

Nama : Maulana Alhif Ikhsan

NIM : L200180120

Tugas Praktikum Sistem Operasi Modul 1

1. ASCII

ASCII (American Standard Code for Information Interchange) merupakan Kode Standar Amerika untuk pertukaran informasi atau sebuah standar internasional dalam pengkodean huruf dan symbol seperti Unicode dan Hex tetapi ASCII lebih bersifat universal. Contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakterkarakter angka maupun huruf didalam komputer, sebagai contoh dapat kita lihat pada karakter 1, 2, 3, A, B, C, dan sebagainya.

TABEL ASCII

<i>Decimal</i>	<i>Hexadecimal</i>	<i>Binary</i>	<i>Character</i>	<i>Description</i>
0	00	0000000 0	NU L	Null
1	01	0000000 1	SOH	Start of Header
2	02	0000001 0	STX	Start of Text
3	03	0000001 1	ETX	End of Text
4	04	0000010 0	EOT	End of Transmision
5	05	0000010 1	ENQ	Enquiry
6	06	0000011 0	ACK	Acknowledge
7	07	0000011 1	BEL	Bell
8	08	0000100 0	BS	Backspace
9	09	0000100 1	HT	Horizontal Tab
10	0A	0000101 0	LF	Line Feed
11	0B	0000101 1	VT	Vertical Tab
12	0C	0000110 0	FF	Form Feed

13	0D	0000110 1	CR	Carriage Return
14	0E	0000111 0	SO	Shift Out
15	0F	0000111 1	SI	Shift In
16	10	0001000 0	DLE	Data Link Escape

17	11	00010001	DC1	Device Control 1
18	12	00010010	DC2	Device Control 2
19	13	00010011	DC3	Device Control 3
20	14	00010100	DC4	Device Control 4
21	15	00010101	NAK	Negative Acknowledge
22	16	00010110	SYN	Synchronize
23	17	00010111	ETB	End of Transmission Block
24	18	00011000	CAN	Cancel
25	19	00011001	EM	End of Medium
26	1A	00011010	SUB	Substitute
27	1B	00011011	ESC	Escape
28	1C	00011100	FS	File Separator
29	1D	00011101	GS	Group Separator
30	1E	00011110	RS	Record Separator
31	1F	00011111	US	Unit Separator
32	20	00100000	Space	Space
33	21	00100001	!	Exclamation Mark
34	22	00100010	"	Double Quote
35	23	00100011	#	Number
36	24	00100100	\$	Dollar Sign
37	25	00100101	%	Percent
38	26	00100110	&	Ampersand
39	27	00100111	'	Single Quote
40	28	00101000	(Left Parenthesis
41	29	00101001)	Right Parenthesis
42	2A	00101010	*	Asterisk
43	2B	00101011	+	Plus
44	2C	00101100	,	Comma

45	2D	00101101	-	Minus
46	2E	00101110	.	Period
47	2F	00101111	/	Slash
48	30	00110000	0	Zero
49	31	00110001	1	One
50	32	00110010	2	Two
51	33	00110011	3	Three
52	34	00110100	4	Four
53	35	00110101	5	Five
54	36	00110110	6	Six
55	37	00110111	7	Seven
56	38	00111000	8	Eight
57	39	00111001	9	Nine
58	3A	00111010	:	Colon
59	3B	00111011	;	Semicolon
60	3C	00111100	<	Less Than
61	3D	00111101	=	Equality Sign
62	3E	00111110	>	Greater Than
63	3F	00111111	?	Question mark
64	40	01000000	@	At Sign
65	41	01000001	A	Capital A
66	42	01000010	B	Capital B
67	43	01000011	C	Capital C
68	44	01000100	D	Capital D
69	45	01000101	E	Capital E
70	46	01000110	F	Capital F
71	47	01000111	G	Capital G
72	48	01001000	H	Capital H
73	49	01001001	I	Capital I

74	4A	0100101 0	J	Capital J
75	4B	0100101 1	K	Capital K
76	4C	0100110 0	L	Capital L
77	4 D	0100110 1	M	Capital M
78	4E	0100111 0	N	Capital N
79	4F	0100111 1	O	Capital O
80	50	0101000 0	P	Capital P
81	51	0101000 1	Q	Capital Q
82	52	0101001 0	R	Capital R
83	53	0101001 1	S	Capital S
84	54	0101010 0	T	Capital T
85	55	0101010 1	U	Capital U
86	56	0101011 0	V	Capital V
87	57	0101011 1	W	Capital W
88	58	0101100 0	X	Capital X
89	59	0101100 1	Y	Capital Y
90	5A	0101101 0	Z	Capital Z
91	5B	0101101 1	[Left square bracket
92	5C	0101110 0	\	Backslash
93	5 D	0101110 1]	Right square bracket
94	5E	0101111 0	^	Caret/circumflex
95	5F	0101111 1	_	Underscore
96	60	0110000 0	`	Grave / accent
97	61	0110000 1	a	Small a
98	62	0110001	b	Small b

		0		
99	63	0110001 1	c	Small c
100	64	0110010 0	d	Small d
101	65	0110010 1	e	Small e
102	66	0110011 0	f	Small f
103	67	0110011 1	g	Small g

104	68	0110100 0	h	Small h
105	69	0110100 1	i	Small i
106	6A	0110101 0	j	Small j
107	6B	0110101 1	k	Small k
108	6C	0110110 0	l	Small l
109	6 D	0110110 1	m	Small m
110	6E	0110111 0	n	Small n
111	6F	0110111 1	o	Small o
112	70	0111000 0	p	Small p
113	71	0111000 1	q	Small q
114	72	0111001 0	r	Small r
115	73	0111001 1	s	Small s
116	74	0111010 0	t	Small t
117	75	0111010 1	u	Small u
118	76	0111011 0	v	Small v
119	77	0111011 1	w	Small w
120	78	0111100 0	x	Small x
121	79	0111100 1	y	Small y
122	7A	0111101 0	z	Small z
123	7B	0111101 1	{	Left curly bracket
124	7C	0111110 0		Vertical bar
125	7 D	0111110 1	}	Right curly bracket
126	7E	0111111 0	~	Tilde
127	7F	0111111 1	DEL	Delete

2. Daftar Perintah Bahasa Assembly untuk intel x86

Terdapat 2 jenis yang ditulis dalam program Bahasa assembly ,antara lain:

Assembly Directive

Merupakan kode yang menjadi arahan bagi compiler untuk menata program.

Instruksi

Kode yang harus dieksekusi oleh CPU mikrokontroler dengan melakukan operasi tertentu sesuai dengan daftar yang sudah tertanam dalam CPU.

DAFTAR ASSEMBLY DIRECTIVE

<i>Assembly Directive</i>	<i>Keterangan</i>
EQU	Pendefinisian Kosntanta
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 kode segment
XSEG	Penanda penempatan di eksternal data segment
DSEG	Penanda penempatan di internal direct data segment
ISEG	Penanda penempatan di internal indirect data segment
BSEG	Penanda penempatan dibit data segment
CODE	Penanda mulai pendefinisian program
XDATA	Pendefinisian eksternal data
DATA	Pendefinisian internal direct data
IDATA	Pendefinisian internal indirect data
BIT	Pendefinisian data bit
INCLUDE	Mengikutertakan file program lain

DAFTAR INTRUKSI

<i>Intruksi</i>	<i>Keterangan Singkatan</i>
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
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

Penjelasan :

a *MOV*

Perintah MOV adalah perintah untuk mengisi, memindahkan, memperbarui 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 computer 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 (interrupt 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 computer 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 sedang aktif.