



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

## Programare în limbaj de asamblare

**42. Utilizare Debug și TurboDebugger pentru realizare  
de programe simple și depanarea lor.**

## Utilitarele *Debug* și *Turbo Debugger*

### Debug

Această utilitară este un program de depanare care furnizează un mediu de testare controlat pentru fișiere obiect binare și executabile; elimină nevoia de reasamblare a programului pentru a vedea dacă o problemă a fost rezolvată printr-o modificare minoră.

El permite să se modifice conținutul unui fișier sau conținutul registrelor CPU, și apoi imediat, să se reexecute un program pentru a verifica validitatea modificărilor făcute.

Toate comenzile **debug** pot fi abandonate în orice moment prin tastarea **Ctrl-C**. Secvența **Ctrl-S** suspendă afișarea, astfel ca să poată fi citita imaginea de pe ecran, înainte de a se continua defilarea liniilor pe ecran. Afișarea se reia cu orice tastă, în afară de **Ctrl-C** sau **Ctrl-S**.

Sunt două moduri în care poate fi lansat programul de depanare **debug** :

- primul permite toate comenzile ca răspuns la promptul sau;
- cel de-al doilea permite, în plus, specificarea unor comenzi și în linia de comandă, utilizată pentru a lansa **debug**.

### Metoda 1:

**>debug**

Se va lansa în execuție programul **debug**, care va răspunde cu promptul (-), semnalând că este gata de a primi comenzi. Se poate lucra în memoria curentă, sectoare de disc sau fișiere de pe disc.

Dacă se primește mesajul

**Program terminated normally**

nu relansați un program; trebuie reîncărcat programul cu comenzile **N** (Name) sau **L** (Load) pentru o execuție adecvată.

### Metoda 2:

**>debug [ nume\_fisier [ lista\_argumente ]]**

de exemplu :

**>debug fisier.exe**

În urma acestei comenzi se va încărca în memorie, pe lângă utilitara de depanare - debug, și fisier.exe, începând de la adresa *100h*, în segmentul disponibil, cel mai de jos. În registrele *BX:CX* se încarcă numărul de octeți (în hexazecimal) ocupați în memorie de programul încărcat (fisier.exe).

Câmpul **lista\_arumente** - reprezintă restul liniei de comandă utilizată de către *debug*, când apelează și încarcă fișierul în memorie.

Când se depanează un fișier **.exe** sau **.com** nu trebuie modificat conținutul memoriei mai jos de adresa *5Ch* sau, altfel, programul de depanare se va termina.

Fiecare comandă constă dintr-o singură literă urmată de unul sau mai mulți parametrii. Iată, enumerate, aceste comenzi:

**A** [ adresă ] - asamblare de la '*adresa*' sau *100h*;

**C** domeniu adresă - compară două zone de memorie;

**D** [ domeniu ] - afișează conținutul unei zone de memorie;

**E** adresă [ listă ] - introduce '*lista*' de octeți de la adresa specificată;

**F** domeniu listă - umple zona de memorie cu '*lista*' de octeți;  
**G** [= adresă [ adresă ... ] ] - lansează în execuție o secvență de program de la adresa de după =, până la una din adresele următoare;  
**H** valoare valoare - calculează suma și diferența hexazecimală a celor doi parametri;  
**I** valoare - citește de la portul de intrare specificat;  
**L** [ adresă [ disc : înregistrare înregistrare ] ] - încarcă un fișier sau înregistrări de pe disc;  
**M** domeniu adresă - mută conținutul unei zone de memorie la adresa specificată;  
**N** nume\_fisier [ nume\_fisier ] - specifică numele unui fișier ce va fi invocat de o comandă L sau W, ulterioară;  
**O** valoare octet - scrie octetul specificat la portul respectiv;  
**Q** - sfârșit sesiune de lucru Debug, revenire în MS-DOS;  
**R** [ nume\_registru ] - permite afișarea și modificarea registrelor;  
**S** domeniu listă - caută '*lista*' în domeniul specificat;  
**T** [= adresă ] [ valoare ] - execuție pas cu pas a unui program, cu afișarea registrelor, indicatorilor și a instrucțiunii ce urmează să se execute;  
**U** [ domeniu ] - dezassemblează zona de memorie specificată;  
**W** [ adresă [ disc : înregistrare înregistrare ] ] - scrie pe disc un fișier sau înregistrări;

Parametrii din aceste comenzi sunt despărțiți între ei de spații libere sau de virgule și au următoarea formă:

*disc*: indică discul de pe care se citește sau pe care se scrie: 0 = A: , 1 = B: , 2 = C: , 3 = D: ;

*octet*: valoare de tip octet (două cifre hexazecimale);

*înregistrare*: valoare de 1-3 cifre hexa care indică numărul înregistrării logice de pe disc și numărul de sectoare de pe disc ce vor fi scrise / citite. Înregistrările logice corespund sectoarelor;

*valoare*: o valoare hexa, de cel mult 4 cifre hexa;

*adresă*: reprezintă o destinație, formată din două părți, conținând fie numele unui registru de segment sau o adresă de segment de cel mult 4 cifre hexa, plus o valoare de deplasament (offset), despărțită de prima prin două puncte. Dacă se omite numele sau valoarea registrului segment se consideră implicit registrul segment DS pentru toate comenzile, cu excepția comenzilor G , L , T , U și W, pentru care se consideră registrul CS. De exemplu: **CS:0200**

*domeniu*: conține două adrese sau '*adresa L valoare*', în care *valoare* reprezintă numărul de linii pentru care se dă comanda. Exemplu :

**04BA:0300**  
**CS:100 1B00**  
**200:200 L 20**

Pentru cea de-a doua adresă se poate omite adresa de segment, întrucât se consideră egală cu prima adresă de segment, specificată.

*listă*: o serie de valori de tip octet sau un șir de caractere (string); de exemplu :

## f cs:100 42 45 52 54 41

Parantezele drepte [,] utilizate în descrierea comenzilor nu trebuie tastate, deoarece ele specifică doar faptul că parametrii respectivi sunt opționali; acolo unde aceste paranteze lipsesc înseamnă că parametrii respectivi sunt obligatorii. Dacă se introduce o comandă eronată (eroare de sintaxă), *Debug* nu o ignoră și afișează promptul - așteptând introducerea de noi comenzi, corecte.

### Sintaxa comenzilor Debug

#### A [ adresă ]

Permite editarea unui program în limbaj de asamblare, cu anumite restricții (nu se pot utiliza nume simbolice, deci toate adresele sunt absolute, iar toate constantele sunt specificate prin valori absolute, în hexazecimal, fără a fi urmate de litera **h** corespunzătoare acestei convenții din limbajele de asamblare). Editarea se va începe de la adresa specificată sau, în caz contrar, de la adresa *100h* în cadrul segmentului disponibil, cel mai de jos; pentru operatorii de conversie de tip: **word ptr**, **byte ptr** se pot utiliza formele prescurtate: **wo** și respectiv **by**.

La începutul fiecărei linii se afișează adresa curentă din memorie (*segment : offset*), și se așteaptă introducerea de instrucțiuni, în limbaj de asamblare, terminate cu **Enter (CR)**, după care se reia editarea cu o linie nouă. Pentru linia anterioară el generează codul mașină respectiv și indexează, în mod corespunzător, adresa următoarei instrucțiuni. Dacă trebuie să se specifice un câmp al unei instrucțiuni (ca în cazul unei instrucțiuni de salt la o adresă ulterioară), care nu este cunoscut în momentul introducerii liniei respective, se vor completa octeții corespunzători cu zero (sau altă valoare), iar după editarea celorlalte linii, când câmpurile respective devin cunoscute, se poate reedita (reasambla), integral, numai instrucțiunea respectivă, prin specificarea adresei instrucțiunii.

Dacă se introduce o instrucțiune greșită (eroare sintactică), editorul nu assemblează linia respectivă și afișează aceeași adresă pe o linie nouă, așteptând introducerea de noi instrucțiuni, corecte.

Revenirea în modul comandă, din modul de editare, se realizează tastând **Enter (CR)**, când se afișează pe o linie o nouă adresă, fără a mai edita nimic pe linia respectivă.

Dacă comanda nu este urmată de parametru '*adresa*', se va continua asamblarea (editarea) de la adresa la care rămăsese editarea anterioară.

Exemplu :

```
-A 5A2C:200 (CR)
5A2C:0200 _
```

#### C domeniu adresă

Compară zona de memorie specificată de '*domeniu*' cu o zonă, de aceeași dimensiune, începând de la adresa specificată. Dacă cele două zone de memorie sunt identice nu se afișează nimic, și se revine în modul comandă din *Debug*. Dacă există diferențe între cele două zone de memorie se afișează octeții diferiți și adresele corespunzătoare, din cele două zone, în formatul:

```
adresa_zona_1 octet_zona_1 octet_zona_2 adresa_zona_2
```

Exemplu :

```
-C 3AB2:100 200 300
```

- va compara conținutul zonei de memorie cuprinsă între adresele relative *100h* și *200h*, în segmentul care începe la adresa *3A2Bh*, cu conținutul unei zone de aceeași lungime care începe de la adresa *300h*, în același segment de memorie.

### **D [ domeniu ]**

Afișează conținutul zonei de memorie specificată prin '*domeniu*'. Dacă comanda nu este urmată de parametrul '*domeniu*', atunci se afișează 128 de octeți, începând de la afișarea anterioară. Afișarea se face în două câmpuri: hexazecimal și ASCII. Pentru caracterele netipăribile se afișează '!'. Se afișează pe fiecare linie adresa, conținutul hexa pentru 16 octeți începând de la adresa respectivă, separați în două grupe prin -, iar după aceea caracterele ASCII corespunzătoare codurilor hexa, respective. Exemplu :

```
-D 100 10f  
3AB2:0100 30 41 42 43 44 45 46 47 - 38 39 4a 4b 4c 4d 4e 4f      0ABCDEFGH9JKLMNO
```

### **E adresă [ listă ]**

Comanda permite introducerea în memorie a unor valori de tip octet, începând de la adresa specificată. Dacă '*lista*' nu este specificată se pot introduce valori octet cu octet. *Debug* afișează adresa specificată și vechiul conținut și așteaptă:

- introducerea unei noi valori, după care trece la adresa următoare, afișând din nou vechiul conținut al acesteia;
- dacă se tastează spațiu (blanc), atunci se trece la următoarea locație de memorie, fără modificarea celei curente;
- dacă se tastează - se revine la octetul anterior, pe care îl tipărește pe o linie nouă, precedat de adresa sa;
- dacă se tastează **Enter (CR)**, se încheie comanda **E**.

### **F domeniu listă**

Comanda permite umplerea unei zone de memorie cu valorile din '*lista*'. Dacă domeniul este mai mare decât lungimea listei, aceasta va fi depusă repetat în memorie până se umple domeniul specificat. Dacă lista este mai lungă, valorile suplimentare din listă sunt ignorate.

### **G [= adresă [ adresă adresă ... ] ]**

Această comandă permite execuția unui program; programul începe de la adresa specificată după semnul =, care face deosebirea între adresa de start și celelalte adrese care reprezintă puncte de *breakpoint* (de suspendare a execuției programului în momentul în care sunt atinse). În cazul în care se întâlnește o astfel de adresă, specificată în comandă, se întrerupe execuția programului și se afișează adresa la care s-a ajuns, instrucțiunea executată, conținutul registrelor și a indicatorilor. Se pot specifica până la 10 de astfel de adrese de *breakpoint*, dar numai la adrese ce conțin primul octet al unei instrucțiuni. De fapt la adresele de *breakpoint* se plasează codul întreruperii 3 (*INT 3*), care tratează întreruperea; în stivă trebuie să fie disponibili cel puțin 6 octeți, deoarece se salvează registrul indicatorilor, registrele segment și IP.

Lansarea în continuare a programului se poate face doar cu comanda **G**, fără nici un parametru. Se evită, de obicei, acest mod de lansare în execuție deoarece se poate pierde controlul asupra calculatorului, dacă programul nu este bine pus la punct.

### **H valoare valoare**

Comanda furnizează suma și diferența, fără transport (împrumut), a celor doi parametri.  
Exemplu :

**-H 1234 ABCD**  
**BE01 6667**

### **I valoare**

Citește și afișează un octet de la portul specificat (valoare).

### **L [ adresa [ disc înregistrare înregistrare ]]**

Comanda fără parametri încarcă în memorie, de la adresa *CS:0100*, un fișier al cărui nume a fost specificat într-o comandă *N*, anterioară; se va depune în *BX:CX* numărul de octeți încărcăți. Dacă se specifică adresa, fișierul va fi încărcat, în memorie, de la adresa specificată.

Dacă se utilizează comanda cu toți parametri se vor încărca, de pe disc, sectoare absolute în locul unui fișier. Unitatea de disc de la care se face citirea este specificată de câmpul '*disc*'. Se va încărca prima înregistrare specificată și se continuă cu următoarele, până când numărul de sectoare este egal cu cel de-al doilea parametru din comandă. De exemplu:

**-L04BA:100 2 0F 6D**

se vor încărca *109 (6Dh)* înregistrări, începând cu înregistrarea logică numărul *15 (0Fh)*, în memorie începând cu adresa *04BA:0100*. Parametrul adresă este ignorat pentru fișierele .exe. Dacă comanda *L* conține '*adresa*', *Debug* o va aduna la adresa găsită în fișierul .exe, pentru a determina adresa de start la care se încarcă fișierul.

### **M domeniu adresă**

Comanda permite mutarea unui bloc de memorie, specificat de '*domeniu*', începând de la adresa specificată. Mutarea suprapusă este fără pierdere de date, deoarece octetul de la adresa ce ar putea fi pierdut prin scriere peste el, este salvat primul. Adresele trebuie să fie în ordine crescătoare.

### **N nume\_fisier**

Această comandă realizează două funcții :

- asignează un nume de fișier pentru o comandă *L* sau *W*, ulterioară. Dacă s-a lansat *Debug* fără a se specifica numele unui fișier, atunci trebuie utilizată această comandă înainte de încărcarea sau scrierea unui fișier;
- asignează parametrii *nume\_fisier* la fișierul în depanare.

### **O valoare octet**

Transmite la portul de ieșire specificat (valoare), octetul.

**Q** Sfârșit sesiune de lucru *Debug*; revenire *MS-DOS*.

### **R [ nume\_registru ]**

Această comandă, fără nici un parametru, afișează conținutul tuturor registrelor și starea indicatorilor de condiții. Dacă comanda este urmată de numele unui registru, se afișează conținutul curent al acestuia și se poate modifica conținutul său.

Starea indicatorilor de condiții este afișată astfel :

<i>Indicator</i>	<i>Set(1)</i>	<i>Clear(0)</i>
<i>OF</i>	<i>OV</i>	<i>NV</i>
<i>DF</i>	<i>DN</i>	<i>UP</i>
<i>IF</i>	<i>EI</i>	<i>DI</i>
<i>SF</i>	<i>NG</i>	<i>PL</i>
<i>ZF</i>	<i>ZR</i>	<i>NZ</i>
<i>AF</i>	<i>AC</i>	<i>NA</i>
<i>PF</i>	<i>PE</i>	<i>PO</i>
<i>CF</i>	<i>CY</i>	<i>NC</i>

Indicatorii pot fi afișați separat, deci numai ei, prin comanda **RF**, și se pot modifica prin precizarea mnemonicii dorite a indicatorului respectiv.

### **S domeniul listă**

Se caută în '*domeniul*' specificat '*lista*' de octeți. Lista poate conține unul sau mai mulți octeți, separați prin spațiu sau virgulă. Comanda returnează adresele la care este găsită lista de octeți.

### **T [ = adresă ] [ valoare ]**

Această comandă execută instrucțiune cu instrucțiune începând de la adresa specificată după semnul =, și afișează conținutul tuturor registrelor, indicatorilor și următoarea instrucțiune de executat; dacă parametrul '*valoare*' lipsește, se va executa o singură instrucțiune, altfel se vor executa câte instrucțiuni specifică acest parametru (hexazecimal).

Dacă comanda nu are nici un parametru se execută o singură instrucțiune, de la adresa curentă din CS. De exemplu :

**-T=11A 10**

va executa 16 instrucțiuni începând de la adresa *CS:011A*.

### **U [ adresă ]**

Comanda dezassemblează octeți și afișează declarațiile sursă ce le corespund, împreună cu adresele și valorile octeților. Dacă comanda nu este urmată de parametru sunt dezasamblați 32 octeți, de la prima adresă după cea afișată de comanda *U* anterioară.

### **W [ adresă [ disc înregistrare înregistrare ] ]**

Scrie fișierul în depanare într-un fișier de pe disc, al cărui nume a fost precizat într-o comandă *N*, anterioară. În rest comanda este asemănătoare, din punct de vedere al parametrilor și al semnificației acestora, cu comanda *L*. În registrele *BX:CX* trebuie să se specifice numărul de octeți de transferat pe disc.

### **Mesaje de eroare**

**bf** = s-au tastat caractere greșite pentru indicatorii de condiții (*bad flag*);

**bp** = s-au specificat prea multe puncte de *breakpoint* într-o comandă *G* (*too many breakpoints*);

**br** = nume gresit de registru (*bad register*);

**df** = s-au specificat două valori diferite pentru același indicator (*double flag*).

## Turbo Debugger

Se poate utiliza **Turbo Debugger (TD)** pentru a depana orice program scris în C, Pascal sau limbaj de asamblare. El permite să se găsească eroarea și cauza acesteia; permite execuția parțială a programului (executând instrucțiune cu instrucțiune), astfel că se poate examina starea programului.

TD permite execuția pas cu pas, linie cu linie, puncte de întrerupere, vizualizarea variabilelor, registrelor, conținutul stivei, inspectarea, modificarea și urmărirea programului. De asemenea permite depanarea de programe orientate obiect Pascal și C++. În schimb TD nu permite:

- editarea programului; el nu conține un editor încorporat, și, deci nu permite modificarea codului sursă. Pentru modificarea acestuia se reeditează și recompilază;
- recompilarea programului; el nu poate recompila programul, care se realizează cu programul original.

### **Meniuri și casete de dialog**

TD are un meniu global convenabil accesibil, tot timpul, dintr-o 'bară' de meniuri, afișată în partea de sus a ecranului, prin intermediul unui cursor. Sistemul de meniuri are o organizare ierarhică; pentru fiecare meniu se deschide un submeniu (*pull-down menu*), care permite execuția unei comenzi, sau care permite deschiderea unui nou submeniu (*pop-up menu*), pentru acele elemente urmate de o iconiță meniu (R), sau a unei casete de dialog, pentru acele elemente urmate de o iconiță de dialog (...).

Sunt patru moduri în care se poate deschide un meniu :

- tasta **F10**, urmată de săgețile de deplasare, stânga respectiv dreapta, până la poziționarea pe meniul dorit, și apoi tasta **Enter (CR)**, va deschide meniul selectat;
- **F10**, și apoi prima literă a meniului dorit ( **spațiu, F, V, R, B, R, O, W, H** );
- **Alt** + prima literă a meniului dorit ( **spațiu, F, V, R, B, R, O, W, H** );
- cu mouse-ul se deplasează pe nivelul dorit și se apasă pe butonul din dreapta;

Pentru deplasarea de la un meniu la altul se utilizează săgețile direcționale (stânga, dreapta), iar pentru deplasarea în cadrul unui meniu specificat, de la o comandă la alta se utilizează săgețile de deplasare sus, jos. Pentru deplasarea la primul sau ultimul meniu se utilizează tastele 'Home' și 'End'.

Pentru ieșirea dintr-un submeniu și revenirea la meniul anterior se tastează 'Esc'; la fel se procedează ('Esc') și pentru revenirea dintr-un meniu la fereastra activă. De asemenea tastând **F10**, la orice nivel de meniu se revine la fereastra activă.

Unele comenzi ale meniului permit accesul rapid, prin tastarea unei anumite combinații de taste speciale (*hot key*), care apar, în meniu, în dreapta acestor comenzi. În figura următoare se prezintă meniul complet al **TD**.

Casetele de dialog pot conține unul sau mai multe elemente de tipul:

- *butoane*, reprezentate printr-un text umbrit (sau în video invers). Dacă se alege un buton **TD** realizează imediat acțiunea respectivă. Pentru a confirma alegerea făcută se selectează butonul 'OK' (tasta specială pentru 'ok' este **Alt-K**), iar pentru a se anula se selectează 'Cancel'. Casetele de dialog conțin și butonul 'Help', care oferă informații suplimentare despre ele.
- *casete de listare*, care conțin o listă de elemente dintre care se poate selecta.

Parcurgerea diferitelor casete de dialog se realizează cu tastele **Tab** (înainte) și **Shift-Tab** (înapoi). Pentru a modifica 'butoanele radio' se utilizează săgețile direcționale. Pentru a selecta un 'buton' se tastează 'Tab' și apoi 'Enter'.



*TD* este sensibil la context, adică el știe precis ce se caută și unde este cursorul când se alege o comandă. El utilizează această informație când răspunde la o comandă. De exemplu să considerăm o linie dintr-un program Pascal:

*variabila [ i ] := variabila [ i ] + 1;*

Dacă se tastează **Ctrl-I**, se deschide o fereastră de inspectare (*Inspector Window*). Când cursorul este pe '*variabila*', *TD* prezintă informații despre întregul vector (*variabila*). Dacă însă cursorul e poziționat (deci supraafișat) pe '*variabila [ i ]*', și apoi se tastează *Ctrl-I*, *TD* știe că se dorește inspectarea unui singur element și îl va afișa numai pe acesta. În acest mod se pot obține detalii din ce în ce mai fine asupra programului.

## **Meniul Turbo Debugger**

### **Meniuri**

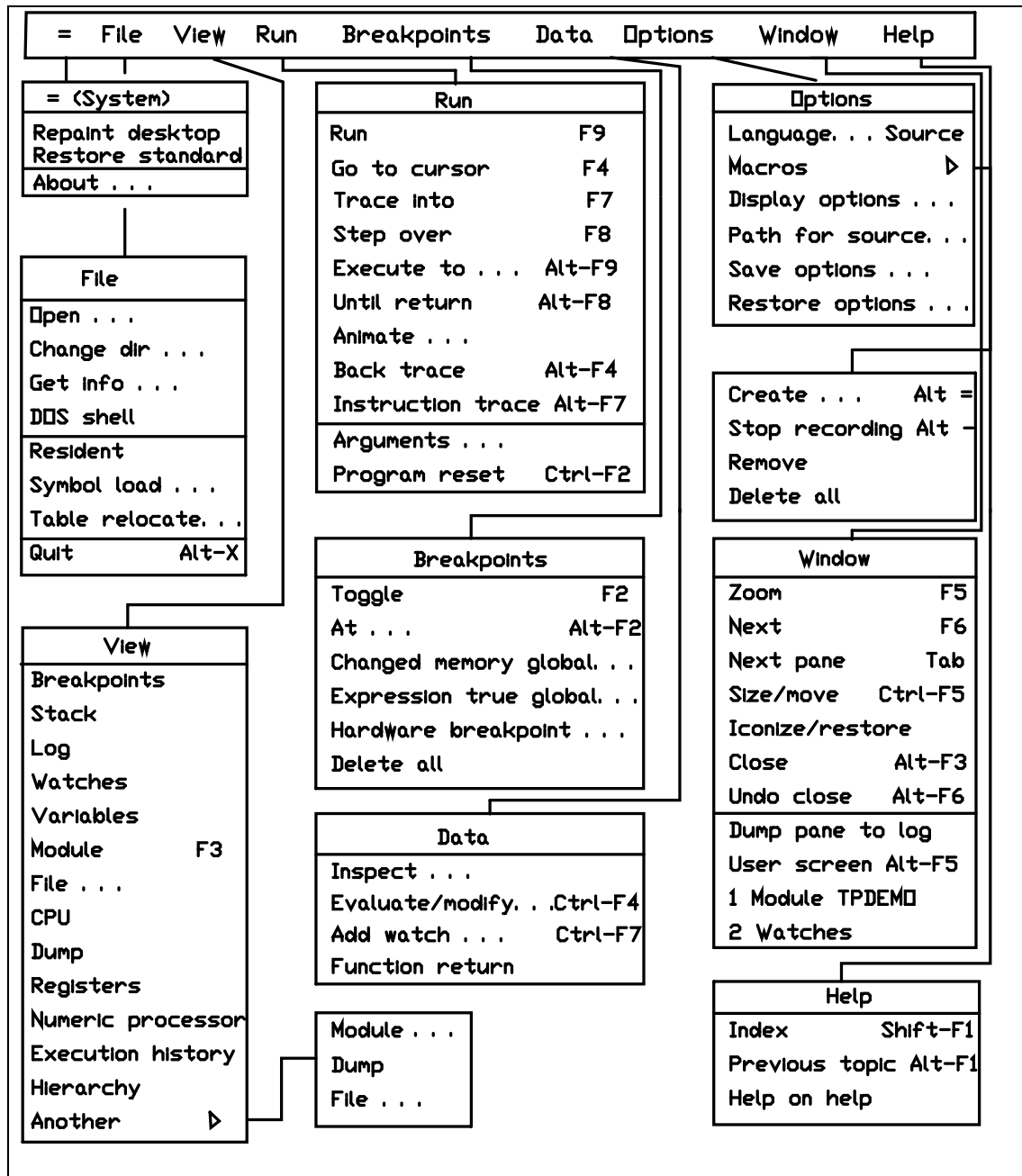
Un alt aspect al dependenței de context a *TD* este la utilizarea meniurilor locale, specifice diferitelor ferestre sau afișări în cadrul ferestrelor. Meniurile locale sunt dependente de fereastra sau afișarea particulară în care se află utilizatorul. Nu pot fi active simultan ambele tipuri de meniuri.

### **Meniurile globale**

- sunt accesibile prin tasta **F10** și săgețile (stânga, dreapta), sau prin tastarea primei litere a meniului ales din bara de meniuri;
- meniurile globale sunt, întotdeauna, accesibile din bara de meniuri (afișată în partea superioară a ecranului );
- conținutul lor este fix;
- unele dintre comenzile meniului au taste speciale, rapide, de activare din orice parte a *TD*.

### **Meniurile locale**

- un meniu local poate fi activat prin **Alt-F10** sau **Ctrl-F10**, sau prin apăsarea butonului drept al mouse-ului;
- plasarea și conținutul meniului depinde de fereastra sau afișajul (panoul) în care este utilizatorul, și unde este cursorul;
- conținutul poate varia de la un meniu la altul; rezultatele comenzilor cu nume asemănătoare, pot fi, totuși, diferite, depinzând de context;
- fiecare comandă dintr-un meniu local are o tastă specială de activare, rapidă, constând din **Ctrl + litera supraafișată a comenzii**.



O altă facilitate oferită de *TD* este aceea ca el pastrează o listă a istoriei comenzilor introduse în casetele de intrare și afișează textul introdus anterior într-o astfel de casetă, ori de câte ori este apelată. Se poate utiliza această listă, ca scurtătură, pentru a tasta, prin utilizarea săgeților și a tastei 'Enter', și a selecta orice intrare anterioară. Dacă se utilizează o intrare nemodificată din lista istoriei, această intrare e copiată în vârful listei. Se pot, de asemenea, edita intrările, prin deplasare cu săgețile pentru a insera cursorul în textul supraafișat, și apoi se editează textul, utilizând 'Del' și 'Backspace'. Lista istoriei afișează, de obicei, ultimele cinci răspunsuri; dimensiunea poate fi modificată utilizând programul *TDINST*.

TD mai permite specificarea incompletă a textului unei intrări, și completarea ei automată prin specificarea corectă a unei părți a numelui simbolic din program, și apoi se tastează **Alt-N**.

### **Construirea de Macro-uri**

Macro-urile sunt taste speciale definite de utilizator. Se poate asigura orice serie de comenzi TD și taste la o singură tastă, pentru reluarea lor ori de câte ori este cazul. Pentru a crea un macro se selectează **Options\Macros**. În acest moment sunt disponibile patru comenzi: **Create**, **Stop Recording**, **Remove**, **Delete all**, care permit crearea, terminarea, înlocuirea sau ștergerea de macro-uri. După introducerea comenzilor se tastează o tastă sau combinație de taste, dorite, la care vor fi asigurate acestea. Pentru a termina sesiunea de editare a unui macro, se poate alege între:

- **Options\Macros\Stop Recording**;
- tastați noua tastă sau combinație de taste, definite anterior;
- tastați **Alt-**.

### **Ferestrele TD**

#### **Meniul Sistem ( S )**

**Repaint Desktop** - reafixează întregul ecran;  
**Restore Standard** - reface amplasarea standard a ferestrei;  
**About** - afixează informații despre TD.

#### **Meniul fisier ( File )**

Acest meniul afixează conținutul unui fișier de pe disc sub forma unor linii de octeți hexa sau ca text ASCII. Se poate căuta un anumit text sau secvența de octeți. Comenzile acestui meniul sunt :

**Open** - deschide un nou program pentru depanare;  
**Change Dir** - modifică directorul sau discul curent;  
**Get Info** - afixează informații asupra programului;  
**DOS Shell** - revine în DOS pentru a primi comenzi DOS;  
**Resident** - termină TD și ramâne rezident;  
**Symbol Load** - încarcă tabela de simboluri, independent de fișierul .exe, pentru a o înlocui pe cea curentă;  
**Table Relocate** - setează segmentele de bază ale tabelii de simboluri;  
**Quit** - revenire în DOS, părăsire TD.

Comanda **File\Resident** se utilizează la depanarea unui program TSR (Terminate and Stay Resident): se încarcă programul TSR, cu tabela de simboluri, ca la un program normal. Apoi se rulează programul, se termină și ramâne rezident. Se poate seta orice *breakpoint*, și comanda pentru a face TD rezident. Pentru revenire în TD se utilizează tasta de 'întrerupere' (**Ctrl-Break**).

#### **Meniul View**

**Breakpoints** - afixează punctele de *breakpoint*;  
**Stack** - afixează stiva pentru apeluri de funcții;  
**Log** - afixează informații pentru evenimente și date;  
**Watches** - afixează variabilele care sunt monitorizate;  
**Variables** - afixează variabilele globale și locale;  
**Module** - afixează modulul programului sursă;

**File** - afișează fișier de pe disc (ASCII sau hexa);  
**CPU** - afișează instrucțiuni CPU, date, stiva;  
**Dump** - afișează date, pe linii;  
**Register** - afișează registrele generale și indicatorii CPU;  
**Numeric processor** - afișează dacă este procesor matematic în sistem, sau este emulat;  
**Execution History** - afișează codul de asamblare salvat, pentru reluare;

Următoarele comenzi sunt utilizate pentru depanarea programelor orientate obiect (OOP):

**Hierarchy** - afișează lista tipurilor de obiecte sau clase și ierarhia lor;  
**Another**  
**Module** - realizează o altă fereastră '*Module*';  
**Dump** - realizează o altă fereastră '*Dump*';  
**File** - realizează o altă fereastră '*File*';

### **Meniul Execuție program (Run)**

**Run** - execută programul încărcat, până la capăt, deci fără opriri;  
**Go To Cursor** - execută programul până la locația curentă a cursorului;  
**Trace Into** - execută o singură linie sursă sau instrucțiune, după cum este activă o fereastră '*Module*' sau '*CPU*';  
**Step Over** - execută o singură linie sursă sau instrucțiune, mai puțin apelurile de proceduri sau funcții (pe care le execută integral);  
**Execute To** - execută programul până la adresa specificată în caseta de dialog;  
**Until Return** - execută programul până când se întâlnește instrucțiunea de revenire din funcție sau procedură;  
**Animate** - realizează o serie de comenzi **Trace Into**, actualizând ecranul după fiecare comandă; permite rularea programului cu o viteză redusă. Se poate întrerupe prin tastarea oricărei taste;  
**Back Trace** - revine, pentru execuție, la o linie sursă sau instrucțiune anterioară;  
**Instruction Trace** - execută o singură instrucțiune;  
**Arguments** - permite introducerea de noi argumente în linia de comandă a programului;  
**Program Reset** - permite reîncărcarea, de pe disc, a programului curent, dar utilizatorul rămâne cu cursorul în poziția în care era la apelarea comenzii;

### **Meniul Breakpoints**

**Toggle** - permite starea (poziționarea) sau ștergerea unui *breakpoint* (punct de suspendare a execuției programului), la adresa curentă (supra-afișată) în fereastra '*Module*' sau '*CPU*';  
**At** - amplasarea unui *breakpoint* la adresa specificată, în caseta de dialog ce se deschide pentru această opțiune;  
**Changed Memory Global** - amplasează punct de *breakpoint*, global, care devine activ când o zonă de memorie modifică o valoare;  
**Expression True Global** - setează punct de *breakpoint*, care este activat când valoarea unei expresii, date de utilizator, devine adevărată;  
**Hardware breakpoint** - determină activarea unui *breakpoint* de un driver de dispozitiv (TDH386.SYS), asistat hardware (placa de depanare hardware);

**Delete All** - șterge toate punctele de *breakpoint*;

### **Meniul de date (Data)**

Acest meniu permite examinarea și modificarea datelor programului; se poate evalua o expresie, modifica valoarea unei variabile, și deschide ferestre '*Inspector*', pentru a afișa conținutul variabilelor.

**Inspect** - inspectează o dată obiect, și deschide o fereastră de inspectare, care afișează conținutul unei expresii sau variabile din program; se poate introduce numele unei variabile sau o expresie complexă.

**Evaluate/Modify** - deschide o casetă de dialog, care permite introducerea și evaluarea unei expresii;

**Add Watch** - permite adăugarea de variabile/expresii în fereastra '*Watches*';

**Function Return** - afișează într-o fereastră '*Inspector*' valoarea de returnat a funcției curente ;

### **Meniul Opțiuni (Options)**

Acest meniu permite setarea sau ajustarea unor parametrii care controlează operarea *TD*.

**Language** - permite precizarea limbajului sursă (C, Pascal sau Asamblare), pentru evaluarea expresiilor din caseta de dialog Evaluate/Modify;

**Macros** - permite definirea de macro-uri , de noi taste, sau ștergerea unora, deja asignate la o tastă (a fost descrisă anterior, la construirea de macro); conține următoarele comenzi :

**Create** - definește un macro, asociat la o tastă;

**Stop recording** - sfârșitul sesiunii de definire de macro;

**Removes** - șterge un macro;

**Delete All** - șterge toate macro-urile și restabilește semnificația inițială a tastelor;

**Display Options** - deschide o casetă de dialog, în care se pot poziționa câteva opțiuni, care comandă modul de afișare;

**Path for Source** - se precizează directoarele în care *TD* caută fișiere sursă;

**Save Options** - se deschide o casetă de dialog, din care se pot salva opțiunile curente (macro, formate de afișare, ferestre și toate setările făcute în meniul '*Options*') într-un fișier de configurație, pe disc;

**Restore Options** - reface opțiunile de pe disc, prin selectarea fișierului de configurație, dorit;

### **Meniul Window**

**Zoom** - permite modificarea dimensiunii ferestrei la întregul ecran, și invers;

**Next** - activează ferestrele următoare;

**Next Pane** - trece la următoarea casetă dintr-o fereastră;

**Size/Move** - mută sau modifică dimensiunile ferestrei;

**Iconize/Restore** - reduce fereastra la o iconiță (un simbol redus) sau o reface;

**Close** - închide fereastra;

**Undo Close** - redeschide ultima fereastră, care a fost închisă;

**Dump Pane to Log** - scrie caseta curentă în fereastra '*Log*';

**User Screen** - afișează 'ecranul utilizator' - adică afișările (ieșirile) programului;

**Open Window list** - afișează lista ferestrelor deschise, pentru activare;

**Window Pick** - afișează un meniu cu meniurile deschise, dacă sunt deschise mai mult de 9;

### **Meniul de informatii (Help)**

**Index** - acces la indexul de informații, pentru selectarea temei pentru care se doresc informații;

**Previous Topic** - se reafixează la ultimul ecran cu informații;

**Help on Help** - informații despre utilizarea acestui meni;

### **Lista Comenzilor (Taste speciale)**

F1 = informații despre dependența de context a TD;

F2 = **B**reakpoints**T**oggle;

F3 = **V**iew**M**odule;

F4 = **R**un**G**o to Cursor;

F5 = **W**indow**Z**oom;

F6 = **W**indow**N**ext Window;

F7 = **R**un**S**tep Over;

F8 = **R**un**R**un;

F10 = acces la bara meni;

Alt-F1 = **H**elp**P**revious Topic;

Alt-F2 = **B**reakpoints**A**t;

Alt-F3 = **W**indows**C**lose;

Alt-F4 = **R**un**B**ack Trace;

Alt-F5 = **W**indow**U**ser Screen;

Alt-F6 = **W**indow**U**ndo Close;

Alt-F7 = **R**un**I**nstruction Trace;

Alt-F8 = **R**un**U**ntil Return;

Alt-F9 = **R**un**E**xecute To;

Alt-F10 = meniul local al ferestrei;

Alt-1,9 = comutare între ferestrele numerotate de la 1 la 9;

Alt-spatiu = intra în meniul Sistem;

Alt-B = meniul **B**reakpoints;

Alt-D = meniul **D**ata;

Alt-F = meniul **F**ile;

Alt-H = meniul **H**elp;

Alt-O = meniul **O**ptions;

Alt-R = meniul **R**un;

Alt-V = meniul **V**iew;

Alt-W = meniul **W**indow;

Alt-X = meniul **F**ile**Q**uit - iesire di TD, revenire în DOS;

Alt-= = meniul **O**ptions**M**acros**C**reate;

Alt-- = meniul **O**ptions**M**acros**S**top Recording;

Ctrl-F2 = **R**un**P**rogram Reset - sfârșit sesiune de depanare și reinițializare, o nouă lansare;

Ctrl-F4 = **D**ata**E**valuate;

Ctrl-F5 = **W**indow**S**ize/Move;

Ctrl-F7 = **D**ata**A**dd Watch;

Ctrl-F8 = **B**reakpoints**T**oggle;

Ctrl-F9 = **Run**Run;  
Ctrl-F10 = meniul local al ferestrei;  
Ctrl ↑ = deplasează adresa de start, în caseta Code/ Data /Stack (CPU), cu un octet, în sus;  
Ctrl ↓ = deplasează adresa de start, în caseta Code/Data/Stack (CPU ), cu un octet, în jos;  
Ctrl-A = deplasare pe cuvântul anterior;  
Ctrl-C = defilare un ecran, în jos;  
Ctrl-D = deplasare cu o coloană;  
Ctrl-E = deplasare cu o linie, în sus;  
Ctrl-F = deplasare pe cuvântul următor;  
Ctrl-R = defilare un ecran, în sus;  
Ctrl-S = deplasare cu o coloană, la stânga;  
Ctrl-X = deplasare cu o linie, în jos;  
Shift-F1 = **Help**Index;  
Shift-Tab = deplasarea cursorului în caseta anterioară a ferestrei, sau în caseta de dialog;  
Shift ↑ = deplasarea cursorului între casetele  
Shift → = unei ferestre (caseta din direcția săgeții  
Shift ← = devine caseta activa );  
Esc = închide o fereastră '*Inspector*', sau ieșire din meniu;  
Ins = selecția unui bloc de text (ce va fi supra-afișat), împreună cu săgețile ↑, ↓ ;  
Tab = **Window**Next**Pane**;