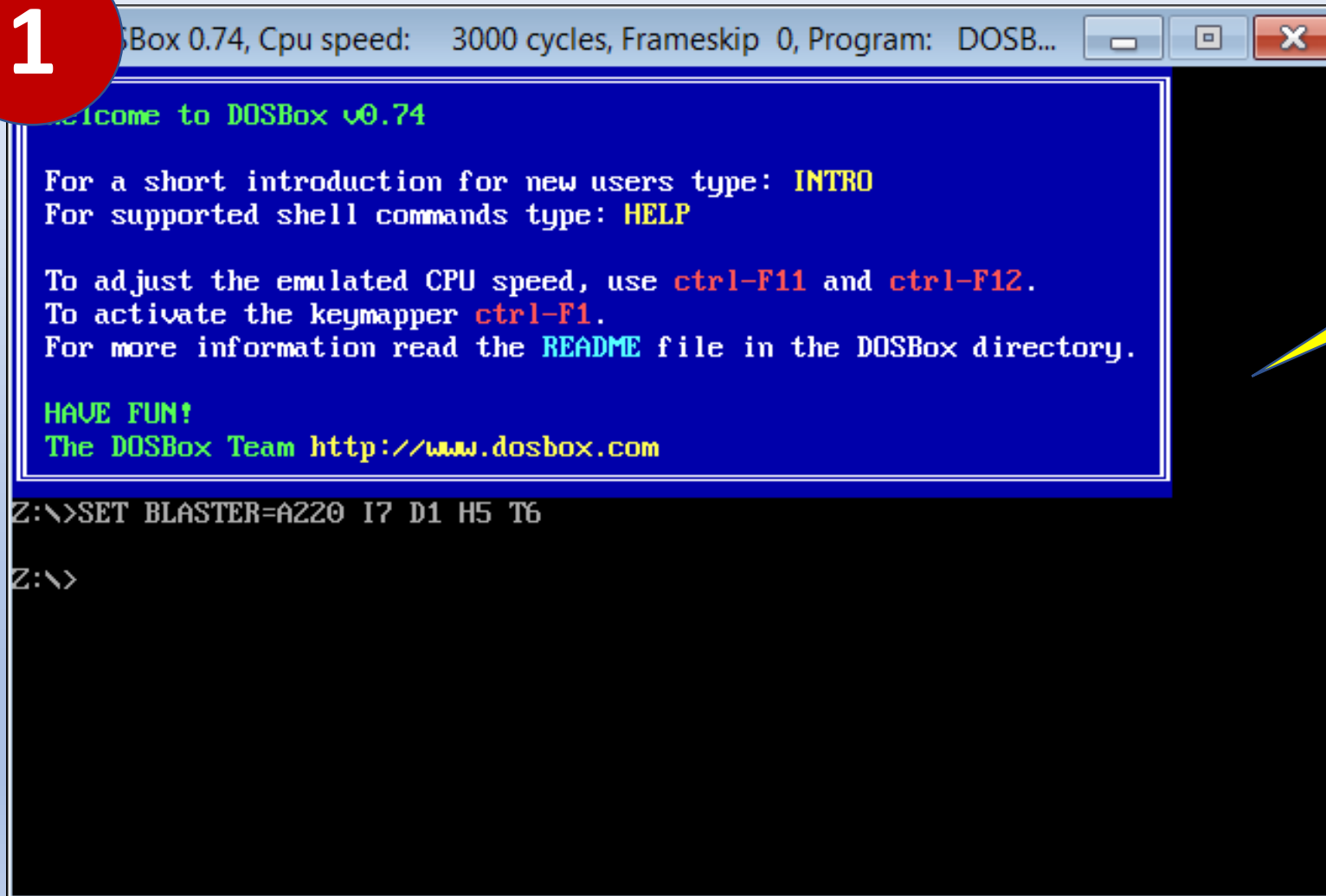


Διαδικασία εκτέλεσης του προγράμματος Turbo Debugger

Δρ. Παναγιώτης Παπάζογλου

Διαδικασία εκτέλεσης του προγράμματος Turbo Debugger

1



```
Box 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSB...  
Welcome to DOSBox v0.74  
For a short introduction for new users type: INTRO  
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:\>SET BLASTER=A220 I7 D1 H5 T6  
  
Z:\>
```

Αρχικά
εκτελούμε το
DOSBox

Διαδικασία εκτέλεσης του προγράμματος Turbo Debugger

2

```
SBBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSB...
Welcome to DOSBox v0.74
For a short introduction for new users type: INTRO
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:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount c c:\td
Drive C is mounted as local directory c:\td\
Z:\>c:
C:\>td
```

Κάνουμε αντιστοίχιση ενός drive letter στον κατάλογο που βρίσκεται ο debugger, μεταφερόμαστε εκεί και εκτελούμε το πρόγραμμα

mount c c:\td

Αντιστοίχιση του c στον πραγματικό κατάλογο c:\td

c:

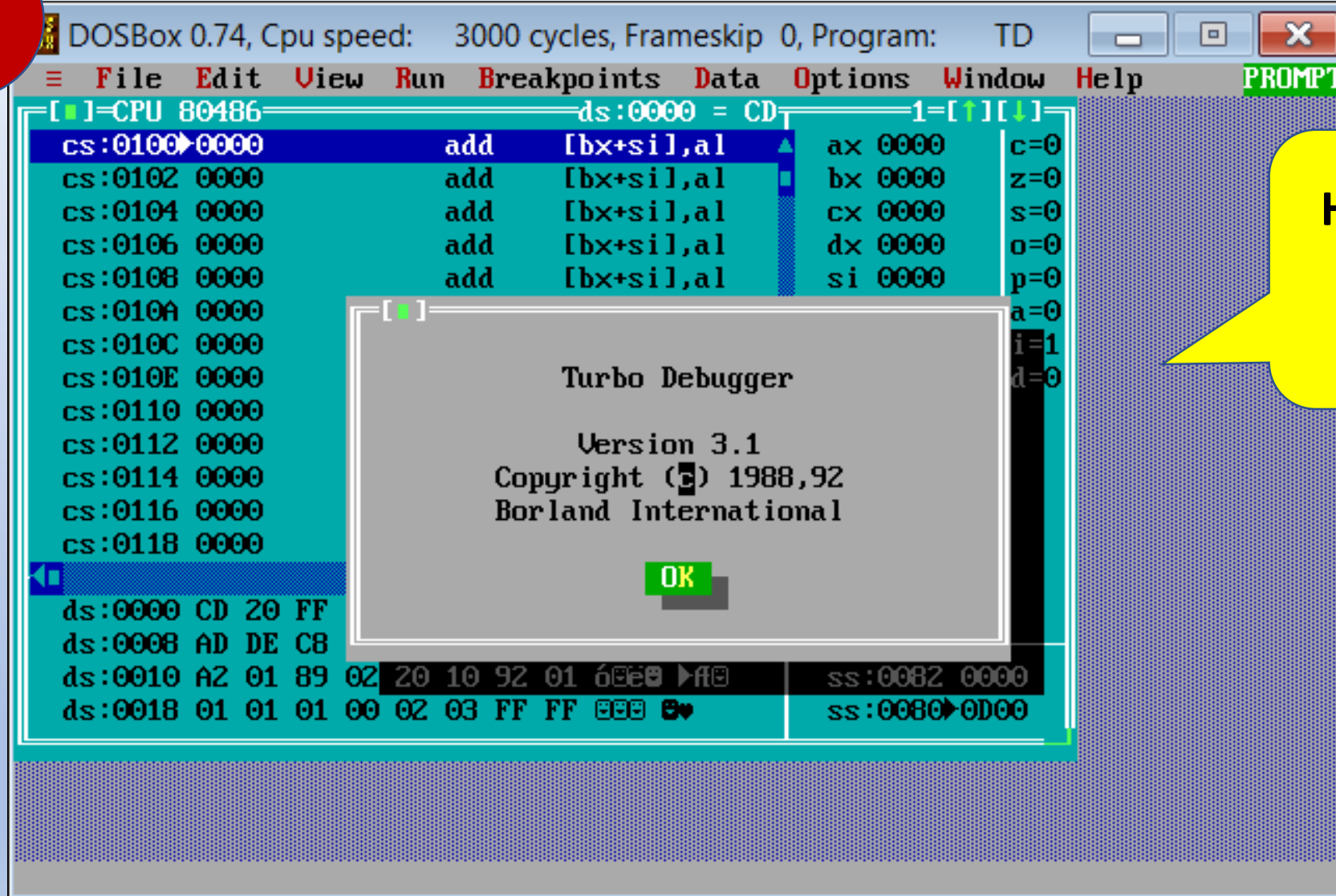
Μεταφορά στο c

td

Εκτέλεση του debugger

Διαδικασία εκτέλεσης του προγράμματος Turbo Debugger

3



Η εφαρμογή του Debugger είναι τώρα ενεργή

Το περιβάλλον του Debugger (1)

The screenshot shows the DOSBox 0.74 debugger interface. The title bar indicates 'DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Breakpoints', 'Data', 'Options', 'Window', and 'Help'. The status bar at the bottom shows 'Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local'.

The main window is divided into several panes:

- Disassembly Pane (top left):** Shows assembly instructions. A red box highlights a block of code from address `cs:0100` to `cs:011C`, all of which are `add [bx+si],al`. A yellow callout bubble points to this area with the text 'Τμήμα προγράμματος' (Code block).
- Registers Pane (top right):** Lists registers and their values: `ax 0000`, `bx`, `cx`, `dx`, `si`, `di`, `sp`, `ds 4893`, `es 4893`, `ss 4893`, `cs 4893`, and `ip 0100`.
- Memory Pane (bottom left):** Shows memory contents in hexadecimal and ASCII: `ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω`, `ds:0008 AD DE C8 20 00 F0 CC 0A ; |" ≡ ||o`, `ds:0010 A2 01 89 02 20 10 92 01 óEe0 ▶f@`, `ds:0018 01 01 01 00 02 03 FF FF 000 0♥`, and `ds:0020 FF FF FF FF FF FF FF`.
- Stack Pane (bottom right):** Shows stack memory addresses and values: `ss:0082 0000`, `ss:0080 0D00`, `ss:007E 0000`, `ss:007C 0000`, and `ss:007A 0000`.

Το περιβάλλον του Debugger (2)

The screenshot displays the DOSBox 0.74 debugger interface. The title bar shows 'DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Breakpoints', 'Data', 'Options', 'Window', and 'Help'. The status bar at the bottom shows 'Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local'.

The main window is divided into several sections:

- Assembly List:** A list of assembly instructions with their addresses and hex values. The current instruction is highlighted in blue: `cs:0100 0000 add [bx+si],al`. Other instructions include `cs:0102 0000 add [bx+si],al` through `cs:011C 0000 add [bx+si],al`.
- Registers:** A list of registers and their values: `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 0080 d=0`, `ds 4893`, `es 4893`, `ss 4893`, `cs 4893`, and `ip 0100`.
- Memory Dump:** A section showing memory addresses and their contents. A red box highlights the following data: `ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω`, `ds:0008 AD DE CB 20 00 F0 CC 0A ; |u ≡ |o`, `ds:0010 A2 01 89 02 20 10 92 01 óεεθ ▶fθ`, `ds:0018 01 01 01 00 02 03 FF FF εεε εε`, and `ds:0020 FF FF FF FF FF FF FF FF`.

A yellow callout box with the text 'Τμήμα δεδομένων' (Data section) points to the memory dump area.

Το περιβάλλον του Debugger (3)

The screenshot shows the DOSBox 0.74 debugger interface. The title bar reads "DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD". The menu bar includes "File", "Edit", "View", "Run", "Breakpoints", "Data", "Options", "Window", "Help", and "READY". The main window is titled "[]=CPU 80486" and shows the following assembly code:

```
cs:0100 0000 add [bx+si],al
cs:0102 0000 add [bx+si],al
cs:0104 0000 add [bx+si],al
cs:0106 0000 add [bx+si],al
cs:0108 0000 add [bx+si],al
cs:010A 0000 add [bx+si],al
cs:010C 0000 add [bx+si],al
cs:010E 0000 add [bx+si],al
cs:0110 0000 add [bx+si],al
cs:0112 0000 add [bx+si],al
cs:0114 0000 add [bx+si],al
cs:0116 0000 add [bx+si],al
cs:0118 0000 add [bx+si],al
cs:011A 0000 add [bx+si],al
cs:011C 0000 add [bx+si],al
```

Registers are listed on the right:

```
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 0080 d=0
ds 4893
es 4893
ss 4893
cs 4893
ip 0100
```

Below the assembly code, memory dump data is shown:

```
ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE CB 20 00 F0 CC 0A ; |" ≡
ds:0010 A2 01 89 02 20 10 92 01 óεεθ ▶
ds:0018 01 01 01 00 02 03 FF FF εεε ε▶
ds:0020 FF FF FF FF FF FF FF FF
```

A yellow callout bubble points to the memory dump with the text "Τμήμα σωρού" (Garbage area). A red box highlights the stack segment registers:

```
ss:0082 0000
ss:0080 0D00
ss:007E 0000
ss:007C 0000
ss:007A 0000
```

At the bottom, a status bar shows keyboard shortcuts: "Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local".

Το περιβάλλον του Debugger (4)

The screenshot displays the DOSBox 0.74 debugger interface. The title bar shows 'DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Breakpoints', 'Data', 'Options', 'Window', and 'Help'. The status bar at the bottom shows 'Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local'.

The main window is divided into several sections:

- Assembly Code:** A list of instructions starting with 'add [bx+si],al' at memory addresses from 0100 to 011C. A red vertical bar is positioned at the instruction at address 0104.
- Registers:** A table on the right side of the assembly code, enclosed in a red box, showing the current values of various registers: ax (0000), bx (0000), cx (0000), dx (0000), si (0000), di (0000), bp (0000), sp (0080), ds (4893), es (4893), ss (4893), cs (4893), and ip (0100). To the right of these values are flags: c=0, z=0, s=0, o=0, p=0, a=0, i=1, and d=0.
- Memory:** A section at the bottom left showing memory contents for segments ds:0000, ss:0082, ss:0080, ss:007E, ss:007C, and ss:007A.

A yellow speech bubble with the Greek text 'Καταχωρητές' (Registers) points to the register table. The register table itself is highlighted with a red border.

Το περιβάλλον του Debugger (5)

The screenshot shows the DOSBox 0.74 debugger interface. The title bar indicates 'DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD'. The menu bar includes File, Edit, View, Run, Breakpoints, Data, Options, Window, and Help. The status bar at the bottom shows keyboard shortcuts: Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local.

The main window displays assembly code for the instruction `add [bx+si],al` at memory address `cs:0100`. A yellow callout bubble with the text **Καταχωρητής κατάστασης** (Status Register) points to the register list on the right. The register list shows the following values:

- c=0
- z=0
- s=0
- o=0
- p=0
- a=0
- i=1
- d=0

Below the assembly code, the memory dump shows the contents of the data segment (ds) starting at address `ds:0000`. The registers `bp`, `sp`, `ds`, `es`, `ss`, `cs`, and `ip` are also listed with their current values.

```
cs:0100 0000 add [bx+si],al
cs:0102 0000 add [bx+si],al
cs:0104 0000 add [bx+si],al
cs:0106 0000 add [bx+si],al
cs:0108 0000 add [bx+si],al
cs:010A 0000 add [bx+si],al
cs:010C 0000 add [bx+si],al
cs:010E 0000 add [bx+si],al
cs:0110 0000 add [bx+si],al
cs:0112 0000 add [bx+si],al
cs:0114 0000 add [bx+si],al
cs:0116 0000 add [bx+si],al
cs:0118 0000 add [bx+si],al
cs:011A 0000 add [bx+si],al
cs:011C 0000 add [bx+si],al

ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE CB 20 00 F0 CC 0A ; |u ≡ ||o
ds:0010 A2 01 89 02 20 10 92 01 ó|e|0 ▶|f|0
ds:0018 01 01 01 00 02 03 FF FF 0000 00
ds:0020 FF FF FF FF FF FF FF FF

bp 0000
sp 0080
ds 4893
es 4893
ss 4893
cs 4893
ip 0100

ss:0082 0000
ss:0080 0D00
ss:007E 0000
ss:007C 0000
ss:007A 0000
```

Το περιβάλλον του Debugger (6)

The screenshot shows the DOSBox 0.74 debugger interface. The window title is "DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD". The menu bar includes File, Edit, View, Run, Breakpoints, Data, Options, Window, and Help. The main display area is divided into several sections:

- Code Segment (CS):** A list of memory addresses and their corresponding assembly instructions. The instructions are all "add [bx+si],al". A red box highlights this section, with a yellow callout bubble pointing to it that says "Εντολές Assembly (πρόγραμμα)".
- Registers:** A list of registers and their values. The registers shown are ax, bx, cx, dx, si, di, bp, sp, ds, es, ss, cs, and ip. The values are mostly 0000, except for ds, es, ss, cs, and ip which are 4893, 4893, 4893, 4893, and 0100 respectively.
- Memory Dump:** A section showing memory addresses and their contents. The address ds:0020 contains FF FF FF FF FF FF FF FF.
- Registers (continued):** A list of registers and their values. The registers shown are ss:0082, ss:0080, ss:007E, ss:007C, and ss:007A, all with values 0000.

At the bottom of the window, there is a status bar with the following text: "Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local".

Annotations:

- A red box highlights the assembly code section, with a red callout bubble pointing to it that says "Τμήμα προγράμματος".
- A yellow callout bubble points to the memory addresses in the code segment, saying "Διευθύνσεις στο τμήμα κώδικα (Code Segment)".
- A yellow callout bubble points to the assembly instructions, saying "Κώδικας εντολής".

Το περιβάλλον του Debugger (7)

The screenshot shows the DOSBox 0.74 debugger interface. The title bar reads "DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD". The menu bar includes "File", "Edit", "View", "Run", "Breakpoints", "Data", "Options", "Window", and "Help". The status bar at the bottom shows "F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn F9-To F10-Local".

The main window displays assembly code for the CPU 80486. The code consists of a series of "add [bx+si],al" instructions at memory addresses from 0100 to 011C. The registers on the right are shown with their current values: ax=0000, bx=0000, cx=0000, dx=0000, si=0000, di=0000, bp=0000, sp=0080, ds=4893, es=4893, ss=4893, cs=4893, and ip=0100.

A red box highlights a memory dump in the "Data" window, showing addresses from ds:0008 to ds:0020. The contents are: AD DE CB 20 00 F0 CC 0A, A2 01 89 02 20 10 92 01, 01 01 01 00 02 03 FF FF, and FF FF FF FF FF FF FF FF. A yellow callout points to this area with the text "Περιεχόμενα".

A red callout points to the right side of the debugger with the text "Τμήμα δεδομένων".

A yellow callout points to the left side of the debugger with the text "Διευθύνσεις στο τμήμα δεδομένων (Data Segment)".

Οδηγίες για εγκατάσταση στο σπίτι

(1) Δημιουργούμε τους ακόλουθους καταλόγους στο drive C του υπολογιστή μας:

dosbox (δηλαδή c:\dosbox)

td (δηλαδή c:\td)

(2) Στην εγκατάσταση του dosbox επιλέγουμε ως προορισμό το **c:\dosbox**

(3) Αντιγράφουμε το TD (εφαρμογή debugger) στον κατάλογο **c:\td**

Στη συνέχεια ακολουθούμε τα βήματα που αναφέρονται στις διαφάνειες για την εκτέλεση της εφαρμογής