

St. Francis Institute of Technology

Class: SE-ITA/ITB Semester: IV; A.Y. 2023-2024

Subject: Microprocessor Lab

Experiment – 7: Compute factorial of a positive number

1. Aim:

Write an ALP to compute factorial of a positive number.

2. Requirements

DOSBox (an x86 emulator with DOS), Turbo Assembler, Turbo Debugger

3. Pre-Experiment Exercise

Algorithm:

- a. Initialize the data segment.
- b. Display message “enter the number to calculate factorial:\$” and use INT 21h commands to take a single digit input from the user.
- c. Call a procedure to calculate the factorial of the number.
- d. Store the result in data memory.

4. Laboratory Exercise:

Procedure:

- a. Open DOSbox and go to TASM.
- b. Open a new document using the command - edit <filename>.asm
- c. Write the Program and save the changes to the same file.
- d. Assemble the program using the command - tasm <filename.asm>
- e. If any errors are displayed, then change the code in <filename>
- f. If no errors are displayed, execute the command - tlink <filename>.obj to create the executable file.
- g. Next execute the command - td <filename>
- h. Try to RUN the program step by step and view the changes in the registers, flags, memory, etc.

5. Post Experiment Exercise:

a. Results/Calculations/Observations:

Attach appropriate screenshots of the input from the user and output stored in memory along with the ALP.

b. Questions:

- i. Write an ALP in TASM to calculate LCM of two 16-bit numbers. Attach appropriate screenshots.
- ii. Write and ALP in TASM to calculate GCD of two 16-bit numbers. Attach appropriate screenshots.

c. Conclusion:

Write the conclusion/comments based on the experiment performed and the output obtained.

d. References:

Mention two book references and two web references.

EXPERIMENT 7 MPL

1. Write an ALP to compute the factorial of a positive number.

data segment

msg db "enter the number:\$"

num db ?

res1 dw ?

res2 dw ?

data ends

code segment

assume cs:code,ds:data

start:

mov ax,data

mov ds,ax

lea dx,msg

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,48

call facte

mov ah,4ch

int 21h

facte proc

mov num,al

mov cl,al

mov ch,00h

mov ax,0001h

mov dx,0000h

l1:mul cx

loop l1

mov res1,ax

mov res2,dx

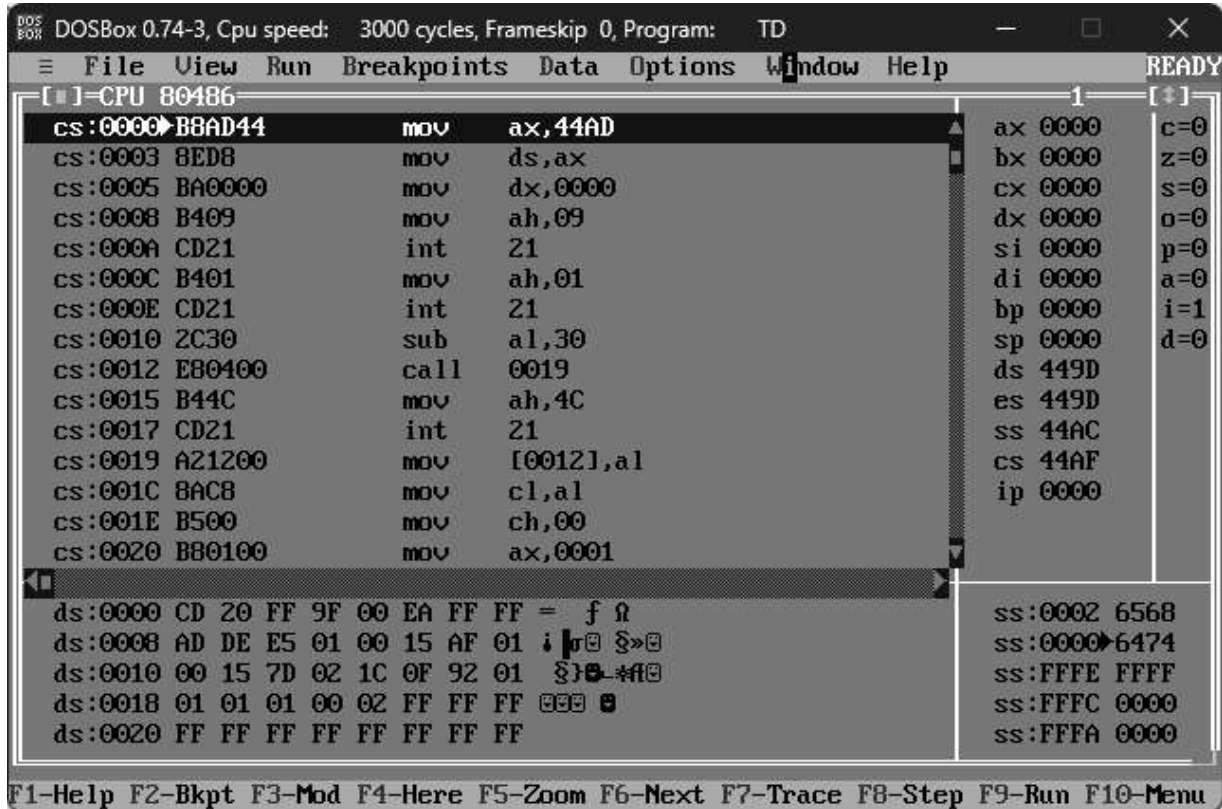
ret

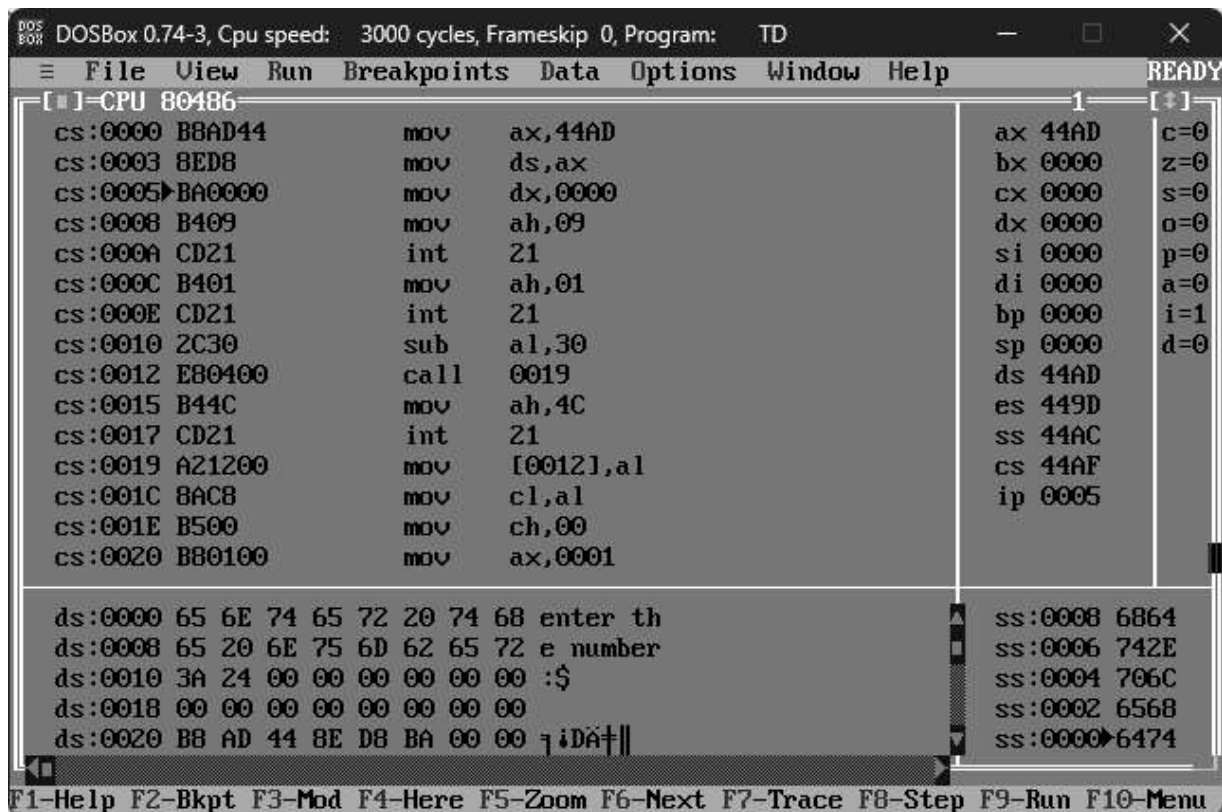
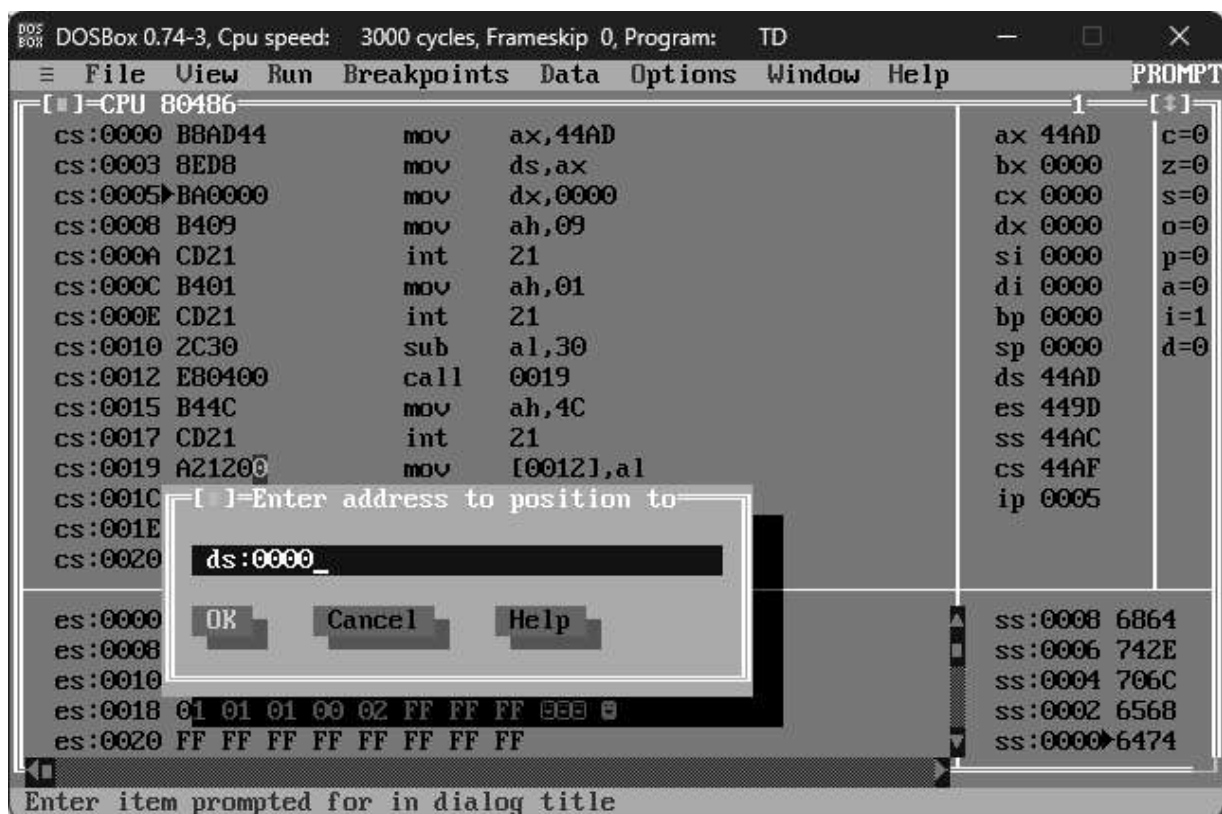
facte endp

code ends

end start

Output:





```
DOS
BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
Assembling file: EXP.asm
**Fatal** Command line: Can't locate file: EXP.asm
Error messages: 1
Warning messages: None
Passes: None
Remaining memory: 492k

C:\>tasm EXP7.asm
Turbo Assembler Version 2.51 Copyright (c) 1988, 1991 Borland International

Assembling file: EXP7.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 491k

C:\>tlink EXP7.obj
Turbo Link Version 4.0 Copyright (c) 1991 Borland International
Warning: No stack

C:\>td EXP7
Turbo Debugger Version 2.51 Copyright (c) 1988,91 Borland International
enter the number:_
```

DOS
BOX

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

≡

File

View

Run

Breakpoints

Data

Options

Window

Help

STATUS

[]-CPU 80486

44AF:0015 B44C

mov

ah,4C

ax 0192

c=1

44AF:0017 CD21

int

Z1

bx 02F4

z=0

44AF:0019 A21200

mov

[0012],al

cx 00D8

s=1

44AF:001C 8AC8

mov

cl,al

dx 0990

o=0

44AF:001E B500

mov

ch,00

si 0019

p=0

44AF:0020 B80100

mov

ax,0001

di 0FA6

a=0

44AF:0023 BA0000

mov

dx,0000

bp 0100

i=1

44AF:0026 F7E1

mul

cx

sp 0106

d=1

44AF:0028 E2FC

loop

0026

ds 1D09

44AF:002A A31300

mov

[0013],ax

es 02F4

44AF:002D 89161500

mov

[0015],dx

ss 0192

44AF:0031 C3

ret

cs 0000

44AF:0032 0000

add

[bx+si],al

ip 0000

44AF:0034 0000

add

[bx+si],al

44AF:0036 0

44AD:0000 6

Terminated, exit code 120

44AC:0008 6864

44AD:0008 6

44AC:0006 742E

44AD:0010 3

OK

44AC:0004 706C

44AD:0018 0

Help

44AC:0002 6568

44AD:0020 B

44AC:0000 6474

Post-Experiment:

1. Write an ALP in TASM to calculate LCM of two 16-bit numbers.

```
data segment
    n1 DW 0000Ah
    n2 DW 00004h
    gcd DW ?
    lcm DW ?
data ends

code segment
    assume cs:code,ds:data
    start:
        mov ax,data
        mov ds,ax
        mov ax,n1
        mov bx,n2
    compare: cmp ax,bx
        je ans
        jb swap
    division: mov dx,0000h
        div bx
        cmp dx,0000h
        je ans
                mov ax,dx
                jmp compare
    swap: xchg ax,bx
        jmp division
    ans: mov gcd,bx
        mov ah,4ch
        int 21h

code ends
end start
```

DOS BOX 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

Address	Instruction	Comment	Register	Value
cs:0000	B8AD44	mov	ax	44AD
cs:0003	8ED8	mov	ds	ax
cs:0005	A10000	mov	ax	[0000]
cs:0008	8B1E0200	mov	bx	[0002]
cs:000C	3BC3	cmp	ax	bx
cs:000E	7413	je		0023
cs:0010	720E	jb		0020
cs:0012	BA0000	mov	dx	0000
cs:0015	F7F3	div	bx	
cs:0017	83FA00	cmp	dx	0000
cs:001A	7407	je		0023
cs:001C	8BC2	mov	ax	dx
cs:001E	EBEC	jmp		000C
cs:0020	93	xchg	bx	ax
cs:0021	EBEF	jmp		0012

ax 0000 c=0
bx 0000 z=0
cx 0000 s=0
dx 0000 o=0
si 0000 p=0
di 0000 a=0
bp 0000 i=1
sp 0000 d=0
ds 449D
es 449D
ss 44AC
cs 44AE
ip 0000

ds:0000 CD 20 FF 9F 00 EA FF FF = f n
ds:0008 AD DE E5 01 00 15 AF 01 i r S>@
ds:0010 00 15 7D 02 1C 0F 92 01 S)B-+f@
ds:0018 01 01 01 00 02 FF FF FF @@@ @
ds:0020 FF FF FF FF FF FF FF FF

ss:0002 6568
ss:0000 6474
ss:FFFE FFFF
ss:FFFC 0000
ss:FFFA 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOS BOX 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

Address	Instruction	Comment	Register	Value
cs:0000	B8AD44	mov	ax	44AD
cs:0003	8ED8	mov	ds	ax
cs:0005	A10000	mov	ax	[0000]
cs:0008	8B1E0200	mov	bx	[0002]
cs:000C	3BC3	cmp	ax	bx
cs:000E	7413	je		0023
cs:0010	720E	jb		0020
cs:0012	BA0000	mov	dx	0000
cs:0015	F7F3	div	bx	
cs:0017	83FA00	cmp	dx	0000
cs:001A	7407	je		0023
cs:001C	8BC2	mov	ax	dx
cs:001E	EBEC	jmp		000C
cs:0020	93	xchg	bx	ax
cs:0021	EBEF	jmp		0012

ax 44AD c=0
bx 0000 z=0
cx 0000 s=0
dx 0000 o=0
si 0000 p=0
di 0000 a=0
bp 0000 i=1
sp 0000 d=0
ds 44AD
es 449D
ss 44AC
cs 44AE
ip 0005

ds:0000 0A 00 04 00 00 00 00 00 0 +
ds:0008 00 00 00 00 00 00 00 00
ds:0010 BB AD 44 8E D8 A1 00 00 i DÄ+i
ds:0018 8B 1E 02 00 3B C3 74 13 i A ;|t!!
ds:0020 72 0E BA 00 00 F7 F3 83 rB|| ≈≤â

ss:0008 6864
ss:0006 742E
ss:0004 706C
ss:0002 6568
ss:0000 6474

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

cs:0012	BA0000	mov	dx,0000	ax 4C02	c=0
cs:0015	F7F3	div	bx	bx 0002	z=1
cs:0017	83FA00	cmp	dx,0000	cx 0000	s=0
cs:001A	7407	je	0023	dx 0000	o=0
cs:001C	8BC2	mov	ax,dx	si 0000	p=1
cs:001E	EBEC	jmp	000C	di 0000	a=0
cs:0020	93	xchg	bx,ax	bp 0000	i=1
cs:0021	EBEF	jmp	0012	sp 0000	d=0
cs:0023	891E0400	mov	[0004],bx	ds 44AD	
cs:0027	B44C	mov	ah,4C	es 449D	
cs:0029	CD21	int	21	ss 44AC	
cs:002B	0000	add	[bx+si],al	cs 44AE	
cs:002D	0000	add	[bx+si],al	ip 0029	
cs:002F	0000	add	[bx+si],al		
cs:0031	0000	add	[bx+si],al		

ds:0000 0A 00 04 00 02 00 00 00 00

ds:0008 00 00 00 00 00 00 00 00

ds:0010 B8 AD 44 8E D8 A1 00 00

ds:0018 8B 1E 02 00 3B C3 74 13

ds:0020 72 0E BA 00 00 F7 F3 83

ss:0008 6864

ss:0006 742E

ss:0004 706C

ss:0002 6568

ss:0000 6474

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

44AE:0012	BA0000	mov	dx,0000	ax 0192	c=1
44AE:0015	F7F3	div	bx	bx 02F4	z=0
44AE:0017	83FA00	cmp	dx,0000	cx 0990	s=1
44AE:001A	7407	je	0023	dx F5DC	o=0
44AE:001C	8BC2	mov	ax,dx	si 0019	p=0
44AE:001E	EBEC	jmp	000C	di 0FA6	a=0
44AE:0020	93	xchg	bx,ax	bp 0100	i=1
44AE:0021	EBEF	jmp	0012	sp 0106	d=1
44AE:0023	891E0400	mov	[0004],bx	ds 1D09	
44AE:0027	B44C	mov	ah,4C	es 02F4	
44AE:0029	CD21	int	21	ss 0192	
44AE:002B	0000	add	[bx+si],al	cs 0000	
44AE:002D	0000	add	[bx+si],al	ip 0000	
44AE:002F	0000	add	[bx+si],al		
44AE:0031	0000	add	[bx+si],al		

ds:030D = A7

44AD:0000 00

44AD:0008 00

44AD:0010 B8

44AD:0018 8B

44AD:0020 72 0E BA 00 00 F7 F3 83

44AC:0008 6864

44AC:0006 742E

44AC:0004 706C

44AC:0002 6568

44AC:0000 6474

Terminated, exit code 2

OK Help

2. Write an ALP in TASM to calculate GCD of two 16-bit numbers

data segment

n1 DW 0000Ah

n2 DW 00004h

gcd DW ?

lcm DW ?

data ends

code segment

assume cs:code,ds:data

start:

mov ax,data

mov ds,ax

mov ax,n1

mov bx,n2

compare: cmp ax,bx

je ans

jb swap

division: mov dx,0000h

div bx

cmp dx,0000h

je ans

mov ax,dx

jmp compare

swap: xchg ax,bx

jmp division

ans: mov gcd,bx

mov ah,4ch

int 21h

code ends

end start

DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

cs:0000	B8AD44	mov	ax,44AD	ax 44AD	c=0
cs:0003	8ED8	mov	ds,ax	bx 0000	z=0
cs:0005	A10000	mov	ax,[0000]	cx 0000	s=0
cs:0008	8B1E0200	mov	bx,[0002]	dx 0000	o=0
cs:000C	3BC3	cmp	ax,bx	si 0000	p=0
cs:000E	7413	je	0023	di 0000	a=0
cs:0010	720E	jb	0020	bp 0000	i=1
cs:0012	BA0000	mov	dx,0000	sp 0000	d=0
cs:0015	F7F3	div	bx	ds 449D	
cs:0017	83FA00	cmp	dx,0000	es 449D	
cs:001A	7407	je	0023	ss 44AC	
cs:001C	8BC2	mov	ax,dx	cs 44AE	
cs:001E	EBEC	jmp	000C	ip 0003	
cs:0020	93	xchg	bx,ax		
cs:0021	EBEF	jmp	0012		

ds:0000 CD 20 FF 9F 00 EA FF FF = f n
ds:0008 AD DE E5 01 00 15 AF 01 i r S>
ds:0010 00 15 7D 02 1C 0F 92 01 S> -f
ds:0018 01 01 01 00 02 FF FF FF 000 0
ds:0020 FF FF FF FF FF FF FF FF

ss:0002 6568
ss:0000 6474
ss:FFFE 3302
ss:FFFC 44AE
ss:FFFA 0003

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

0000	B8AD44	mov	ax,44AD	ax 44AD	c=0
0003	8ED8	mov	ds,ax	bx 0000	z=0
0005	A10000	mov	ax,[0000]	cx 0000	s=0
0008	8B1E0200	mov	bx,[0002]	dx 0000	o=0
000C	3BC3	cmp	ax,bx	si 0000	p=0
000E	7413	je	0023	di 0000	a=0
0010	720E	jb	0020	bp 0000	i=1
0012	BA0000	mov	dx,0000	sp 0000	d=0
0015	F7F3	div	bx	ds 44AD	
0017	83FA00	cmp	dx,0000	es 449D	
001A	7407	je	0023	ss 44AC	
001C	8BC2	mov	ax,dx	cs 44AE	
001E	EBEC	jmp	000C	ip 0005	
0020	93	xchg	bx,ax		
0021	EBEF	jmp	0012		

ds:0000 0A 00 04 00 00 00 00 00 0 +
ds:0008 00 00 00 00 00 00 00 00
ds:0010 BB AD 44 BE D8 A1 00 00 i DÄ+í
ds:0018 8B 1E 02 00 3B C3 74 13 iÄ ;|t!!
ds:0020 72 0E BA 00 00 F7 F3 83 rJ|| ≈≤â

ss:0008 6864
ss:0006 742E
ss:0004 706C
ss:0002 6568
ss:0000 6474

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

cs:0000	B8AD44	mov	ax,44AD	ax 0002	c=1
cs:0003	8ED8	mov	ds,ax	bx 0004	z=0
cs:0005	A10000	mov	ax,[0000]	cx 0000	s=1
cs:0008	8B1E0200	mov	bx,[0002]	dx 0002	o=0
cs:000C	3BC3	cmp	ax,bx	si 0000	p=0
cs:000E	7413	je	0023	di 0000	a=1
cs:0010	720E	jb	0020	bp 0000	i=1
cs:0012	BA0000	mov	dx,0000	sp 0000	d=0
cs:0015	F7F3	div	bx	ds 44AD	
cs:0017	83FA00	cmp	dx,0000	es 449D	
cs:001A	7407	je	0023	ss 44AC	
cs:001C	8BC2	mov	ax,dx	cs 44AE	
cs:001E	EBEC	jmp	000C	ip 0020	
cs:0020	93	xchg	bx,ax		
cs:0021	EBEF	jmp	0012		

ds:0000	0A 00 04 00 00 00 00 00	ss:0008	6864
ds:0008	00 00 00 00 00 00 00 00	ss:0006	742E
ds:0010	B8 AD 44 8E D8 A1 00 00	ss:0004	706C
ds:0018	8B 1E 02 00 3B C3 74 13	ss:0002	6568
ds:0020	72 0E BA 00 00 F7 F3 83	ss:0000	6474

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

44AE:0012	BA0000	mov	dx,0000	ax 0192	c=1
44AE:0015	F7F3	div	bx	bx 02F4	z=0
44AE:0017	83FA00	cmp	dx,0000	cx 00D8	s=1
44AE:001A	7407	je	0023	dx 0990	o=0
44AE:001C	8BC2	mov	ax,dx	si 0019	p=0
44AE:001E	EBEC	jmp	000C	di 0FA6	a=0
44AE:0020	93	xchg	bx,ax	bp 0100	i=1
44AE:0021	EBEF	jmp	0012	sp 0106	d=1
44AE:0023	891E0400	mov	[0004],bx	ds 1D09	
44AE:0027	B44C	mov	ah,4C	es 02F4	
44AE:0029	CD21	int	21	ss 0192	
44AE:002B	0000	add	[bx+si],al	cs 0000	
44AE:002D	0000	add	[bx+si],al	ip 0000	
44AE:002F	0000	add	[bx+si],al		
44AE:0031	0				

44AD:0000	0	44AC:0008	6864
44AD:0008	0	44AC:0006	742E
44AD:0010	B	44AC:0004	706C
44AD:0018	8	44AC:0002	6568
44AD:0020	7	44AC:0000	6474

Terminated, exit code 2

OK Help