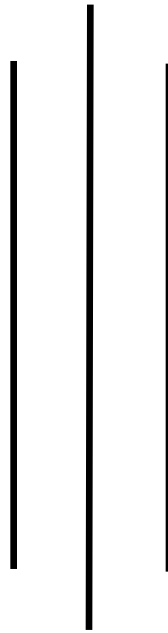




**LAPORAN PRAKTIKUM SISTEM OPERASI**  
**MODUL 1 : SIMULATOR 'BOCHS'**



**DISUSUN OLEH :**

**NAMA : BIMA TRIADMAJA**

**NIM : L200210137**

**KELAS : C**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**  
**UNIVERSITAS MUHAMMADIYAH SURAKARTA**  
**TAHUN 2021/2022**

Berikut adalah screenshot dari command prompt di laptop saya :

```
Bochs for Windows - Console
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Infinity>cd ../../..
C:\>cd os
C:\OS>dir
Volume in drive C is Acer
Volume Serial Number is D88D-A952

Directory of C:\OS

20/09/2022  15:47    <DIR>          .
20/09/2022  15:47    <DIR>          Bochs-2.3.5
20/09/2022  15:47    <DIR>          Dev-Cpp
17/12/2008  00:08             1.096.291 i386.pdf
20/09/2022  15:47    <DIR>          LAB
18/09/2019  04:48                0 notepad
17/12/2008  00:07             846.920 pcasm-book.pdf
17/12/2008  01:44                86 Setpath.bat
13/12/2008  14:12             716.512 winima81.exe
               5 File(s)      2.659.809 bytes
               4 Dir(s)      84.385.927.168 bytes free

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
C:\OS\LAB>cd lab1
C:\OS\LAB\LAB1>Notepad boot.asm
C:\OS\LAB\LAB1>
C:\OS\LAB\LAB1>Notepad M
C:\OS\LAB\LAB1>Notepad Makefile
C:\OS\LAB\LAB1>make fp.disk
```

```
Bochs for Windows - Console

C:\OS\LAB\LAB1>Notepad Makefile

C:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out

C:\OS\LAB\LAB1>dir
Volume in drive C is Acer
Volume Serial Number is D88D-A952

Directory of C:\OS\LAB\LAB1

20/09/2022  15:47    <DIR>          .
20/09/2022  15:47    <DIR>          ..
15/09/2022  09:21             1.474.560 a.img
14/09/2022  11:06             10.210 bochsout.txt
15/12/2008  16:17             1.628 bochsrc.bxrc
14/12/2008  12:02             14.365 boot.asm
11/09/2019  21:17                0 boot.asm
20/09/2022  15:55             512 boot.bin
13/09/2022  09:30             512 boots.bim
13/09/2022  09:16             512 boots.bin
15/12/2008  00:47                78 dosfp.bat
20/09/2022  15:55             1.474.560 floppya.img
14/12/2008  11:45             7.966 kernel.asm
15/12/2008  16:21             227 Makefile
15/12/2008  12:20                44 s.bat
               13 File(s)      2.985.174 bytes
               2 Dir(s)      84.383.354.880 bytes free

C:\OS\LAB\LAB1>del floppya.img /P
C:\OS\LAB\LAB1>floppya.img, Delete (Y/N)? Y
C:\OS\LAB\LAB1>dir
```



```
Bochs for Windows - Console
# In bx_win32_gui_c::exit(void)!
=====
Bochs is exiting with the following message:
[VGUI ] POWER button turned off.
=====

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

C:\OS\LAB\LAB1>jump 0x7C3E
'jump' is not recognized as an internal or external command,
operable program or batch file.

C:\OS\LAB\LAB1>dd if=floppya.img of=boots.bin count=1
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out

C:\OS\LAB\LAB1>debug boots.bin
'debug' is not recognized as an internal or external command,
operable program or batch file.

C:\OS\LAB\LAB1>boots.bin

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 29 EF 17 42 38 4E 4F 20 4E 41 .....B8NO 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 88 C1 99 ...|.E..8N$} ...
000070: E8 7E 01 83 EB 3A 66 A1 1C 7C 66 3B 07 8A 57 FC ~...f...fj..W.
000080: 75 06 80 CA 02 88 56 02 80 C3 10 73 ED 33 C9 FE u.....V...s.3..
000090: 06 D8 7D 8A 46 10 98 F7 66 16 03 46 1C 13 56 1E ..}.F...f..F..V.
0000A0: 03 46 0E 13 D1 8B 76 11 60 89 46 FC 89 56 FE B8 .F...v..F..V..
0000B0: 20 00 F7 E6 88 5E 00 03 C3 48 F7 F3 01 46 FC 11 ....^..H...F..
0000C0: 4E FE 61 BF 00 07 E8 28 01 72 3E 38 2D 74 17 60 N.a....(r>8-t.
0000D0: B1 0B BE D8 7D F3 A6 61 74 3D 4E 74 09 83 C7 20 .....at=Nt...
0000E0: 3B FB 72 E7 EB D0 FE 0E D8 7D 7B A7 BE 7F 7D AC ;r.....}{...}.
0000F0: 98 03 F0 AC 98 40 74 0C 48 74 13 B4 0E B8 07 00 .....@t.Ht.....
000100: CD 10 EB EF BE 82 7D EB E6 BE 80 7D EB E1 CD 16 .....)}.....}
000110: 5E 1F 66 8F 04 CD 19 BE 81 7D 8B 7D 1A 8D 45 FE ^f.....}.E..
000120: 8A 4E 0D F7 E1 03 46 FC 13 56 FE B1 04 E8 C2 00 .N...F..V.....
000130: 72 D7 EA 00 02 70 00 52 50 06 53 6A 01 6A 10 91 r....p.RP.Sj.j..
000140: 8B 46 18 A2 26 05 96 92 33 D2 F7 F6 91 F7 F6 42 .F..&...3...B
000150: 87 CA F7 76 1A 8A F2 8A E8 C0 CC 02 0A CC B8 01 ...V.....
000160: 02 80 7E 02 0E 75 04 B4 42 8B F4 8A 56 24 CD 13 ...~.u..B...V$.
000170: 61 61 72 0A 40 75 01 42 03 5E 0B 49 75 77 C3 03 aar.@.B.^Iuw..
000180: 18 01 27 00 0A 49 6E 76 61 6C 69 64 20 73 79 73 ...Invalid sys
000190: 74 65 6D 20 64 69 73 68 FF 0D 0A 44 69 73 68 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 68 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
0001D0: 20 68 65 79 0D 0A 00 00 49 4F 20 20 20 20 20 20 key....IO
0001E0: 53 59 53 4D 53 44 4F 53 20 20 20 53 59 53 7F 01 SYSHSDOS SYS..
0001F0: 00 41 B8 00 07 60 66 6A 00 E9 3B FF 00 00 55 AA .A...fj...;...U.
```

```
Bochs for Windows - Console
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 88 C1 99 ...|.E..8N$} ...
000070: E8 7E 01 83 EB 3A 66 A1 1C 7C 66 3B 07 8A 57 FC ~...f...fj..W.
000080: 75 06 80 CA 02 88 56 02 80 C3 10 73 ED 33 C9 FE u.....V...s.3..
000090: 06 D8 7D 8A 46 10 98 F7 66 16 03 46 1C 13 56 1E ..}.F...f..F..V.
0000A0: 03 46 0E 13 D1 8B 76 11 60 89 46 FC 89 56 FE B8 .F...v..F..V..
0000B0: 20 00 F7 E6 88 5E 00 03 C3 48 F7 F3 01 46 FC 11 ....^..H...F..
0000C0: 4E FE 61 BF 00 07 E8 28 01 72 3E 38 2D 74 17 60 N.a....(r>8-t.
0000D0: B1 0B BE D8 7D F3 A6 61 74 3D 4E 74 09 83 C7 20 .....at=Nt...
0000E0: 3B FB 72 E7 EB D0 FE 0E D8 7D 7B A7 BE 7F 7D AC ;r.....}{...}.
0000F0: 98 03 F0 AC 98 40 74 0C 48 74 13 B4 0E B8 07 00 .....@t.Ht.....
000100: CD 10 EB EF BE 82 7D EB E6 BE 80 7D EB E1 CD 16 .....)}.....}
000110: 5E 1F 66 8F 04 CD 19 BE 81 7D 8B 7D 1A 8D 45 FE ^f.....}.E..
000120: 8A 4E 0D F7 E1 03 46 FC 13 56 FE B1 04 E8 C2 00 .N...F..V.....
000130: 72 D7 EA 00 02 70 00 52 50 06 53 6A 01 6A 10 91 r....p.RP.Sj.j..
000140: 8B 46 18 A2 26 05 96 92 33 D2 F7 F6 91 F7 F6 42 .F..&...3...B
000150: 87 CA F7 76 1A 8A F2 8A E8 C0 CC 02 0A CC B8 01 ...V.....
000160: 02 80 7E 02 0E 75 04 B4 42 8B F4 8A 56 24 CD 13 ...~.u..B...V$.
000170: 61 61 72 0A 40 75 01 42 03 5E 0B 49 75 77 C3 03 aar.@.B.^Iuw..
000180: 18 01 27 00 0A 49 6E 76 61 6C 69 64 20 73 79 73 ...Invalid sys
000190: 74 65 6D 20 64 69 73 68 FF 0D 0A 44 69 73 68 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 68 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
0001D0: 20 68 65 79 0D 0A 00 00 49 4F 20 20 20 20 20 20 key....IO
0001E0: 53 59 53 4D 53 44 4F 53 20 20 20 53 59 53 7F 01 SYSHSDOS SYS..
0001F0: 00 41 B8 00 07 60 66 6A 00 E9 3B FF 00 00 55 AA .A...fj...;...U.

C:\OS\LAB\LAB1>type s.bat
...\\bochs-2.3.5\bochs -q -f bochs.rc.bxrc

C:\OS\LAB\LAB1>s
C:\OS\LAB\LAB1>...\\bochs-2.3.5\bochs -q -f bochs.rc.bxrc
0000000000i[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
```

```
Bochs for Windows - Console

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 ] Window closed, exiting!
=====

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

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
=====
                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"
```

```
Bochs for Windows - Console

=====

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

C:\OS\LAB\LAB1>dir B:
The system cannot find the path specified.

C:\OS\LAB\LAB1>DosFp

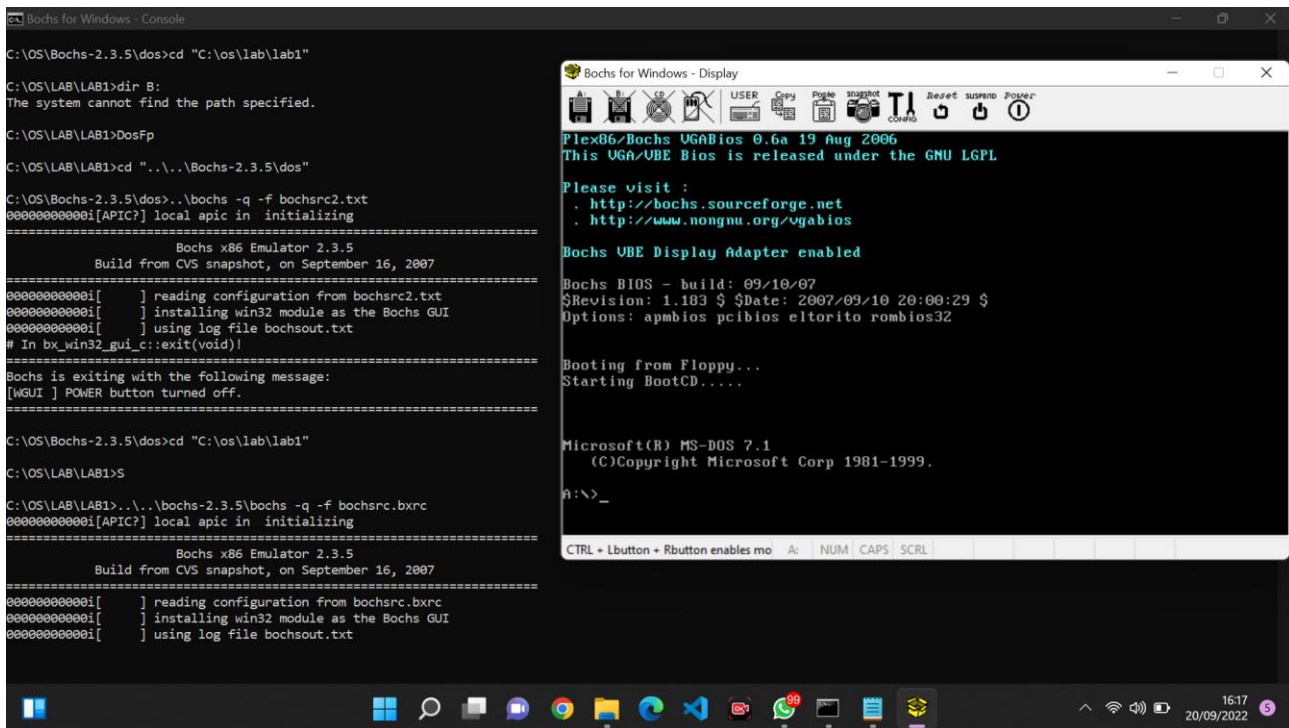
C:\OS\LAB\LAB1>cd "..\..\Bochs-2.3.5\dos"

C:\OS\Bochs-2.3.5\dos>..\bochs -q -f bochsrc2.txt
00000000000i[APIC?] local apic in  initializing
=====
                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>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
```



## Tugas

1. Apa yang dimaksud dengan kode 'ASCII', buatlah tabel kode ASCII lengkap cukup kode ASCII yang standar tidak perlu extended, tuliskan kode ASCII dalam format angka desimal, binary dan hexadesimal serta karakter dan simbol yang dikodekan.

Jawab :

(ASCII) merupakan singkatan dari American Standard Code for Information Interchange atau Kode Standar Amerika untuk Pertukaran Informasi, yaitu standar pengkodean karakter untuk alat komunikasi. Kode ASCII mewakili teks dalam komputer, peralatan telekomunikasi, dan perangkat lainnya. Kebanyakan skema pengkodean karakter modern didasarkan pada ASCII, meskipun mereka mendukung banyak karakter tambahan.

Kemudian ada tabel karakter ASCII, Tabel berikut berisi karakter-karakter ASCII . Dalam sistem operasi Windows dan MS-DOS, pengguna dapat menggunakan karakter ASCII dengan menekan tombol Alt+[nomor nilai ANSI (desimal)]. Sebagai contoh, tekan kombinasi tombol Alt+87 untuk karakter huruf latin "W" kapital.

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 ampersan (&)
'	0027	39	Karakter Apostrof
(	0028	40	Tanda kurung buka
)	0029	41	Tanda kurung tutup
*	002A	42	Karakter asteris (bintang)
+	002B	43	Tanda tambah (plus)

,	002C	44	Karakter koma
-	002D	45	Karakter hyphen (strip)
.	002E	46	Tanda titik
/	002F	47	Garis miring ( <i>slash</i> )
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 ( <i>backslash</i> )
]	005D	93	Kurung sikur kanan
^	005E	94	Tanda pangkat
_	005F	95	Garis bawah ( <i>underscore</i> )
`	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
ı	00A1	161	Tanda seru terbalik
¢	00A2	162	Tanda sen (Cent)
£	00A3	163	Tanda Poundsterling

¤	00A4	164	Tanda mata uang ( <i>Currency</i> )
¥	00A5	165	Tanda Yen
¡	00A6	166	Garis tegak putus-putus ( <i>broken bar</i> )
§	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 ( <i>hyphen</i> )
®	00AE	174	Tanda merk terdaftar
ˉ	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

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'.

Jawab :

ACALL (Absolute Call)

ACALL berfungsi untuk memanggil sub rutin program ADD (Add Immediate Data)

ADD berfungsi untuk menambah 8 bit data langsung ke dalam isi akumulator dan menyimpan hasilnya pada akumulator.3. ADDC(Add Carry Plus Immediate Data to Accumulator)

ADDC berfungsi untuk menambahkan isi carry flag (0 atau 1) ke dalam isi akumulator. Data langsung 8 bit ditambahkan keakumulator.

AJMP (Absolute Jump)

AJMP adalah perintah jump mutlak. Jump dalam 2 KB dimulai dari alamat yang mengikuti perintah AJMP.

AJMP berfungsi untukmentransfer kendali program ke lokasi dimana alamat dikalkulasi dengan cara yang sama dengan perintah ACALL. Konter program ditambahkan dua kali dimana perintah AJMP adalah perintah 2-byte. Konter program di-load dengan a10 – a0 11 bits, untuk membentuk alamat tujuan 16-bit.

ANL (logical AND memori ke akumulator)

ANL berfungsi untuk mengAND-kan isi alamat data dengan isi akumulator.CJNE (Compare Indirect Address to Immediate Data)

CJNE berfungsi untuk membandingkan data langsung dengan lokasi memori yang dialamati oleh register R atau Akumulator A.apabila tidak sama maka instruksi akan menuju ke alamat kode.

Format : CJNE R,#data,Alamat kode.CLR (Clear Accumulator)

CLR berfungsi untuk mereset data akumulator menjadi 00H.

Format : CLR A

CPL (Complement Accumulator)

CPL berfungsi untuk mengkomplemen isi akumulator.DA (Decimal Adjust Accumulator)

DA berfungsi untuk mengatur isi akumulator ke padanan BCD, setelah penambahan dua angka BCD.DEC (Decrement Indirect Address)

DEC berfungsi untuk mengurangi isi lokasi memori yang ditujukan oleh register R dengan 1, dan hasilnya disimpan pada lokasi tersebut.

DIV (Divide Accumulator by B)

DIV berfungsi untuk membagi isi akumulator dengan isi register B. Akumulator berisi hasil bagi, register B berisi sisa pembagian.DJNZ (Decrement Register And Jump If Not Zero)

DJNZ berfungsi untuk mengurangi nilai register dengan 1 dan jika hasilnya sudah 0 maka instruksi selanjutnya akan dieksekusi. Jika belum 0 akan menuju ke alamat kode.

INC (Increment Indirect Address)

INC berfungsi untuk menambahkan isi memori dengan 1 dan menyimpannya pada alamat tersebut.JB (Jump if Bit is Set)

JB berfungsi untuk membaca data per satu bit, jika data tersebut adalah 1 maka akan menuju ke alamat kode dan jika 0 tidak akan menuju ke alamat kode.

JBC (Jump if Bit Set and Clear Bit)

Bit JBC, berfungsi sebagai perintah rel menguji yang terspesifikasikan secara bit. Jika bit di-set, maka Jump dilakukan ke alamat relatif dan yang terspesifikasi secara bit di dalam perintah dibersihkan. Segmen program berikut menguji bit yang kurang signifikan (LSB: Least Significant Byte), dan jika ditemukan bahwa ia telah di-set, program melompat ke READ lokasi. JBC juga berfungsi membersihkan LSB dari akumulator.

JC (Jump if Carry is Set)

Instruksi JC berfungsi untuk menguji isi carry flag. Jika berisi 1, eksekusi menuju ke alamat kode, jika berisi 0, instruksi selanjutnya yang akan dieksekusi.

JMP (Jump to sum of Accumulator and Data Pointer)

Instruksi JMP berfungsi untuk memerintahkan loncat ke suatu alamat kode tertentu.Format : JMP alamat kode.

JNB (Jump if Bit is Not Set)

Instruksi JNB berfungsi untuk membaca data per satu bit, jika data tersebut adalah 0 maka akan menuju ke alamat kode dan jika 1 tidak akan menuju ke alamat kode.

Format : JNB alamat bit,alamat kode.JNC (Jump if Carry Not Set)

JNC berfungsi untuk menguji bit Carry, dan jika tidak di-set, maka sebuah lompatan akan dilakukan ke alamat relatif yang telah ditentukan.

JNZ (Jump if Accumulator Not Zero)

JNZ adalah mnemonik untuk instruksi jump if not zero (lompat jika tidak nol). Dalam hal ini suatu lompatan akan terjadi bila mana bendera nol dalam keadaan “clear”, dan tidak akan terjadi lompatan bila mana bendera nol tersebut dalam keadaan set. Andaikan bahwa JNZ 7800H disimpan pada lokasi 2100H. Jika Z=0, instruksi berikutnya akan berasal dari lokasi 7800H: dan bila mana Z=1, program akan turun ke instruksi urutan berikutnya pada lokasi 2101H.

JZ ( Jump if Accumulator is Zero )

JZ berfungsi untuk menguji konten-konten akumulator. Jika bukan nol, maka lompatan dilakukan ke alamat relatif yang ditentukan dalam perintah.

LCALL ( Long Call )

LCALL berfungsi untuk memungkinkan panggilan ke subrutin yang berlokasi dimanapun dalam memori program 64K. Operasi LCALL berjalan seperti berikut :

- Menambahkan ke dalam konter program sebanyak 3, karena perintahnya adalah perintah 3-byte.

- Menambahkan penunjuk stack sebanyak 1.
- Menyimpan byte yang lebih rendah dari konter program ke dalam stack.
- Menambahkan penunjuk stack.
- Menyimpan byte yang lebih tinggi dari program ke dalam stack.
- Me-load konter program dengan alamat tujuan 16-bit.