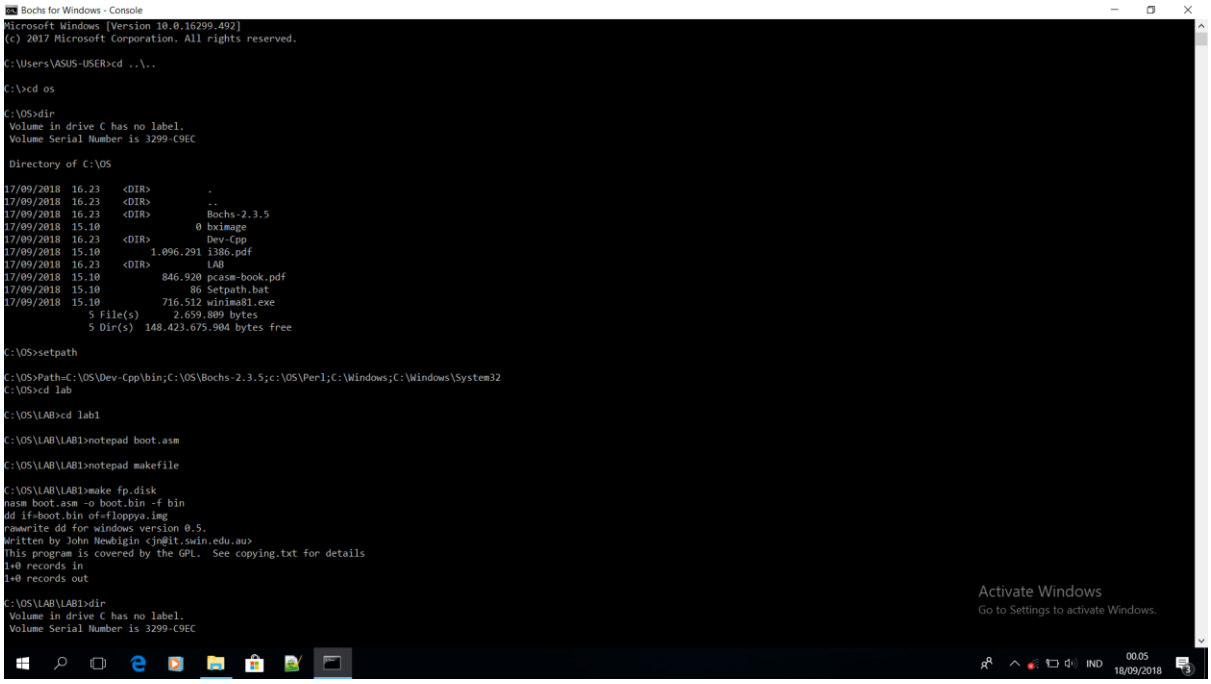
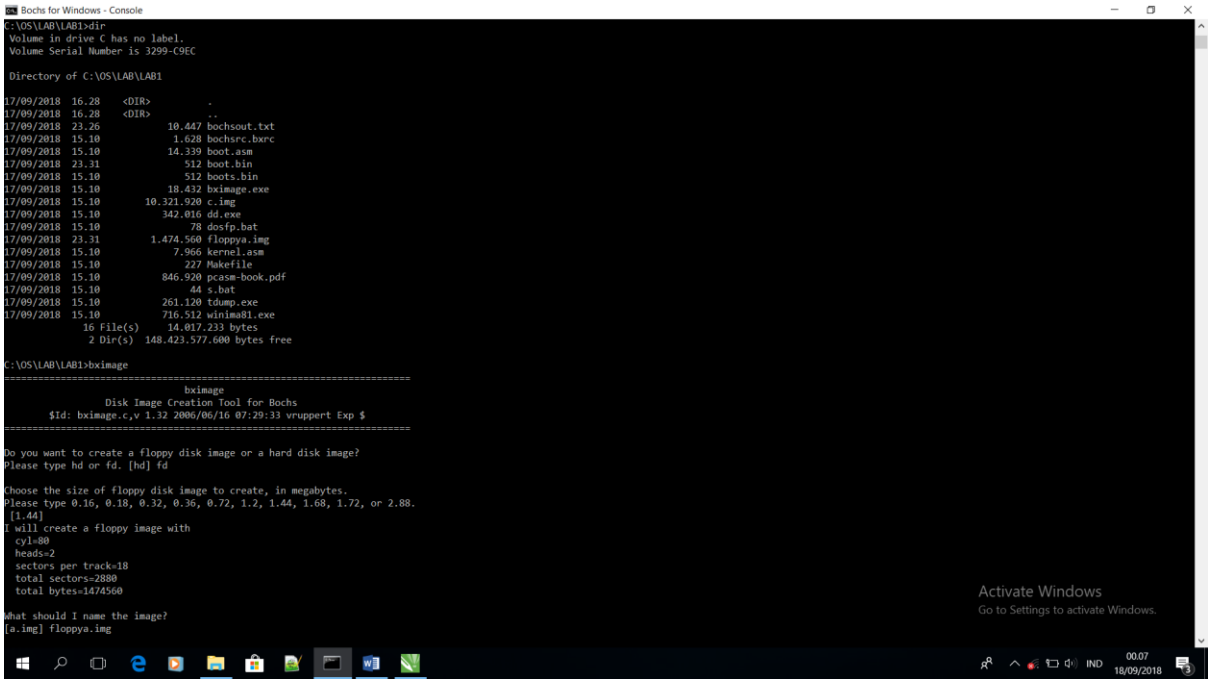
Nama : Rizky Tri Setya W

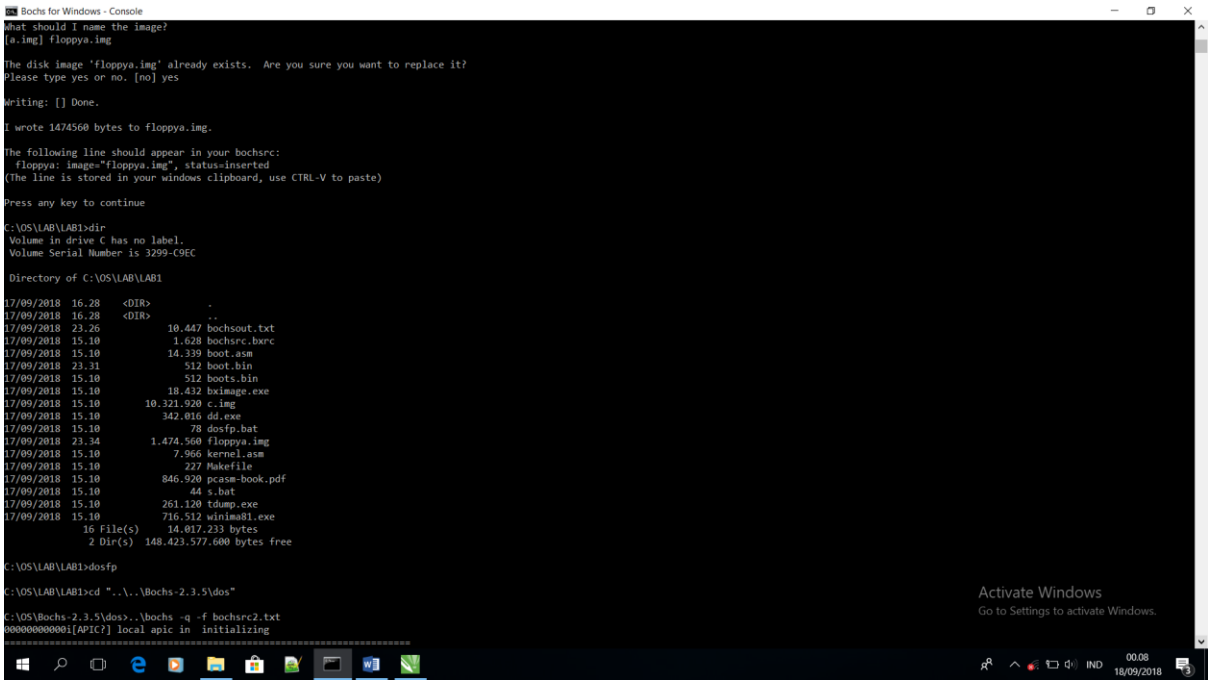
NIM : L200170054

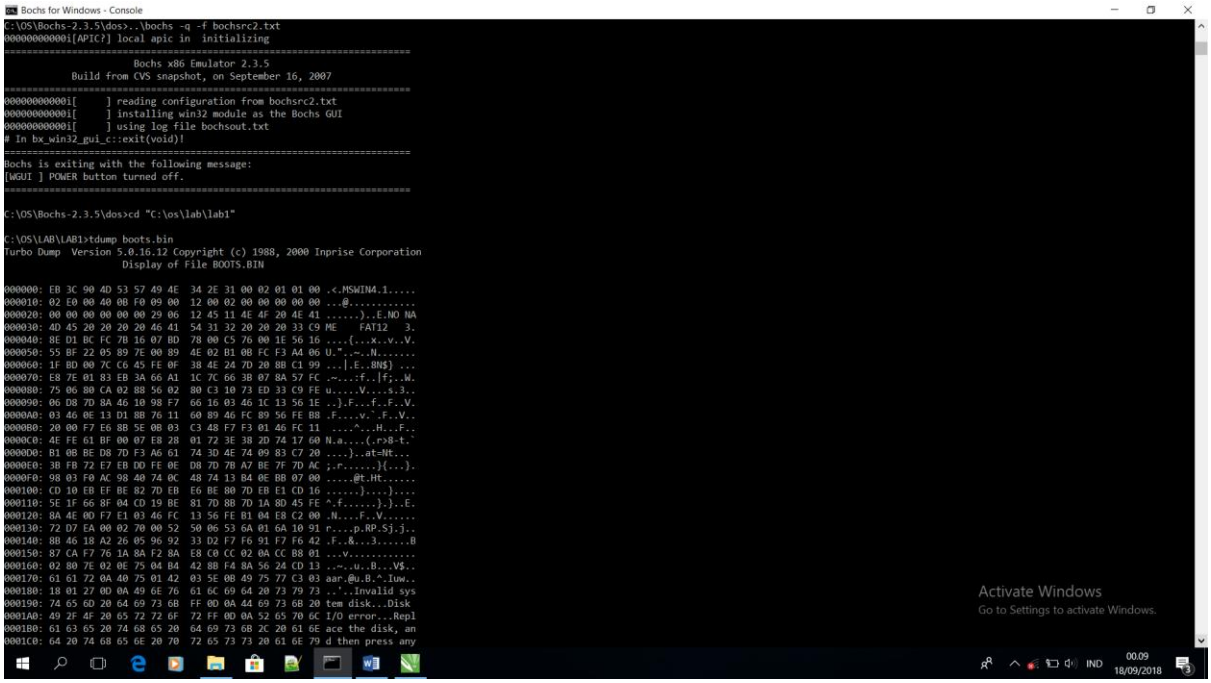
Kelas : C

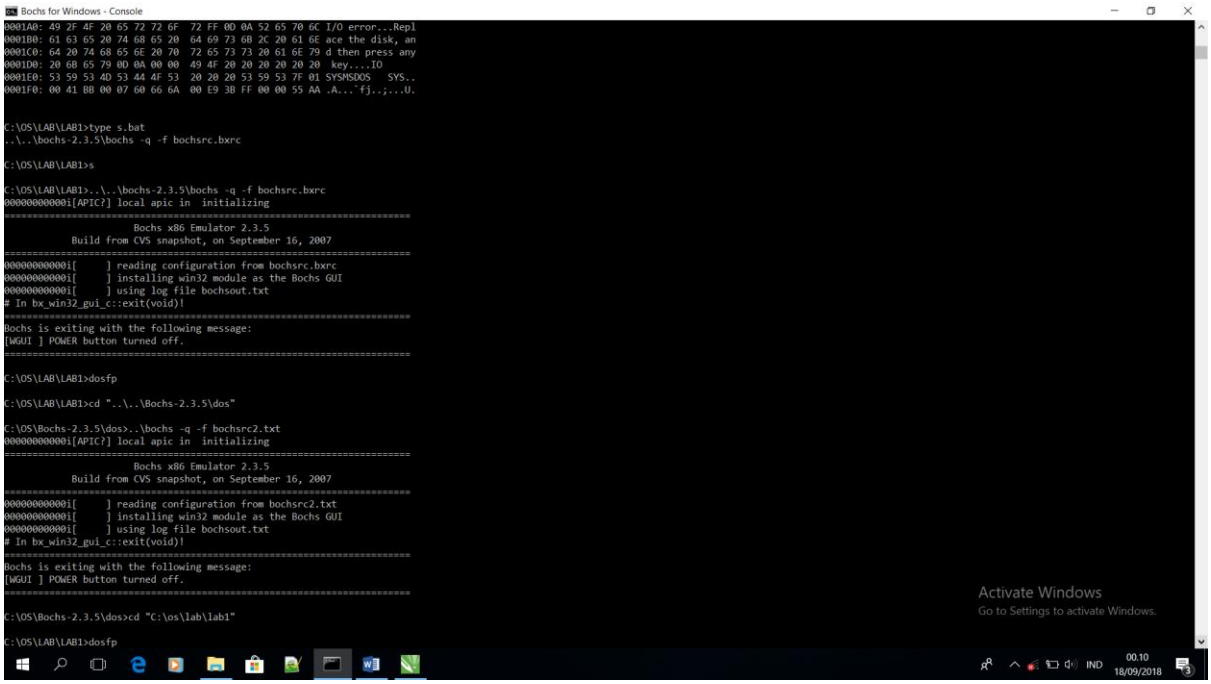
MODUL 1

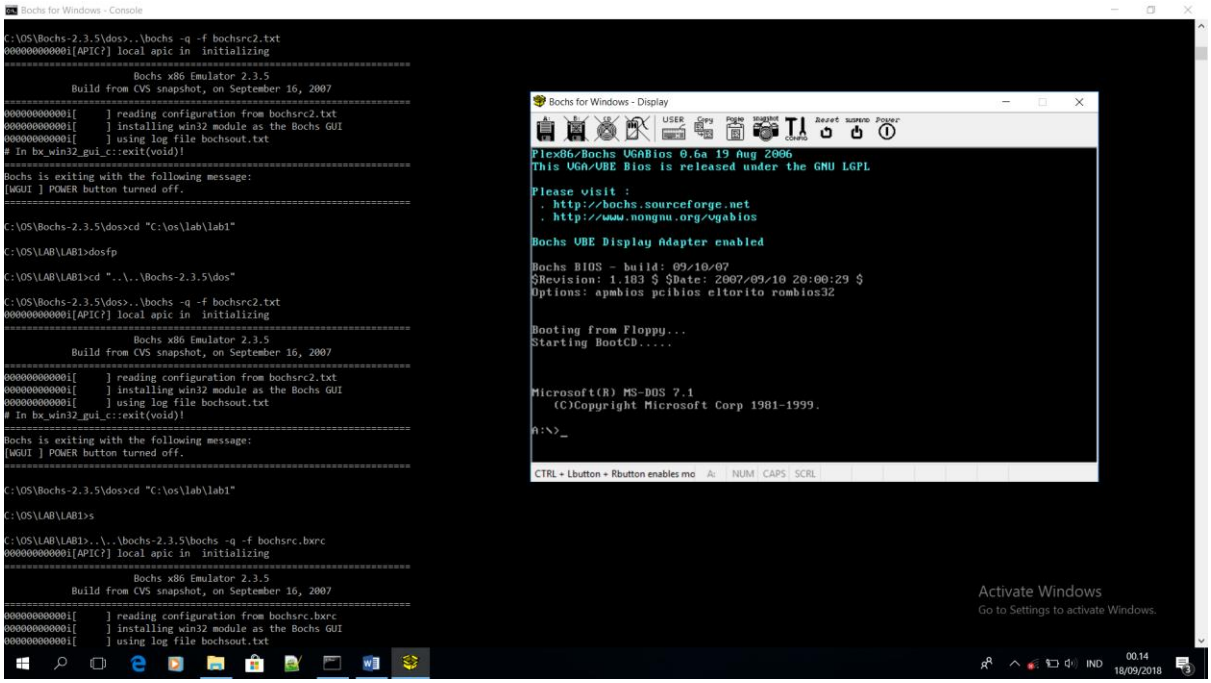





Bochs for Windows - Console
What should I name the image?
[a.img] floppya.img
The disk image 'floppya.img' already exists. Are you sure you want to replace it?
Please type yes or no. [no] yes
Writing: [] Done.
I wrote 1474560 bytes to floppya.img.
The following line should appear in your bochsrc:
floppya: image="floppya.img", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)
Press any key to continue
C:\OS\LAB\LAB1>dir
Volume in drive C has no label.
Volume Serial Number is 3299-C9EC
Directory of C:\OS\LAB\LAB1
17/09/2018 16.28 <DIR> .
17/09/2018 16.28 <DIR> ..
17/09/2018 23.26 10.447 bochsout.txt
17/09/2018 15.10 1.628 bochsrc.bxrc
17/09/2018 15.10 14.339 boot.asm
17/09/2018 23.31 512 boot.bin
17/09/2018 15.10 512 boots.bin
17/09/2018 15.10 18.432 bximage.exe
17/09/2018 15.10 10.321.920 c.img
17/09/2018 15.10 342.016 dd.exe
17/09/2018 15.10 78 dosfp.bat
17/09/2018 23.34 1.474.560 floppya.img
17/09/2018 15.10 7.966 kernel.asm
17/09/2018 15.10 227 Makefile
17/09/2018 15.10 846.920 pcasm-book.pdf
17/09/2018 15.10 44 s.bat
17/09/2018 15.10 261.120 tdump.exe
17/09/2018 15.10 716.512 winima81.exe
16 File(s) 14.017.233 bytes
2 Dir(s) 148.423.577.600 bytes free
C:\OS\LAB\LAB1>dosfp
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
Activate Windows
Go to Settings to activate Windows.
IND
00.08
18/09/2018


Bochs for Windows - Console
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
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 boots.bin
Turbo Dump Version 5.0.16.12 Copyright (c) 1988, 2000 Inprise Corporation
Display of File BOOTS.BIN
000000: EB 3C 90 4D 53 57 49 4E 34 2E 31 00 02 01 01 00 .<.MSWIN4.1.....
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 06 12 45 11 4E 4F 20 4E 41 ......)..E.NO NA
000030: 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 33 C9 ME FAT12 3.
000040: 8E D1 BC FC 7B 16 07 BD 78 00 C5 76 00 1E 56 16 ....{...x..v..V.
000050: 55 BF 22 05 89 7E 00 89 4E 02 B1 0B FC F3 A4 06 U."..~..N.......
000060: 1F BD 00 7C C6 45 FE 0F 38 4E 24 7D 20 8B C1 99 ...|.E..8N$} ...
000070: E8 7E 01 83 EB 3A 66 A1 1C 7C 66 3B 07 8A 57 FC .~...:f..|f;..W.
000080: 75 06 80 CA 02 88 56 02 80 C3 10 73 ED 33 C9 FE u.....V....s.3..
000090: 06 D8 7D 8A 46 10 98 F7 66 16 03 46 1C 13 56 1E ..}.F...f..F..V.
0000A0: 03 46 0E 13 D1 8B 76 11 60 89 46 FC 89 56 FE B8 .F....v.`.F..V..
0000B0: 20 00 F7 E6 8B 5E 0B 03 C3 48 F7 F3 01 46 FC 11 ....^...H...F..
0000C0: 4E FE 61 BF 00 07 E8 28 01 72 3E 38 2D 74 17 60 N.a....(.r>8-t.`
0000D0: B1 0B BE D8 7D F3 A6 61 74 3D 4E 74 09 83 C7 20 ....}..at=Nt...
0000E0: 3B FB 72 E7 EB DD FE 0E D8 7D 7B A7 BE 7F 7D AC ;.r......}{...}.
0000F0: 98 03 F0 AC 98 40 74 0C 48 74 13 B4 0E BB 07 00 .....@t.Ht......
000100: CD 10 EB EF BE 82 7D EB E6 BE 80 7D EB E1 CD 16 ......}....}....
000110: 5E 1F 66 8F 04 CD 19 BE 81 7D 8B 7D 1A 8D 45 FE ^.f.....}.}..E.
000120: 8A 4E 0D F7 E1 03 46 FC 13 56 FE B1 04 E8 C2 00 .N....F..V......
000130: 72 D7 EA 00 02 70 00 52 50 06 53 6A 01 6A 10 91 r....p.RP.Sj.j..
000140: 8B 46 18 A2 26 05 96 92 33 D2 F7 F6 91 F7 F6 42 .F..&...3......B
000150: 87 CA F7 76 1A 8A F2 8A E8 C0 CC 02 0A CC B8 01 ...v............
000160: 02 80 7E 02 0E 75 04 B4 42 8B F4 8A 56 24 CD 13 ..~..u..B...V$..
000170: 61 61 72 0A 40 75 01 42 03 5E 0B 49 75 77 C3 03 aar.@u.B.^.Iuw..
000180: 18 01 27 0D 0A 49 6E 76 61 6C 69 64 20 73 79 73 ..'..Invalid sys
000190: 74 65 6D 20 64 69 73 6B FF 0D 0A 44 69 73 6B 20 tem disk...Disk
0001A0: 49 2F 4F 20 65 72 72 6F 72 FF 0D 0A 52 65 70 6C I/O error...Repl
0001B0: 61 63 65 20 74 68 65 20 64 69 73 6B 2C 20 61 6E ace the disk, an
0001C0: 64 20 74 68 65 6E 20 70 72 65 73 73 20 61 6E 79 d then press any
Activate Windows
Go to Settings to activate Windows.
IND
00.09
18/09/2018





Tugas

1. Apa yang di maksud dengan kode ‘ASC II’, buatlahb tabel kode ASC II lengkap cukup kode ASC II yang standar tidak perlu extended, tuliskan kode ASC II dalam format angka desimal, binary dan hexadesimal serta karakter dan simbol yang dikodekan

2. Carilah daftar perintah bahasa assembly untuk mesin intel keluarga x86 lengkap(dari buku referensi atau internet). Daftar perintah ini dapat digunakan sebagai pedoman untuk memahami program ’boot.asm’ dan ‘kernel.asm’

Jawaban

1. **Kode Standar Amerika untuk Pertukaran Informasi** atau *American Standard Code for Information Interchange* (**ASCII**) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambakan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi logical communication, Device control, Information separator, Code extention, dan physical communication. Code ASCII ini banyak dijumpai pada papan ketik (keyboard) computer atau instrument-instrument digital.

Jumlah kode ASCII adalah 255 kode. Kode ASCII 0..127 merupakan kode ASCII untuk manipulasi teks; sedangkan kode ASCII 128..255 merupakan kode ASCII untuk manipulasi grafik. Kode ASCII sendiri dapat dikelompokkan lagi kedalam beberapa bagian:

- Kode yang tidak terlihat simbolnya seperti Kode 10(Line Feed), 13(Carriage Return), 8(Tab), 32(Space)
- Kode yang terlihat simbolnya seperti abjad (A..Z), numerik (0..9), karakter khusus (~!@#$%^&*()_+?:”{})
- Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik.

Dalam pengkodean kode ASCII memanfaatkan 8 bit. Pada saat ini kode ASCII telah tergantikan oleh kode UNICODE (Universal Code). UNICODE dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya.

Pada papan keyboard, aktifkan numlock, tekan tombol ALT secara bersamaan dengan kode karakter maka akan dihasilkan karakter tertentu. Misalnya: ALT + 44 maka akan muncul karakter koma (,). Mengetahui kode-kode ASCII sangat bermanfaat misalnya untuk membuat karakter-karakter tertentu yang tidak ada di keyboard.

Karakter	Nilai Unicode (heksadesimal)	Nilai ANSI ASCII (desimal)	Keterangan
NUL	0000	0	Null (tidak tampak)

SOH	0001	1	Start of heading (tidak tampak)
STX	0002	2	Start of text (tidak tampak)
ETX	0003	3	End of text (tidak tampak)
EOT	0004	4	End of transmission (tidak tampak)
ENQ	0005	5	Enquiry (tidak tampak)
ACK	0006	6	Acknowledge (tidak tampak)
BEL	0007	7	Bell (tidak tampak)
BS	0008	8	Menghapus satu karakter di belakang kursor (Backspace)
HT	0009	9	Horizontal tabulation
LF	000A	10	Pergantian baris (Line feed)
VT	000B	11	Tabulasi vertikal
FF	000C	12	Pergantian baris (Form feed)
CR	000D	13	Pergantian baris (carriage return)
SO	000E	14	Shift out (tidak tampak)
SI	000F	15	Shift in (tidak tampak)
DLE	0010	16	Data link escape (tidak tampak)

DC1	0011	17	Device control 1 (tidak tampak)
DC2	0012	18	Device control 2 (tidak tampak)
DC3	0013	19	Device control 3 (tidak tampak)
DC4	0014	20	Device control 4 (tidak tampak)
NAK	0015	21	Negative acknowledge (tidak tampak)
SYN	0016	22	Synchronous idle (tidak tampak)
ETB	0017	23	End of transmission block (tidak tampak)
CAN	0018	24	Cancel (tidak tampak)
EM	0019	25	End of medium (tidak tampak)
SUB	001A	26	Substitute (tidak tampak)
ESC	001B	27	Escape (tidak tampak)
FS	001C	28	File separator
GS	001D	29	Group separator
RS	001E	30	Record separator
US	001F	31	Unit separator
SP	0020	32	Spasi

!	0021	33	Tanda seru (exclamation)
"	0022	34	Tanda kutip dua
#	0023	35	Tanda pagar (kres)
$	0024	36	Tanda mata uang dolar
%	0025	37	Tanda persen
&	0026	38	Karakter ampersand (&)
‘	0027	39	Karakter Apostrof
(	0028	40	Tanda kurung buka
)	0029	41	Tanda kurung tutup
*	002A	42	Karakter asterisk (bintang)
+	002B	43	Tanda tambah (plus)
,	002C	44	Karakter koma
-	002D	45	Karakter hyphen (strip)
.	002E	46	Tanda titik
/	002F	47	Garis miring (*slash*)
0	0030	48	Angka nol

1	0031	49	Angka satu
2	0032	50	Angka dua
3	0033	51	Angka tiga
4	0034	52	Angka empat
5	0035	53	Angka lima
6	0036	54	Angka enam
7	0037	55	Angka tujuh
8	0038	56	Angka delapan
9	0039	57	Angka sembilan
:	003A	58	Tanda titik dua
;	003B	59	Tanda titik koma
<	003C	60	Tanda lebih kecil
=	003D	61	Tanda sama dengan
>	003E	62	Tanda lebih besar
?	003F	63	Tanda tanya
@	0040	64	A keong (@)

A	0041	65	Huruf latin A kapital
B	0042	66	Huruf latin B kapital
C	0043	67	Huruf latin C kapital
D	0044	68	Huruf latin D kapital
E	0045	69	Huruf latin E kapital
F	0046	70	Huruf latin F kapital
G	0047	71	Huruf latin G kapital
H	0048	72	Huruf latin H kapital
I	0049	73	Huruf latin I kapital
J	004A	74	Huruf latin J kapital
K	004B	75	Huruf latin K kapital
L	004C	76	Huruf latin L kapital
M	004D	77	Huruf latin M kapital
N	004E	78	Huruf latin N kapital
O	004F	79	Huruf latin O kapital
P	0050	80	Huruf latin P kapital

Q	0051	81	Huruf latin Q kapital
R	0052	82	Huruf latin R kapital
S	0053	83	Huruf latin S kapital
T	0054	84	Huruf latin T kapital
U	0055	85	Huruf latin U kapital
V	0056	86	Huruf latin V kapital
W	0057	87	Huruf latin W kapital
X	0058	88	Huruf latin X kapital
Y	0059	89	Huruf latin Y kapital
Z	005A	90	Huruf latin Z kapital
[	005B	91	Kurung siku kiri
\	005C	92	Garis miring terbalik (*backslash*)
]	005D	93	Kurung sikur kanan
^	005E	94	Tanda pangkat
_	005F	95	Garis bawah (underscore)
`	0060	96	Tanda petik satu

a	0061	97	Huruf latin a kecil
b	0062	98	Huruf latin b kecil
c	0063	99	Huruf latin c kecil
d	0064	100	Huruf latin d kecil
e	0065	101	Huruf latin e kecil
f	0066	102	Huruf latin f kecil
g	0067	103	Huruf latin g kecil
h	0068	104	Huruf latin h kecil
i	0069	105	Huruf latin i kecil
j	006A	106	Huruf latin j kecil
k	006B	107	Huruf latin k kecil
l	006C	108	Huruf latin l kecil
m	006D	109	Huruf latin m kecil
n	006E	110	Huruf latin n kecil
o	006F	111	Huruf latin o kecil
p	0070	112	Huruf latin p kecil

q	0071	113	Huruf latin q kecil
r	0072	114	Huruf latin r kecil
s	0073	115	Huruf latin s kecil
t	0074	116	Huruf latin t kecil
u	0075	117	Huruf latin u kecil
v	0076	118	Huruf latin v kecil
w	0077	119	Huruf latin w kecil
x	0078	120	Huruf latin x kecil
y	0079	121	Huruf latin y kecil
z	007A	122	Huruf latin z kecil
{	007B	123	Kurung kurawal buka
¦	007C	124	Garis vertikal (pipa)
}	007D	125	Kurung kurawal tutup
~	007E	126	Karakter gelombang (tilde)
DEL	007F	127	Delete
	0080	128	Dicadangkan

	0081	129	Dicadangkan
	0082	130	Dicadangkan
	0083	131	Dicadangkan
IND	0084	132	Index
NEL	0085	133	Next line
SSA	0086	134	Start of selected area
ESA	0087	135	End of selected area
	0088	136	Character tabulation set
	0089	137	Character tabulation with justification
	008A	138	Line tabulation set
PLD	008B	139	Partial line down
PLU	008C	140	Partial line up
	008D	141	Reverse line feed
SS2	008E	142	Single shift two
SS3	008F	143	Single shift three
DCS	0090	144	Device control string

PU1	0091	145	Private use one
PU2	0092	146	Private use two
STS	0093	147	Set transmit state
CCH	0094	148	Cancel character
MW	0095	149	Message waiting
	0096	150	Start of guarded area
	0097	151	End of guarded area
	0098	152	Start of string
	0099	153	Dicadangkan
	009A	154	Single character introducer
CSI	009B	155	Control sequence introducer
ST	009C	156	String terminator
OSC	009D	157	Operating system command
PM	009E	158	Privacy message
APC	009F	158	Application program command
	00A0	160	Spasi yang bukan pemisah kata

¡	00A1	161	Tanda seru terbalik
¢	00A2	162	Tanda sen (Cent)
£	00A3	163	Tanda Poundsterling
¤	00A4	164	Tanda mata uang (*Currency*)
¥	00A5	165	Tanda Yen
¦	00A6	166	Garis tegak putus-putus (*broken bar*)
§	00A7	167	Section sign
¨	00A8	168	Diaeresis
©	00A9	169	Tanda hak cipta (Copyright)
ª	00AA	170	Feminine ordinal indicator
«	00AB	171	Left-pointing double angle quotation mark
¬	00AC	172	Not sign
	00AD	173	Tanda strip (*hyphen*)
®	00AE	174	Tanda merk terdaftar
¯	00AF	175	Macron
°	00B0	176	Tanda derajat

±	00B1	177	Tanda kurang lebih (plus-minus)
²	00B2	178	Tanda kuadrat (pangkat dua)
³	00B3	179	Tanda kubik (pangkat tiga)
´	00B4	180	Acute accent
µ	00B5	181	Micro sign
¶	00B6	182	Pilcrow sign
·	00B7	183	Middle dot

Dalam program bahasa assembly terdapat 2 jenis yang kita tulis dalam program :

1. Assembly Directive (yaitu merupakan kode yang menjadi arahan bagi assembler/compiler untuk menata program)
2. Instruksi (yaitu kode yang harus dieksekusi oleh CPU mikrokontroler dengan melakukan operasi tertentu sesuai dengan daftar yang sudah tertanam dalam CPU)

Daftar Assembly Directive

Assembly Directive	Keterangan
EQU	Pendefinisian konstanta
DB	Pendefinisian data dengan ukuran satuan 1 byte
DW	Pendefinisian data dengan ukuran satuan 1 word
DBIT	Pendefinisian data dengan ukuran satuan 1 bit
DS	Pemesanan tempat penyimpanan data di RAM
ORG	Inisialisasi alamat mulai program
END	Penanda akhir program
CSEG	Penanda penempatan di code segment
XSEG	Penanda penempatan di external data segment

DSEG	Penanda penempatan di internal direct data segment
ISEG	Penanda penempatan di internal indirect data segment
BSEG	Penanda penempatan di bit data segment
CODE	Penanda mulai pendefinisian program
XDATA	Pendefinisian external data
DATA	Pendefinisian internal direct data
IDATA	Pendefinisian internal indirect data
BIT	Pendefinisian data bit
#INCLUDE	Mengikutsertakan file program lain

Daftar Instruksi

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 adalah memasukan 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 merupakan tempat penampungan panjang program yang sedan aktif pun, ada yang demikian:

1. Definisi Stack
   Secara harfiah stack berarti tumpukan, yaitu bagian memori yang digunakan untuk menyimpan nilai suatu register untuk sementara, membentuk tumpukan

nilai. Stack dapat dibayangkan sebagai tabung memanjang (seperti tabung penyimpan koin). Sedangkan nilai suatu register dapat dibayangkan sebagai koin yang dapat dimasukkan dalam tabung tersebut. Jika ada data yang disimpan maka data-data tersebut akan bergeser ke arah memori rendah, dan akan bergeser kembali ke arah memori tinggi bila data yang disimpan telah diambil.

2. Perintah Perpindahan Data
   Terkait perpindahan data, bahasa assembler mempunyai beberapa perintah yang dapat dibedakan yaitu untuk memindahkan data tunggal seperti huruf atau angka dan untuk memindahkan data string yang berupa deretan huruf. Tetapi di sini hanya akan menjelaskan beberapa perintah yang dipakai dalam aplikasi.

   2.1. PUSH/POP
   Syntax :
   PUSH Reg16Bit
   POP Reg16Bit

   PUSH adalah perintah penyimpanan data ke memori stack secara langsung, dan untuk mengambil keluar nilai yang disimpan tersebut gunakan perintah POP. Nilai terakhir yang dimasukkan dalam stack, dengan perintah PUSH, akan terletak pada puncak tabung stack. Dan perintah POP pertama kali akan mengambil nilai pada stack yang paling atas kemudian nilai berikutnya, demikian seterusnya. Jadi nilai yang terakhir dimasukkan akan merupakan yang pertama dikeluarkan. Operasi ini dinamakan LIFO (Last In First Out). Perhatikan contoh berikut ini:

   ```
   push ax;
   push bx;
   push cx;
   mov ax, $31C;
   mov bx; $31D;
   mov cx, $31E;
   pop cx;
   pop bx;
   pop ax;
   ```

   2.2. MOV
   Syntax :

   MOV destination, source

   Digunakan untuk menyalin data dari memori/register ke memori/register atau dari data langsung ke register. Nilai pada source yang dipindahkan tidaklah berubah. Pada contoh di bawah, register al diberi nilai $31C kemudian nilai register al disalin ke register ax. Jadi sekarang nilai register al dan register ax adalah $31C.

   ```
   mov al, $31C;
   mov ax, al;
   ```

Hal-hal yang tidak boleh dilakukan dalam penyalinan data :

a. Penyalinan data antarregister segmen (ds, es, cs, ss)

mov ds, es ? tidak dibenarkan
Gunakan register general, misalnya register ax, sebagai perantara

mov ax, es
mov ds, ax
atau gunakan stack sebagai perantara
push es
pop ds

b. Penyalinan data secara langsung untuk register segmen (ds, es, cs, ss)

mov ds, $31C ? tidak dibenarkan
Gunakan register general, misalnya register ax, sebagai perantara
mov ax, $31C
mov ds, ax

c. Penyalinan data langsung antar memori
mov memB, memA ? tidak dibenarkan
Gunakan register general, misalnya register ax, sebagai perantara
mov ax, memA
mov memB, ax
d. Penyalinan data antarregister general yang berbeda daya tampungnya (8 bit dengan 16 bit) tanpa pointer
mov al, bx ? tidak dibenarkan

2.3. IN/OUT
Syntax :

IN Reg16Bit, port
OUT port, Reg16Bit

Untuk membaca data dari suatu port dan memasukkan nilainya ke dalam suatu register gunakan perintah IN. Dan perintah OUT digunakan untuk memasukkan suatu nilai ke dalam suatu port. Nilai yang akan dimasukkan diberikan pada register al/ax dan alamat port diberikan pada register dx. Pada contoh berikut ini, pertama kali register dx disimpan pada stack, menyalin nilai $31E pada register dx kemudian perintah IN akan membaca nilai pada register dx (port bernilai $31E) dan memasukkannya ke dalam register al. Dan terakhir nilai tersebut disalin ke variabel Data.

push dx
mov dx, $31E
in al, dx
mov Data, al
pop dx

Dan contoh berikut untuk memberi nilai ($8A) pada suatu port.

```
push dx
mov dx, $31E
mov al, $8A
out dx, al
```

pop dx

3. Operasi Aritmatika
3.1. Penjumlahan
Syntax :

```
ADD destination, source
ADC destination, source
INC destination
```

Perintah ADD akan menjumlahkan nilai pada destination dan source tanpa menggunakan carry (ADD), dimana hasil yang didapat akan ditaruh pada destination. Dalam bahasa pascal pernyataan ini sama dengan pernyataan destination := destination + source. Daya tampung destination dan sourceharus sama misalnya register al (8 bit) dan ah (8 bit), ax (16 bit) dan bx (16 bit). Perhatikan contoh berikut, nilai register ah sekarang menjadi $10 :

```
mov ah, $5;
mov al, $8;
add ah, al
```

Perintah ADC digunakan untuk menangani penjumlahan dengan hasil yang melebihi daya tampung destination yaitu dengan menggunakan carry (ADD), dalam bahasa pascal sama dengan pernyataan destination := destination + source + carry. Misalnya register ax (daya tampung 16 bit) diberi nilai $1234 dan bx (16 bit) diberi nilai $F221, penjumlahan kedua register ini adalah $10455. Jadi ada bit ke 17 padahal daya tampung register bx hanya 16 bit, penyelesaiannya adalah nilai bx = $0455 dengan carry flag = 1.

Perintah INC digunakan untuk operasi penjumlahan dengan nilai 1. Jadi nilai pada destination akan ditambah 1, seperti perintah destination := destination + 1dalam bahasa Pascal.

3.2. Pengurangan
Syntax :

```
SUB destination, source
SBB destination, source
DEC destination
```

Perintah SUB untuk mengurangkan 2 operand tanpa carry flag. Hasilnya diletakkan pada destination dalam bahasa pasca sama dengan pernyataan destination := destination – source. Untuk mengenolkan suatu register, kurangkan dengan dirinya sendiri seperti

contoh berikut ini. Pertama kali register ax bernilai $5, kemudian nilai register tersebut dikurangi dengan dirinya sendiri sehingga terakhir nilai register ax adalah 0.

```
mov ax, $15;
mov bx, $10;
sub ax, bx;
sub ax, ax;
```

Perintah SBB mengurangkan nilai destination dengan nilai source kemudian dikurangi lagi dengan carry flag (destination := destination – source – carry flag).

Dan perintah DEC untuk mengurangi nilai destination dengan 1.

3.3. Perkalian
Syntax :

MUL source

Digunakan untuk mengalikan data pada accumulator dengan suatu operand dan hasilnya diletak pada register source. Register source dapat berupa suatu register 8 bit (misal bl, bh, dan sebagainya), register 16 bit (bx, dx, dan sebagainya) atau suatu variabel.

3.4. Pembagian
Syntax :

DIV source
Operasi aritmatika ini pada dasarnya sama dengan operasi perkalian.