

Nama : Ninda	Nilai
Nim : L200210265	
Nama dosen : Heru Setiya Nugraha,ST, M.Kom	Tanda tangan
Kelas : Praktikum Sistem Operasi	

Modul 1 Pengenalan Sistem Pengembangan OS dengan PC Simulator 'Bochs'

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 (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.

Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
\$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110

/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[91	1011011	{	123	1111011
<	60	111100		92	1011100		124	1111100
=	61	111101]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

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 :

Definisi data

DB : define bytes. Membentuk data byte demi byte.

DW : define words.

DD : define double words.

EQU : equals. Membentuk konstanta.

Ada assembly yang melibatkan bilangan pecahan (floating point), bilangan bulat (integer), DF (define far words), DQ (define quad words), dan DT (define ten bytes).

Perpindahan data

MOV : move. Memindahkan suatu nilai dari register ke memori, memori ke register, atau register ke register.

LEA : load effective address. Mengisi suatu register dengan alamat offset sebuah data. sintaks: LEA {register}, {sumber} contoh: lea DX, teks1
XCHG : exchange. Menukar dua buah register langsung.

Operasi logika

AND : melakukan bitwise and. sintaks: AND {register}, {angka} AND {register 1}, {register 2} hasil disimpan di register 1.

OR : melakukan bitwise or. sintaks: OR {register}, {angka} OR {register 1}, {register 2} hasil disimpan di register 1.

NOT : melakukan bitwise not (*one's complement*) sintaks: NOT {register} hasil disimpan di register itu sendiri.

XOR : melakukan bitwise eksklusif or. sintaks: XOR {register}, {angka} XOR {register 1}, {register 2} hasil disimpan di register 1. Tips: sebuah register yang di-XOR-kan dengan dirinya sendiri akan menjadi berisi nol.

SHL : shift left. Menggeser bit ke kiri. Bit paling kanan diisi nol. sintaks: SHL {register}, {banyaknya}

SHR : shift right. Menggeser bit ke kanan. Bit paling kiri diisi nol. sintaks: SHR {register}, {banyaknya}

ROL : rotate left. Memutar bit ke kiri. Bit paling kiri jadi paling kanan kali ini. sintaks: ROL {register}, {banyaknya}

ROR : rotate right. Memutar bit ke kanan.

Operasi matematika

ADD : add. Menjumlahkan dua buah register.

ADC : add with carry. Menjumlahkan dua register dan carry flag (CF).

INC : increment. Menjumlah isi sebuah register dengan 1.

SUB : subtract. Mengurangkan dua buah register.

SBB : subtract with borrow. Mengurangkan dua register dan carry flag (CF).

DEC : decrement. Mengurang isi sebuah register dengan 1.

MUL : multiply. Mengalikan register dengan AX atau AH.

IMUL : signed multiply.

DIV : divide. Membagi AX atau DX:AX dengan sebuah register.

IDIV : signed divide. Sama dengan DIV, hanya saja IDIV menganggap bit-bit yang ada di register sumber sudah dalam bentuk *two's complement*.

NEG : negate. Membuat isi register menjadi negatif (*two's complement*).

Pengulangan

LOOP : loop. Mengulang sebuah proses.

LOOPE : loop while equal. Melakukan pengulangan selama CX \neq 0 dan ZF = 1. CX tetap dikurangi 1 sebelum diperiksa.

LOOPZ : loop while zero. Identik dengan LOOPE.

LOOPNE : loop while not equal.

Melakukan pengulangan selama CX \neq 0 dan ZF = 0. CX tetap dikurangi 1 sebelum diperiksa.

LOOPNZ : loop while not zero. Identik dengan LOOPNE.

REP : repeat. Mengulang perintah sebanyak CX kali.

REPE : repeat while equal. Mengulang perintah sebanyak CX kali, tetapi pengulangan segera dihentikan bila didapati ZF = 1.

REPZ : repeat while zero. Identik dengan REPE.

REPNE : repeat while not equal. Mengulang perintah sebanyak CX kali, tetapi pengulangan segera dihentikan bila didapati ZF = 0.

REPNZ : repeat while not zero. Identik dengan REPNE.

Perbandingan

CMP : compare. Membandingkan dua buah operand.

Operasi stack

PUSH : push. Menambahkan sesuatu ke stack.

POP : pop. Mengambil sesuatu dari stack.

PUSHF : push flags. Mem-push **semua** isi register flag ke dalam stack.

POPF : pop flags. Lawan dari pushf. Sintaks: POPF ;(saja).

POPA : pop all general-purpose registers.

PUSHA : push all general-purpose registers.

Operasi pada register flag

CLC : clear carry flag. Menjadikan CF = 0. Sintaks: CLC ;(saja).

STC : set carry flag. Menjadikan CF = 1. Sintaks: STC ;(saja).

CMC : complement carry flag. Melakukan operasi NOT pada CF. Yang tadinya 0 menjadi 1, dan sebaliknya.

CLD : clear direction flag. Menjadikan DF = 0. Sintaks: CLD ;(saja).

STD : set direction flag. Menjadikan DF = 1.

CLI : clear interrupt flag.

STI : set interrupt flag. Menjadikan IF = 1.

Perintah lainnya

ORG : origin. Mengatur awal dari program (bagian static data).

IRET : interrupt-handler return.

CALL : call procedure. Memanggil sebuah prosedur.

RET : return. Tanda selesai prosedur.

HLT : halt. Membuat prosesor menjadi tidak aktif.