

PRAKTIKUM SISTEM OPERASI
PENGENALAN SISTEM PENGEMBANGAN OS DENGAN PC SIMULATOR
“BOCHS”



Oleh :
Dian Putri Mutiara Hapsari
L200210238

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA

2022

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 adalah singkatan dari *American Standart Code for Information Interchange* merupakan pengkodean karakter di dalam suatu komunikasi elektronik. Kode ASCII dapat digambarkan sebagai teks pada komputer, peralatan komunikasi dan perangkat lainnya. Kode ASCII merepresentasikan sebagai suatu nilai bilangan integer yang digunakan sebagai nilai pada suatu karakter seperti karakter huruf, angka, simbol, dan karakter lainnya yang tidak dapat dicetak seperti karakter khusus dan juga pengendali karakter seperti carriage return, tabulasi, dan DEL.

Decimal	Binary	Hexadecimal	Character
0	0000	00	NUL
1	0001	01	SOH
2	0010	02	STX
3	0011	03	ETX
4	0100	04	EOT
5	0101	05	ENQ
6	0110	06	ACK
7	0111	07	BEL
8	1000	08	BS
9	1001	09	HT
10	1010	0A	LF
11	1011	0B	VT
12	1100	0C	FF
13	1101	0D	CR
14	1110	0E	SO
15	1111	0F	SI
16	10000	10	DLE
17	10001	11	DC1
18	10010	12	DC2
19	10011	13	DC3

20	10100	14	DC4
21	10101	15	NAK
22	10110	16	SYN
23	10111	17	ETB
24	11000	18	CAN
25	11001	19	EM
26	11010	1A	SUB
27	11011	1B	ESC
28	11100	1C	FS
29	11101	1D	GS
30	11110	1E	RS
31	11111	1F	US
32	100000	20	Space
33	100001	21	!
34	100010	22	“
35	100011	23	#
36	100100	24	\$
37	100101	25	%
38	100110	26	&
39	100111	27	'
40	101000	28	(
41	101001	29)
42	101010	2A	*
43	101011	2B	+
44	101100	2C	,
45	101101	2D	-
46	101110	2E	.
47	101111	2F	/
48	110000	30	0
49	110001	31	1
50	110010	32	2
51	110011	33	3

52	110100	34	4
53	110101	35	5
54	110110	36	6
55	110111	37	7
56	111000	38	8
57	111001	39	9
58	111010	3A	:
59	111011	3B	;
60	111100	3C	<
61	111101	3D	=
62	111110	3E	>
63	111111	3F	?
64	1000000	40	@
65	1000001	41	A
66	1000010	42	B
67	1000011	43	C
68	1000100	44	D
69	1000101	45	E
70	1000110	46	F
71	1000111	47	G
72	1001000	48	H
73	1001001	49	I
74	1001010	4A	J
75	1001011	4B	K
76	1001100	4C	L
77	1001101	4D	M
78	1001110	4E	N
79	1001111	4F	O
80	1010000	50	P
81	1010001	51	Q
82	1010010	52	R
83	1010011	53	S

84	1010100	54	T
85	1010101	55	U
86	1010110	56	V
87	1010111	57	W
88	1011000	58	X
89	1011001	59	Y
90	1011010	5A	Z
91	1011011	5B	[
92	1011100	5C	\
93	1011101	5D]
94	1011110	5E	^
95	1011111	5F	_
96	1100000	60	`
97	1100001	61	a
98	1100010	62	b
99	1100011	63	c
100	1100100	64	d
101	1100101	65	e
102	1100110	66	f
103	1100111	67	g
104	1101000	68	h
105	1101001	69	i
106	1101010	6A	j
107	1101011	6B	k
108	1101100	6C	l
109	1101101	6D	m
110	1101110	6E	n
111	1101111	6F	o
112	1110000	70	p
113	1110001	71	q
114	1110010	72	r
115	1110011	73	s

116	1110100	74	t
117	1110101	75	u
118	1110110	76	v
119	1110111	77	w
120	1111000	78	x
121	1111001	79	y
122	1111010	7A	z
123	1111011	7B	{
124	1111100	7C	
125	1111101	7D	}
126	1111110	7E	~
127	1111111	7F	delete

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 :

No	Instruksi	Fungsi
1	ACALL	Memanggil sub rutin program.
2	ADD	Menambah 8 bit data langsung ke dalam isi akumulator dan menyimpan hasilnya pada akumulator.
3	ADDC	Menambahkan isi carry flag (0 atau 1) ke dalam isi akumulator.
4	AJMP	Mentransfer kendali program ke lokasi dimana alamat dikalkulasi dengan cara yang sama dengan perintah ACALL.
5	ANL	MengAND-kan isi alamat data dengan isi akumulator.
6	CJNE	Membandingkan data langsung dengan lokasi memori yang dialamati oleh register R atau Akumulator A.
7	CLR	Mereset data akumulator menjadi 00H.
8	CPL	Mengkomplemen isi akumulator.
9	DA	Mengatur isi akumulator ke padanan BCD, setelah penambahan dua angka BCD.

10	DEC	Mengurangi isi lokasi memori yang ditujukan oleh register R dengan 1, dan hasilnya disimpan pada lokasi tersebut.
11	DIV	Membagi isi akumulator dengan isi register B. Akumulator berisi hasil bagi, register B berisi sisa pembagian.
12	DJNZ	Mengurangi nilai register dengan 1 dan jika hasilnya sudah 0 maka instruksi selanjutnya akan dieksekusi.
13	INC	Menambahkan isi memori dengan 1 dan menyimpannya pada alamat tersebut.
14	JB	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.
15	JBC	Perintah rel menguji yang terspesifikasikan secara bit.
16	JC	Menguji isi carry flag.
17	JMP	Memerintahkan loncat ke suatu alamat kode tertentu.
18	JNB	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.
19	JNC	Menguji bit Carry, dan jika tidak di-set, maka sebuah lompatan akan dilakukan ke alamat relatif yang telah ditentukan.
20	JNZ	
21	JZ	Menguji konten-konten akumulator.
22	LCALL	Memungkinkan panggilan ke subrutin yang berlokasi dimanapun dalam memori program 64K.
23	LJMP	Memungkinkan lompatan tak bersyarat kemana saja dalam lingkup ruang memori program 64K.
24	MOV	Memindahkan isi akumulator/register atau data dari nilai luar atau alamat lain.
25	MOVC	Mengisi accumulator dengan byte kode atau konstanta dari program memory.

26	MOVX	Memindahkan isi akumulator ke memori data eksternal yang alamatnya ditunjukkan oleh isi data pointer.
27	MUL	Mengalikan unsigned 8 bit integer pada accumulator dan register B.
28	NOP	Eksekusi program akan dilanjutkan ke instruksi berikutnya.
29	ORL	Instruksi Gerbang logika OR yang akan menjumlahkan Accumulator terhadap nilai yang ditentukan.
30	POP	Menempatkan byte yang ditunjukkan oleh stack pointer ke suatu alamat data.
31	PUSH	Menaikkan stack pointer kemudian menyimpan isinya ke suatu alamat data pada lokasi yang ditunjuk oleh stack pointer.
32	RET	Kembali dari suatu subrutin program ke alamat terakhir subrutin tersebut di panggil.
33	RETI	Mengambil nilai byte tinggi dan rendah dari PC dari stack dan mengembalikan kondisi logika interrupt agar dapat menerima interrupt lain dengan prioritas yang sama dengan prioritas interrupt yang baru saja diproses.
34	RL	Memutar setiap bit dalam akumulator satu posisi ke kiri.
35	RLC	Memutar akumulator ke kiri melalui carry flag.
36	RR	Memutar akumulator ke kanan.
37	RRC	Memutar akumulator ke kanan melalui carry flag.
38	SETB	Mereset carry flag.
39	SJMP	Mentransfer kendali ke alamat tujuan dalam 127 bytes yang mengikuti dan 128 yang mengawali perintah SJMP.
40	SUBB	Pengurangan dengan peminjaman.
41	SWAP	Menukar upper nibble dan lower nibble dalam akumulator.
42	XCH	Menukar akumulator dengan variabel byte.
43	XCHD	Menukar digit.
44	XRL	Variabel Byte XRL akan melakukan operasi bitwise logika exclusive OR antara kedua variabel yang dinyatakan.

No	Assembly Directive	Keterangan
1	EQU	Pendefinisian konstanta
2	DB	Pendefinisian data dengan ukuran satuan 1 byte
3	DW	Pendefinisian data dengan ukuran satuan 1 word
4	DBIT	Pendefinisian data dengan ukuran satuan 1 bit
5	DS	Pemesanan tempat penyimpanan data di RAM
6	ORG	Inisialisasi alamat mulai program
7	END	Penanda akhir program
8	CSEG	Penanda penempatan di code segment
9	XSEG	Penanda penempatan di external data segment
10	DSEG	Penanda penempatan di internal direct data segment
11	ISEG	Penanda penempatan di internal indirect data segment
12	BSEG	Penanda penempatan di bit data segment
13	CODE	Penanda mulai pendefinisian program
14	XDATA	Pendefinisian external data
15	DATA	Pendefinisian internal direct data
16	IDATA	Pendefinisian internal indirect data
17	BIT	Pendefinisian data bit
18	#INCLUDE	Mengikutsertakan file program lain