



Microprocessors & Interfacing

MASM

Dr. Gargi Prabhu Department of CS & IS



Exercise

Write a program to take two numbers as input from the user and print its addition and subtraction results.

Questions

- .exit and int 21h with AH= 4CH is same?
- Why cant we move data to DS directly?



How to debug in MASM?

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
 For supported shell commands type: HELP
 To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
 To activate the keymapper ctrl-F1.
 For more information read the README file in the DOSBox directory.
 HAVE FUN!
 The DOSBox Team http://www.dosbox.com
Z:\>mount d_d:/
Dri∨e D is mounted as local directory d:∧
Z:\>d:/
D:\>cd MASM/MASM611/BIN
D:\MASM\MASM611\BIN>debug SIMPLE.EXE
-t
AX-076B BX-0000 CX-0019 DX-0000 SP-0100 BP-0000 SI-0000 DI-0000
DS=075A ES=075A SS=076C CS=076A IP=0003 NV UP EI PL NZ NA PO NC
076A:0003 8ED8
                      MOV
                              DS.AX
```



How to debug in MASM?

DOSBox 0.74, Cpu speed: 3	3000 cycles, Frame:	skip 0, Program:	DEBUG	_		\times				
Z:\>d:/										
D:N>cd MASM/MASM611/BIN										
D:\MASM\MASM611\BIN>debug SIMPLE.EXE -t										
AX=076B BX=0000 CX=0	019 DX=0000	SP=0100	BP=0000 SI=0000 I	I=0000)					
DS=075A ES=075A SS=0	76C CS=076A	IP=0003	NV UP EI PL NZ NA	PO NC						
076A:0003 BED8	MOV DS	,AX								
–u 0003										
076A:0003 BED8		,AX								
076A:0005 A00600	MOV AL	, [0006]								
076A:0008 8A1E0700	MOV BL	, [0007]								
076A:000C 02C3	ADD AL	, BL								
076A:000E A20800	MOV [00	9081,AL								
076A:0011 B44C	MOV AH	,4C								
076A:0013 CD21	INT 21									
076A:0015 00361B00	ADD [00	91B],DH								
076A:0019 0000	ADD [BX	K+SIl,AL								
076A:001B 0000		K+SII,AL								
076A:001D 0000		K+SII,AL								
076A:001F 0000		K+SII,AL								
076A:0021 0000		K+SI1,AL								
_										



ASCII Conversion

Input from and output to the console is performed in ASCII format

Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value
00	NUL	10	DLE	20	SP	30	0	40	@	50	Р	60	•	70	р
01	SOH	11	DC1	21	!	31	1	41	Α	51	Q	61	а	71	q
02	STX	12	DC2	22	"	32	2	42	В	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	С	53	S	63	С	73	S
04	EOT	14	DC4	24	\$	34	4	44	D	54	Т	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	Е	55	U	65	е	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	V
07	BEL	17	ETB	27	•	37	7	47	G	57	W	67	g	77	W
08	BS	18	CAN	28	(38	8	48	Н	58	Χ	68	h	78	X
09	HT	19	EM	29)	39	9	49	I	59	Υ	69	i	79	у
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	Z
0B	VT	1B	ESC	2B	+	3B	;	4 B	K	5B	[6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	I	7C	1
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D]	6D	m	7D	}
0E	SO	1E	RS	2E		3E	>	4E	N	5E	۸	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	О	5F	_	6F	0	7F	DEL

innovate achieve lead

ASCII Conversion

How to perform the conversion?

E.g. Input: 8h



Function 1- Character input with echo

Action: Reads a character from the standard input

device and echoes it to the standard

output device.

If no character is ready it waits until one is

available.

I/O can be re-directed, but prevents

detection of OEF.

On entry: AH = 01h

Returns: AL = 8 bit data input



Function 2 - Character output

Action: Outputs a character to the standard

output device. I/O can be re-directed, but

prevents detection of 'disc full'.

On entry: AH = 02h

DL = 8 bit data (usually ASCII character)

Returns: Nothing



Function 08- Character input with no echo

Action: Reads a character from the standard input

device without copying it to the display.

If no character is ready it waits until one is

available.

On entry: AH = 08h

Returns: AL = 8 bit data input



Function 09- Output character string

Action: Writes a string to the display.

On entry: AH = 09h

DS:DX = segment:offset of string

Returns: Nothing

Notes: The string must be terminated by the \$

character (24h), which is not transmitted.

Any ASCII codes can be embedded within

the string.

innovate achieve lead

Interrupts – Int 21h

Action:

Function 0Ah - Buffered input

On entry:			
Returns:			
Notes:			

Reads a string from the current input device up to and including an ASCII carriage return (0Dh), placing the received data in a user-defined buffer Input can be re directed, but this prevents detection of EOF

AH = OAh

DS:DX = segment:offset of string buffer

Nothing

The first byte of the buffer specifies the maximum number of characters it can hold (1 to 255). This value must be supplied by the user. The second byte of the buffer is set by DOS to the number of characters actually read, excluding the terminating RETURN. If the buffer fills to one less than its maximum size the bell is sounded and subsequent input is ignored.

If a CTRL-C is detected an INT 23h is executed. Normal DOS keyboard editing is supported during input



How to enter two digit number?

mov ah,0ah lea dx,num1 int 21h



Converting to Hexadecimal

```
convert_loop:
  mov al, [si]
                    ; Load ASCII character into AL
  cmp al, '0'
  jb input error
                      ; Jump if input is not a valid hexadecimal digit
  cmp al, '9'
  jbe numeric_digit ; Jump if input is between '0' and '9'
  cmp al, 'A'
  jb input_error
                      ; Jump if input is not a valid hexadecimal digit
  cmp al, 'F'
  ja input_error ; Jump if input is not a valid hexadecimal digit
  sub al, 'A'; Convert ASCII character 'A'-'F' to numeric value 10-15
  add al, 0ah
```



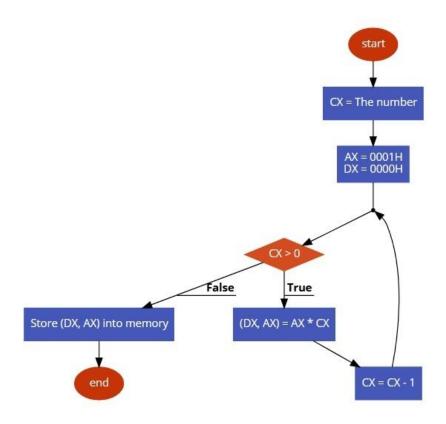
Numerical value in Hex

```
numeric_digit:
and al, OFH ; Mask high nibble (if any)
shl ax, cl ; Shift AL left by 4 (multiply by 16)
add hex_number, ax ; Add to the result
dec cl ; Decrement loop counter
inc si ; Move to next character
loop convert_loop ; Repeat for the second nibble
```

Exercise



Factorial of a number





Exercise

Can you use stack to solve factorial problem?

innovate achieve lead

References

 http://bbc.nvg.org/doc/Master%20512%20Technical%20Guid e/m512techb_int21.htm



Thank You