

כתיבה והרצה תוכנית ראשונה בשפת assembly

במעבדה של קורס הזה נעבוד רק בסביבת Tasm 1.4- למעבד 8086 וירטואלי. קיצור דרך של יישום נראה:



1. כניסה לסביבת העבודה:

```
+=====+
Power Programming Tools 64 Bit By Chaitanya Patel : Techapple.Net
+=====+

Type Proper Command To Perform the Desired Action

Command      .      Action
-----
Edit          -      Open MS-DOS Editor
TASM          -      Compilation
tlink         -      Perform Linking
td            -      Launch Turbo Debugger
Exit          -      Exit Tasm 1.2

For Compiling your files tasm "yourfilename".asm use without quotes
e.g for compiling add.asm command is : tasm add.asm
For Linking and debugging same as 32 bit : tlink,td

Complink,DPMIload and TasmX also available using 32bit commands

C:\TASM>_
```

2. הקלדת קוד התוכנית מתבצע רק ב-text editor בשם edit.

```
+=====+
Power Programming Tools 64 Bit By Chaitanya Patel : Techapple.Net
+=====+

Type Proper Command To Perform the Desired Action

Command      .      Action
-----
Edit          -      Open MS-DOS Editor
TASM          -      Compilation
tlink         -      Perform Linking
td            -      Launch Turbo Debugger
Exit          -      Exit Tasm 1.2

For Compiling your files tasm "yourfilename".asm use without quotes
e.g for compiling add.asm command is : tasm add.asm
For Linking and debugging same as 32 bit : tlink,td

Complink,DPMIload and TasmX also available using 32bit commands

C:\TASM>edit t1.asm_
```

גודל תא זיכרון הוא byte אחד, לכן נשתמש בחצאים של אוגרים להתאמת גודל. גודל של "חצי אוגר" הוא גם byte אחד (8 סיביות-bit).

```

File Edit Search View Options Help
C:\TASMNT1.ASM
code segment
assume cs:code,ds:code
start:  mov ax,code
        mov ds,ax
        mov al,48
        mov ah,72
        mov bx,100h
        add al,ah
        mov [bx],al
        mov ah,4ch
        int 21h
code ends
end start_

```

F1=Help | Line:13 Col:10

תבנית התוכנית

```

code segment
assume cs:code,ds:code
start:  mov ax,code
        mov ds,ax
        mov ah,4ch
        int 21h
code ends
end start

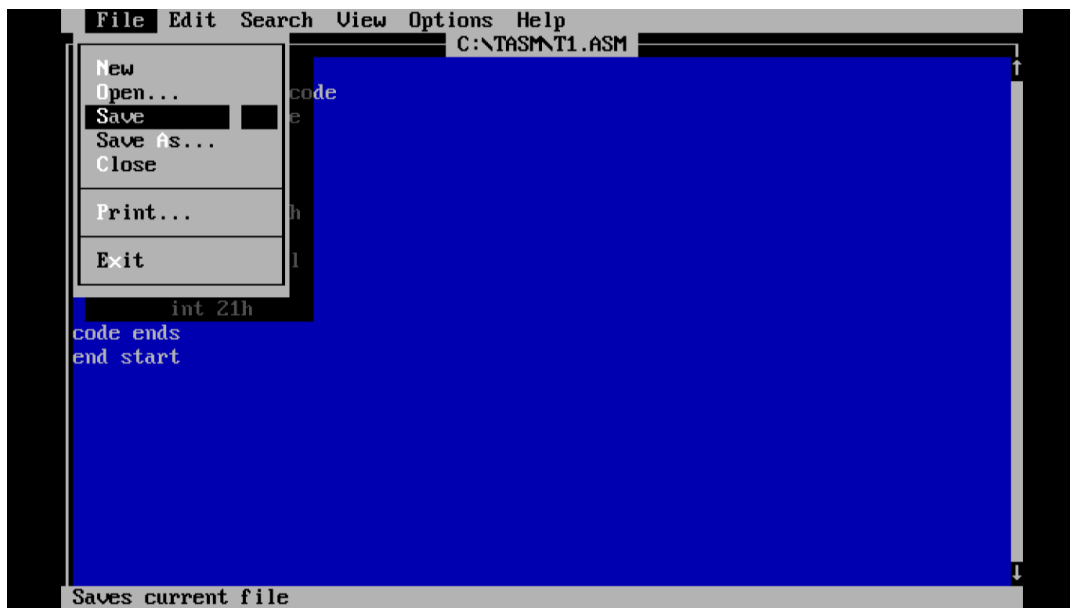
```

} העתקת כתובת תחילת התוכנית לאוגר
 } יציאה ל-DOS
 ; סוף מקטע
 ; סוף התוכנית

גוף תוכנית

- המילה **segment** מציינת שזהו מקטע של תוכנית אסמבלי שגודלו המקסימלי 64K.
- המילה **code** נבחרה כדי לייצג את שם מקטע התוכנית.
- המילה **assume** מרחיבה את ההסבר אודות מקטע תוכנית. אוגר CS אשר מכיל את כתובת ההתחלה של תוכנית, ותחיל במקטע בשם code, כך גם אותה כתובת נמצאת באוגר DS.

לשמירת קובץ נשתמש בתפריט File. (חשוב מאוד לשמור קובץ עם סיומת asm)



יציאה מ-edit: File → Quit – סיימנו **שלב א'** – הקלדת קוד תוכנית.
שלב ב' – בדיקת syntax errors וחישוב כתובות לפקודות התוכנית. בשלב זה תתורגם התוכנית לקובץ בשפת מכונה. הקלט הוא תוכנית המקור שנכתבה בשפת assembly שהסיומת asm והפלט הוא התוכנית מתורגמת שהסיומת שלה היא obj. הקלט:

```
tasm t1,;
```

שלב ג' – בשלב זה נוצר מהתוכנית קובץ בינארי מוכן להרצה. קלט הוא קובץ עם סיומת obj ופלט הוא קובץ עם סיומת exe.

```
tlink t1,;
```

לא הגדרנו מקטע מחסנית (stack segment) – לא טעות.
שלב ד' – הרצה / ניפוי. להריץ תוכנית בסביבת DOS- יש להריץ רק: t1.
 התוכנית שלנו לא מחזירה (מדפיסה על המסך) שום ערך. לכן אנו לא רואים את התוצאה.
 כדי לראות תוצאה של חישובים, נריץ את התוכנית בסביבת debugger.

להריץ בסביבת (turbo debugger) td :

```
td          - Launch Turbo Debugger
Exit        - Exit Tasm 1.2

-----
For Compiling your files tasm "yourfilename".asm use without quotes
e.g for compiling add.asm command is : tasm add.asm
For Linking and debugging same as 32 bit : tlink,td

Complink,DPMIload and TasmX also available using 32bit commands
-----

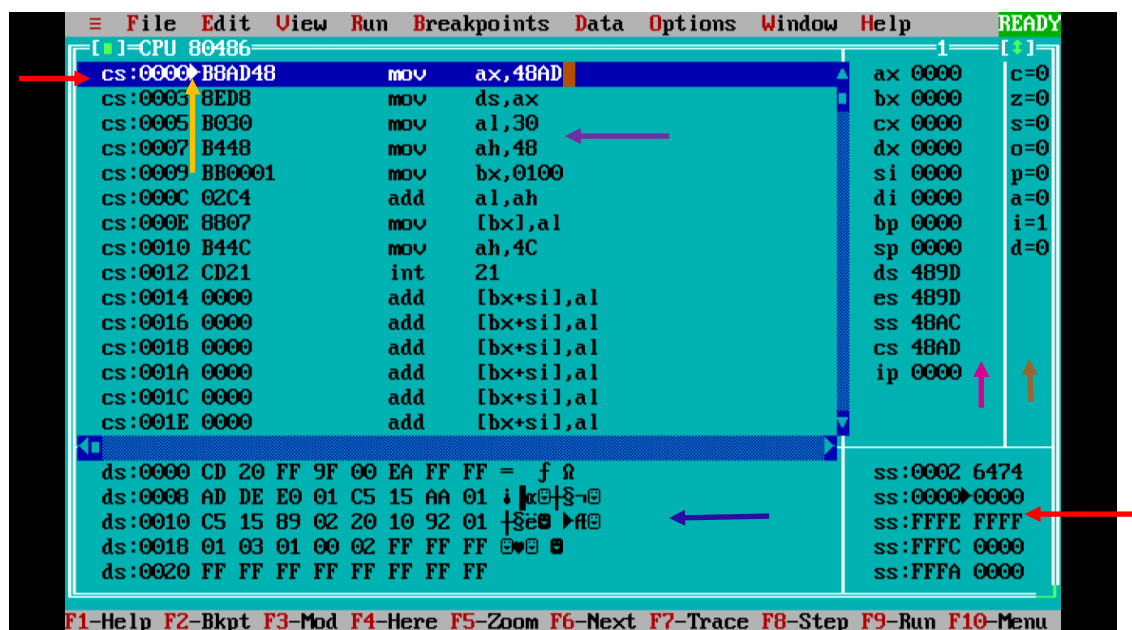
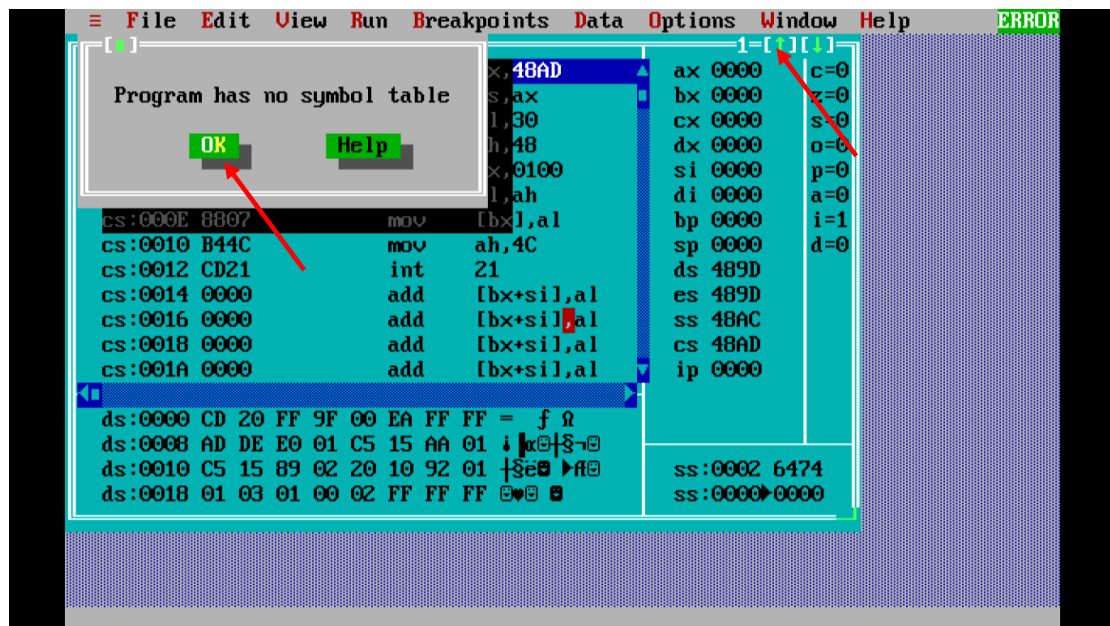
C:\TASM>tasm t1,;
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  t1.ASM
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 471k

C:\TASM>tlink t1,;
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>td t1_
```

אחרי כניסה ל-td לוחצים על לחצן "ok" ואחר-כך על החץ כדי להגדיל את החלון על כל המסך.



חלון של debugger:

- חלון מסומן: חלון של קוד תוכנית. בצד ימין של חלון אנו רואים קוד תוכנית שהקלדנו ב-edit. הקלדנו ערכים עשרוניים וב-debugger רואים אותם בבסיס 16. בצד שמאל של החלון – כתובות של כל פקודות במקטע קוד. כתובת פקודה המסומנת במשולש היא כתובת פקודה להרצה. בדיוק אותה כתובת נמצאת באוגר IP
- למטה נמצא חלון זיכרון.
- בצד ימין נמצאים חלונות אוגרים ותוכן אוגר הדגלים.
- בצד ימין למטה נמצא חלון מחסנית (stack)

כדי להריץ תכנית step by step (בצעדים) לוחצים על F7. המשולש שמסמן פקודה הבאה לריצה עובר כתובת פקודה הבאה. אוגר IP מקבל כתובת פקודה הבאה. אוגר AX מקבל כתובת תחילת מקטע קוד כל שינויים אחרי כל צעד מסומנים בצבע לבן (תסתכל בחצים).

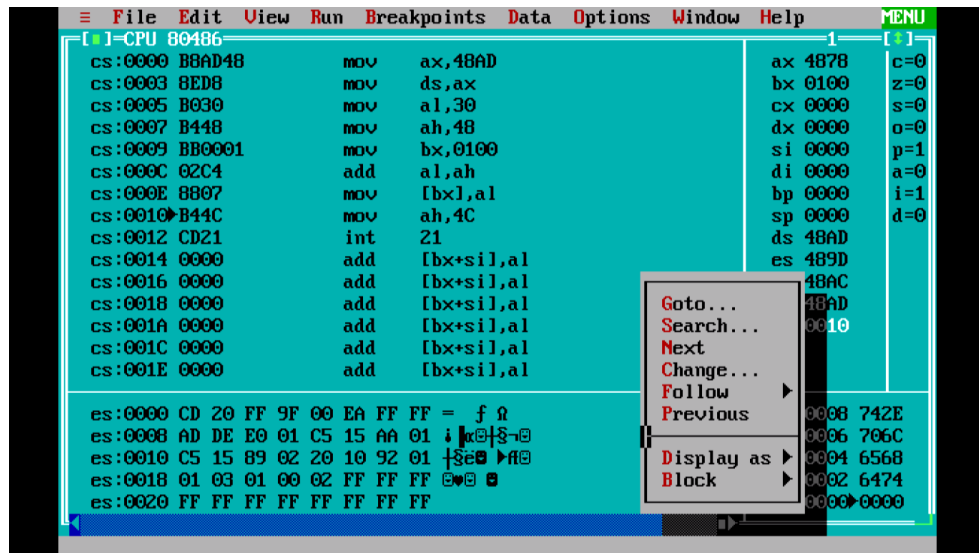
Address	Instruction	Comment
cs:0000 B8AD48	mov ax,48AD	
cs:0003 8ED8	mov ds,ax	
cs:0005 B030	mov al,30	
cs:0007 B448	mov ah,48	
cs:0009 BB0001	mov bx,0100	
cs:000C 02C4	add al,ah	
cs:000E 8807	mov [bx],al	
cs:0010 B44C	mov ah,4C	
cs:0012 CD21	int 21	
cs:0014 0000	add [bx+si],al	
cs:0016 0000	add [bx+si],al	
cs:0018 0000	add [bx+si],al	
cs:001A 0000	add [bx+si],al	
cs:001C 0000	add [bx+si],al	
cs:001E 0000	add [bx+si],al	

מריצים תכנית עד סוף הקוד (במקרה שלנו כתובת 10h).

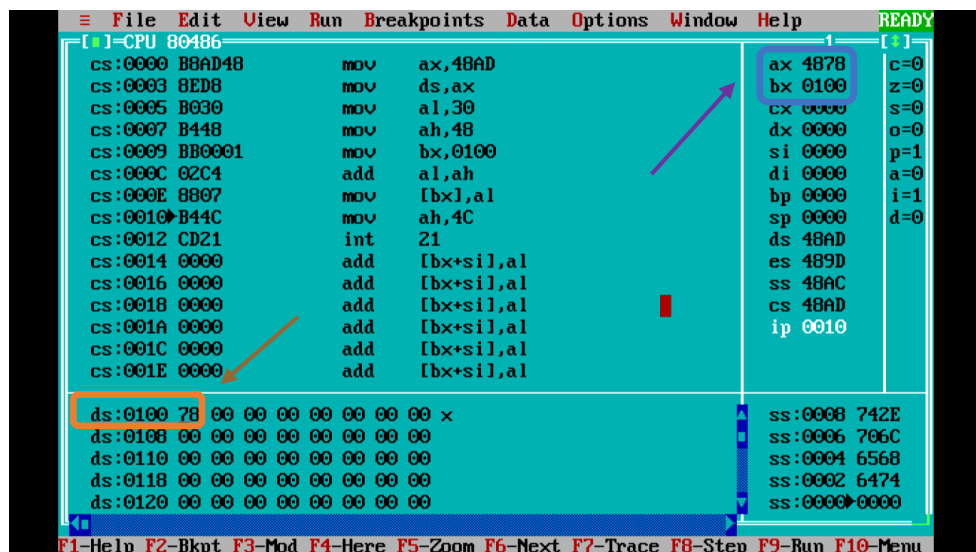
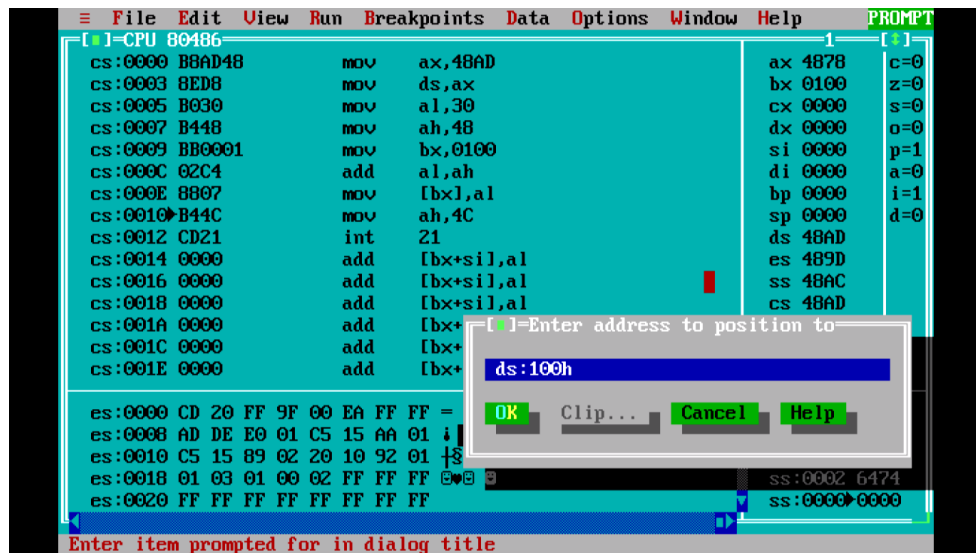
Address	Instruction	Comment
cs:0000 B8AD48	mov ax,48AD	
cs:0003 8ED8	mov ds,ax	
cs:0005 B030	mov al,30	
cs:0007 B448	mov ah,48	
cs:0009 BB0001	mov bx,0100	
cs:000C 02C4	add al,ah	
cs:000E 8807	mov [bx],al	
cs:0010 B44C	mov ah,4C	
cs:0012 CD21	int 21	
cs:0014 0000	add [bx+si],al	
cs:0016 0000	add [bx+si],al	
cs:0018 0000	add [bx+si],al	
cs:001A 0000	add [bx+si],al	
cs:001C 0000	add [bx+si],al	
cs:001E 0000	add [bx+si],al	

כדי לראות תוצאה בכתובת מסוימת בזיכרון אנו משתמשים בעכבר: בחלון זיכרון:

- "קליק" ימני
- "קליק" שמאלי
- בתפריט נופלת בוחרים אופציה go to.

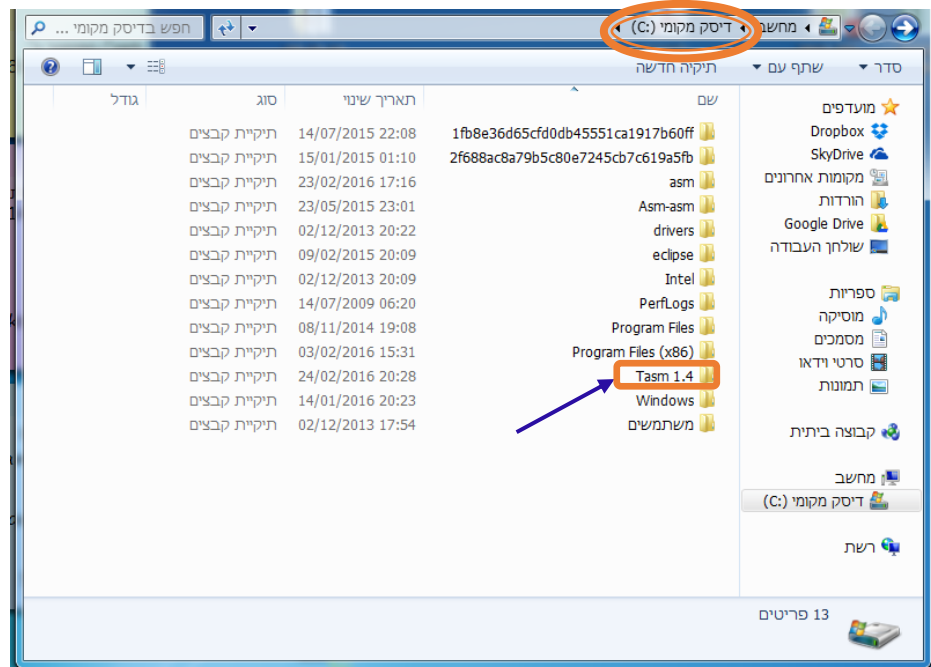


בחלון שנפתח כתובים תא זיכרון שרוצים לבדוק



כד לצאת מסביבת העבודה נשתמש בפקודה exit .

איפה נשמר קובץ המקורי (t1.asm)?



כדי לעבוד בבית כמו בכיתה:

צריך לעבור שלבים:

1. להעתיק תיקיה לשולחן עבודה של מחשב.
2. להריץ את קובץ. נקבל קיצור דרך לסביבת עבודה

עבודה נעימה