20 0D 0A 00 0D 0A 00 2E 00 er 0.01 ....
0001D0: 0D 0A 45 52 52 4F 52 20 3A 20 50 72 65 73 73 20 ..ERROR : Press
0001E0: 41 6E 79 20 4B 65 79 20 74 6F 20 52 65 62 6F 6F Any Key to Reboo
0001F0: 74 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA t.............U.
C:\OS\LAB\LAB1>




```
C:\OS\LAB\LAB1>del floppya.img

C:\OS\LAB\LAB1>bximage
========================================================================
                                bximage
                  Disk Image Creation Tool for Bochs
        $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
========================================================================

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
 [1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560

What should I name the image?
[a.img] floppya.img

Writing: [] Done.

I wrote 1474560 bytes to floppya.img.

The following line should appear in your bochsrc:
  floppya: image="floppya.img", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)

Press any key to continue
```


Bochs for Windows - Display
A:
B:
CD
USER
Copy
Paste
snapshot
CONFIG
Reset
SUSPEND
Power
MSCDEX Version 2.23
Copyright (C) Microsoft Corp. 1986-1993. All rights reserved.
Drive R: = Driver OSLAB unit 0
A:\>Format B:/S
Insert new diskette for drive B:
and press ENTER when ready...
Checking existing disk format.
Formatting 1.44M
Format complete.
System transferred
Volume label (11 characters, ENTER for none)?
1,457,664 bytes total disk space
221,696 bytes used by system
1,235,968 bytes available on disk
512 bytes in each allocation unit.
2,414 allocation units available on disk.
Volume Serial Number is 163C-141D
Format another (Y/N)?
CTRL + Lbutton + Rbutton enables mo
A:
B:
CD:0-S
NUM
CAPS
SCRL


Bochs for Windows - Display
A:
B:
CD
USER
Copy
Paste
snapshot
CONFIG
Reset
SUSPEND
Power
Volume Serial Number is 163C-141D
Format another (Y/N)?y
Insert new diskette for drive B:
and press ENTER when ready...
Checking existing disk format.
Verifying 1.44M
Format complete.
System transferred
Volume label (11 characters, ENTER for none)? n
1,457,664 bytes total disk space
221,696 bytes used by system
1,235,968 bytes available on disk
512 bytes in each allocation unit.
2,414 allocation units available on disk.
Volume Serial Number is 355A-14E4
Format another (Y/N)?
CTRL + Lbutton + Rbutton enables mo
A:
B:
CD:0-S
NUM
CAPS
SCRL




```
C:\OS\Bochs-2.3.5\dos>cd "C:\os\lab\lab1"

C:\OS\LAB\LAB1>s

C:\OS\LAB\LAB1>..\..\bochs-2.3.5\bochs -q -f bochsrc.bxrc
00000000000i[APIC?] local apic in  initializing
========================================================================
                    Bochs x86 Emulator 2.3.5
            Build from CVS snapshot, on September 16, 2007
========================================================================
00000000000i[     ] reading configuration from bochsrc.bxrc
00000000000i[     ] installing win32 module as the Bochs GUI
00000000000i[     ] using log file bochsout.txt
# In bx_win32_gui_c::exit(void)!
========================================================================
Bochs is exiting with the following message:
[WGUI  ] POWER button turned off.
========================================================================

C:\OS\LAB\LAB1>
```

TUGAS PRAKTIKUM

1. ASCII (American Standard Code for Information Interchange) merupakan Kode Standar Amerika untuk Pertukaran Informasi atau sebuah standar internasional dalam pengkodean huruf dan simbol seperti Unicode dan Hex tetapi ASCII lebih bersifat universal. Dalam bahasa komputer 0 dan 1 tidak ada cara lain untuk mewakili huruf dan karakter yang bukan nomer. Semuanya harus menggunakan 0 dan 1. Salah satu jalan untuk berbahasa dengan komputer dengan cara menggunakan tabel ASCII. Tabel ASCII merupakan tabel atau daftar yang bersi semua huruf dalam alfabet romawi ditambah beberapa karakter tambahan. Dalam tabel ini setiap karakter akan selalu diwakili oleh sejumlah kode yang sama.

KODE ASCII DALAM FORMAT ANGKA DESIMAL, BINARY DAN HEXADESIMAL SERTA KARAKTER DAN SIMBOL YANG DIKODEKAN.

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	sp
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

2. DAFTAR INSTRUKSI PERINTAH BAHASA ASSEMBLY UNTUK MESIN INTEL KELUARGA X86 LENGKAP

Instruksi	Keterangan Singkatan
ACALL	Absolute Call
ADD	Add
ADDC	Add with Carry
AJMP	Absolute Jump
ANL	AND Logic
CJNE	Compare and Jump if Not Equal
CLR	Clear
CPL	Complement
DA	Decimal Adjust
DEC	Decrement
DIV	Divide
DJNZ	Decrement and Jump if Not Zero
INC	Increment
JB	Jump if Bit Set
JBC	Jump if Bit Set and Clear Bit
JC	Jump if Carry Set
JMP	Jump to Address
JNB	Jump if Not Bit Set
JNC	Jump if Carry Not Set

JNZ	Jump if Accumulator Not Zero
JZ	Jump if Accumulator Zero
LCALL	Long Call
LJMP	Long Jump
MOV	Move from Memory
MOVC	Move from Code Memory
MOVX	Move from Extended Memory
MUL	Multiply
NOP	No Operation
ORL	OR Logic
POP	Pop Value From Stack
PUSH	Push Value Onto Stack
RET	Return From Subroutine
RETI	Return From Interrupt
RL	Rotate Left
RLC	Rotate Left through Carry
RR	Rotate Right
RRC	Rotate Right through Carry
SETB	Set Bit
SJMP	Short Jump
SUBB	Subtract With Borrow

SWAP	Swap Nibbles
XCH	Exchange Bytes
XCHD	Exchange Digits
XRL	Exclusive OR Logic

UNTUK YANG LEBIH JELAS DAN DETIL:

a. MOV

Perintah MOV adalah perintah untuk mengisi, memindahkan,memperbaruhi isi suatu register, variable ataupun lokasi memory, Adapun tata penulisan perintah MOV adalah :

MOV [operand A], [Operand B]

Contoh :

MOV AH,02

Operand A adalah Register AH

Operand B adalah bilangan 02

Hal yang dilakukan oleh komputer untuk perintah diatas adalahmemasukan 02 ke register AH.

b. INT (Interrupt)

Bila anda pernah belajar BASIC, maka pasti anda tidak asing lagi dengan perintah GOSUB. Perintah INT juga mempunyai cara kerja yang sama dengan GOSUB, hanya saja subroutine yang dipanggil telah disediakan oleh memory komputer yang terdiri 2 jenis yaitu :

- Bios Interrupt (interput yang disediakan oleh BIOS (INT 0 – INT 1F))

- Dos Interrupt (Interrupt yang disediakan oleh DOS (INT 1F – keatas))

c. Push

Adalah perintah untuk memasukan isi register pada stack, dengan tata penulisannya:POP [operand 16 bit]

d. Pop

perintah yang berguna untuk mengeluarkan isi dari register/variable dari stack,dengan tata penulisannya adalah : POP [operand 16 bit]

e. RIP (Register IP)

Perintah ini digunakan untuk memberitahu komputer untuk memulai memproses program dari titik tertentu.

f. A (Assembler)

Perintah Assembler berguna untuk tempat menulis program Assembler.

-A100

0FD8:100

g. RCX (Register CX)

Perintah ini digunakan untuk mengetahui dan memperbaruhi isi register CX yang merupakantempat penampungan panjang program yang sedang aktif