## Instalarea Turbo Assembler

In vederea rulării TASM pentru orice sistem diferit de DOS, avem nevoie în principiu, de un emulator de DOS. Pentru Windows poate fi folosit DOSBox+X, un utilitar ce poate fi download-at de pe siteul <a href="https://github.com/joncampbell123/dosbox-x/releases">https://github.com/joncampbell123/dosbox-x/releases</a>. Este de preferat ca DOSBox-X sa fie instalat in locatia implicită.

Următorul pas constă în dezarhivarea arhivei TASM.zip într-un director pregatit pentru lucru (e.g. C: \TASM).

Se rulează emulatorul DOS și se montează locația TASM-ului, comană prin care driverul C din emulator devine path-ul de lucru cu TASM.

```
Z:\> mount c C:\TASM
```

În urma acestui proces veți primi un mesaj de confirmare:

"Drive C is mounted as local directory C:\TASM,

```
Main CPU Video Sound DOS Capture Drive Debug Help
 Getting started with DOSBox-X:
 Type HELP to see the list of shell commands, INTRO for a brief introduction.
 You can also complete various tasks in DOSBox-X through the drop-down menus.
 Example: Try select the TrueType font or OpenGL pixel-perfect output option.
 Useful default shortcuts:
 - switch between windowed and full-screen mode with key combination F11 + F
 - launch Configuration Tool using F11 + C, and Mapper Editor using F11 + M
  - increase or decrease the emulation speed with F11 + Plus or F11 + Minus
 DOSBox-X project on the web:
 Homepage of project: https://dosbox-x.com/
                                                        Complete DOS emulations
                                                        DOS, Windows 3.x and 9x
 User guides on Wiki: https://dosbox-x.com/wiki
 Issue or suggestion: https://github.com/joncampbell123/dosbox-x/issues
Z:\>mount c c:\tasm\bin
Drive C is mounted as local directory c:\tasm\bin\
Z:\>c:
```

Din acest moment ar trebui să aveți posibilitatea asamblării și linkării programelor (de exemplu programul ex1.asm)

## Compilarea și rularea unei aplicații ASM

Turbo Assembler-ul are un help inline care se poate apela cu parametrul /h:

```
> tasm /h
```

## Asamblarea se poate executa cu comanda

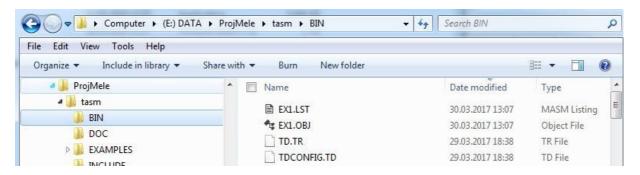
```
> tasm <nume sursa> /zi /la
```

Unde /zi: produce toate informaţiile de debug şi /la: generează listing-ul expandat.

```
C:\BIN>tasm ex1.asm /zi /la
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International
Assembling file: ex1.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 460k

C:\BIN>_
```

Rezultă generarea a două fișiere OBJ și LST:

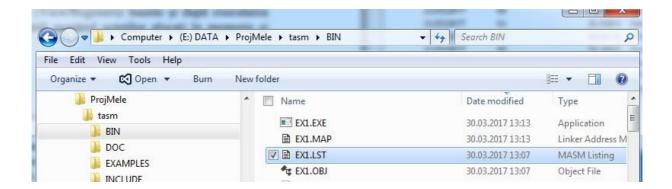


Linkeditarea se face cu comanda:

> tlink /v <nume\_sursa>

```
C:\BIN>tlink /v ex1.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
C:\BIN>_
```

Unde /v: va produce toate informațiile symbol pentru debug și va produce cele două fișiere EXE și MAP.



În acest stadiu va trebui sa inspectați cele două fișiere tip ASCII produse până acum.

Execuția se face prin simpla apelare a executabilului EX1:

```
> ex1<.exe>
```

## Depanarea unei aplicații ASM

Pentru a intra in sesiunea de debugging pe aplicația EX1, se apelează TD (Turbo Debugging):

> td ex1.exe

```
C:\BIN>td ex1.exe
Turbo Debugger Version 5.0 Copyright (c) 1988,96 Borland International
```

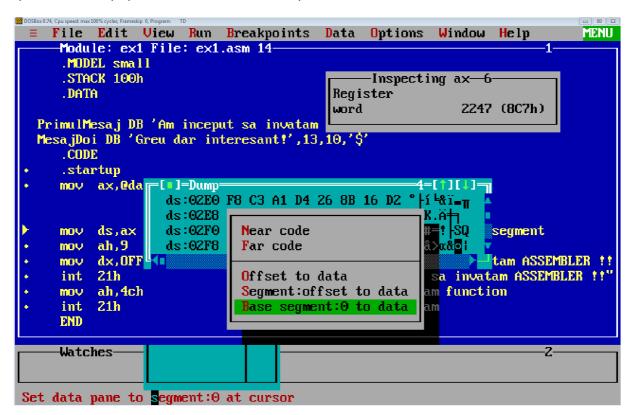
Acum se va deschide aplicația Turbo Debugger:

```
File Edit View Run Breakpoints Data Options Window Help
                                                                                 READS
  [■]=Module: ex1 File:
                          ex1.asm 14
                                                                            -1=[ † ] [ ↓ ]=
                         Copyright (c) 1988, 1991 By Borland International, Inc.
     Turbo Assembler
   : Exercitiu1.ASM - Afiseaza mesajul "Invatam ASSEMBLER !!"
       .MODEL small
       STACK 100h
       . DATA
   PrimulMesaj DB 'Am inceput sa invatam ASSEMBLER !!',13,10,'$'
MesajDD 'Greu dar interesant!',13,10,'$'
      . CODE
       .startup
      mo∨ ax,@data
                                ; cu debug mergeti la adresa data
                               si cu offset-ul respectiv
                               inspectati sirul
                                     set DS to point to the data segment
      mov ds,ax
      Watches
                                byte [20] "Greu dar interesant!"
F1-Help F2-Bknt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

Din menu alegem Data -> Inspect și tastăm ax, va rezulta:

```
READY
      File Edit View Run Breakpoints Data Options Window Help
       Module: ex1 File: ex1.asm 14
       .MODEL small
       .STACK 100h
       . DATA
                                                    Register
   2247 (8C7h)
       . CODE
       .startup
       mo∨ ax,@data
                                     ; cu debug mergeti la adresa data
                                 ; si cu offset-ul respectiv
; inspectati sirul
                                          ;set DS to point to the data segment
;DOS print string function
;point to "Am inceput sa invatam ASSEMBLER !!"
;display "Am inceput sa invatam ASSEMBLER !!"
             ds,ax
       mov
             ah,9
d×,0FFSET PrimulMesaj
       MOV
       MOV
       int
             21h
                                          :DOS terminate program function terminate the program
       MOV
             ah,4ch
       int
             21h
       END
       Watches-
                                                                                       Z-
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

Apoi facem "dump" pe locatia de la adresa 0x8C7 pentru a vedea datele:



Introducem segmentul și offsetul pentru locația de memorie care pointează la variabila noastră:



Şi vom putea inspecta variabila căutată:

```
ds:0000 41 6D 20 69 6E 63 65 70 Am incep
ds:0008 75 74 20 73 61 20 69 6E ut sa in
ds:0010 76 61 74 61 6D 20 41 53 vatam AS
ds:0018 53 45 4D 42 4C 45 52 20 SEMBLER
```