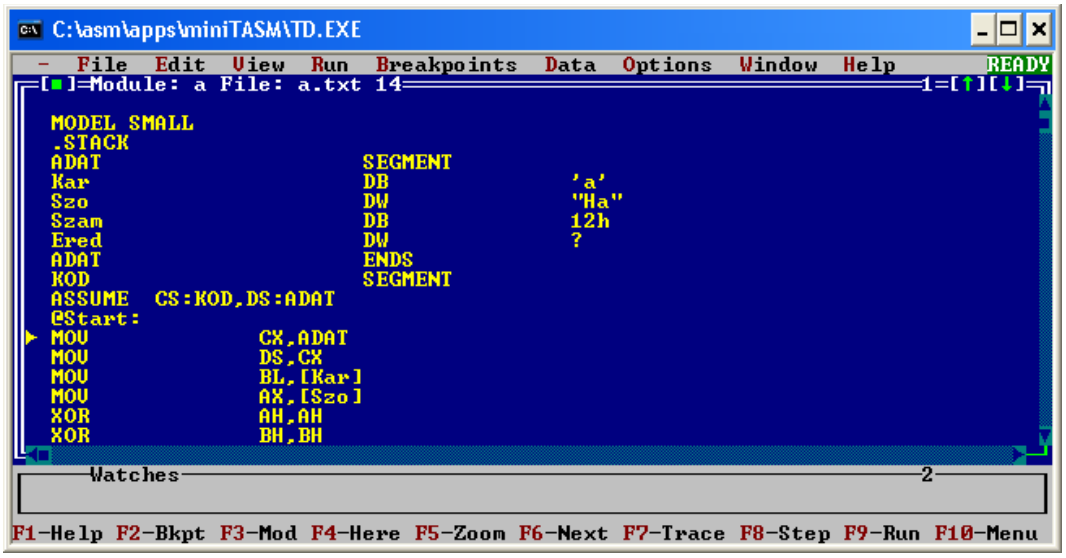
# A TURBO DEBUGGER HASZNÁLATA

Az eddig megírt programjaink működését nem tudtuk ellenőrizni, így elég kényelmetlen programozni, hogy nem látjuk, tényleg azt csinálja-e a program, amit elvárunk tőle. Ha valami rosszul működik, netán a számítógép semmire sem reagál (ezt úgy mondjuk, hogy "kiakadt" vagy "lefagyott"), a hiba megkeresése egyszerűen reménytelen feladat segítség nélkül. Ezt a természetes igényt elégítik ki a különböző debugger (nyomkövető, de szó szerint "bogártalanító ") szoftverek ill. hardverek. (Az elnevezés még a számítástechnika hőskorszakából származik, amikor is az egyik akkori számítógép működését valamilyen rovar zavarta meg. Azóta hívják a hibavadászatot "bogárirtásnak".)

A Turbo Assemblert és Linkert készítő Borland cég is kínál egy szoftveres nyomkövető eszközt, Turbo Debugger (TD) néven. Most ennek használatával fogunk megismerkedni.




A *Turbo Debugger* fő tulajdonsága, hogy képes egy futtatható állományt (.EXE vagy .COM kiterjesztéssel) betölteni, majd annak gépi kódú tartalmát Assembly forrásra visszafejteni. Ezt hívjuk ***disassemblálásnak*** (disassembly). A legszebb a dologban az, hogy a programban szereplő kód- és memóriahivatkozásokat konkrét számok helyett képes az eredeti forrásban szereplő szimbólumokkal megjeleníteni. Ezenkívül minden utasítást egyenként hajthatunk végre, ha akarjuk, többször is, sőt megadhatjuk, hogy a program végrehajtása egy bizonyos feltétel teljesülése esetén szakadjon meg. Figyelhetjük a memória tetszőleges területének tartalmát, a regiszterek és flag-ek értékeit, s mindezeket meg is változtathatjuk. A program futását az eredeti forrásokon is képes követni. Szóval csupa-csupa hasznos szolgáltatással bír, amikkel pont olyan kényelmesen figyelhetjük programunk tevékenységét, mintha mondjuk a Borland C fejlesztő rendszerében (IDE) dolgoznánk.

**Turbo Debugger-nek fordításra és szerkesztésre előkészítés**:

• minden forrást a /zi vagy /zd kapcsolóval lefordítani, formája

C:\>TASM/zi filenév.asm ⇒ eredmény filenév.obj vagy

C:\>TASM/zi filenév.txt ⇒ eredmény filenév.obj

**a** nevű állomány esetén:

Pl. ***c:\> TASM/zi a.asm***, ⇒ eredmény ***a.obj***

• a tárgykódokat a /v kapcsolóval összeszerkeszteni

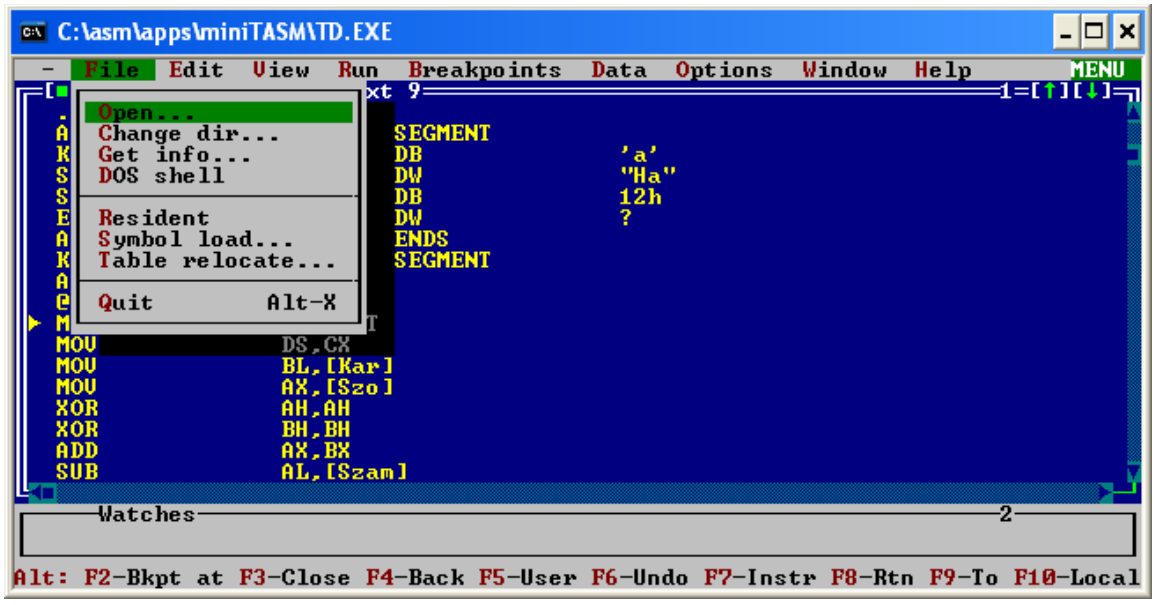
Pl**.*.c:\>TLINK/v a.obj*** ⇒ eredmény ***a.exe***

Ha ezt a két műveletet (*TASM/zi és TLINK/v*) elvégeztük, akkor a .EXE állományunk (remélhetőleg) tartalmazza az összes szimbolikus információt, amire a nyomkövetés során szükség lehet.

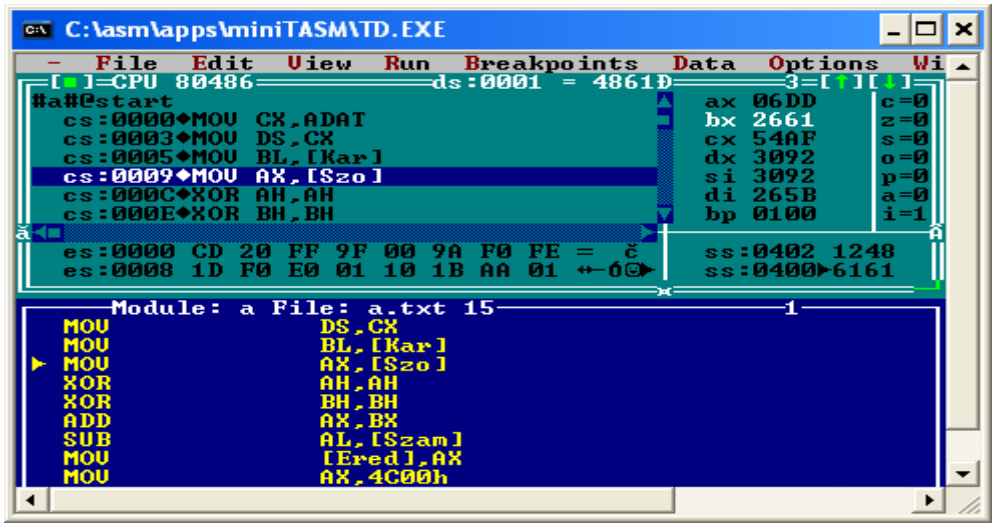
A dolognak két hátránya van: egyrészt .COM programokba nem kerülhet nyomkövetési info, másrészt ez az adathalmaz az .EXE fájl méretét igencsak megnövelheti (előfordulhat, hogy az eredeti többszöröse lesz a kimenet mérete).

A legelső példaprogramot (a.asm) begépeltük és elmentettük a.asm néven, majd lefordítottuk és linkeltük, az összes debug információt belerakva. A TD a.exe parancs kiadása utáni állapotot tükrözi a fenti kép.

A képernyő tetején ill. alján a már megszokott menüsor és státuszsor találhatók.

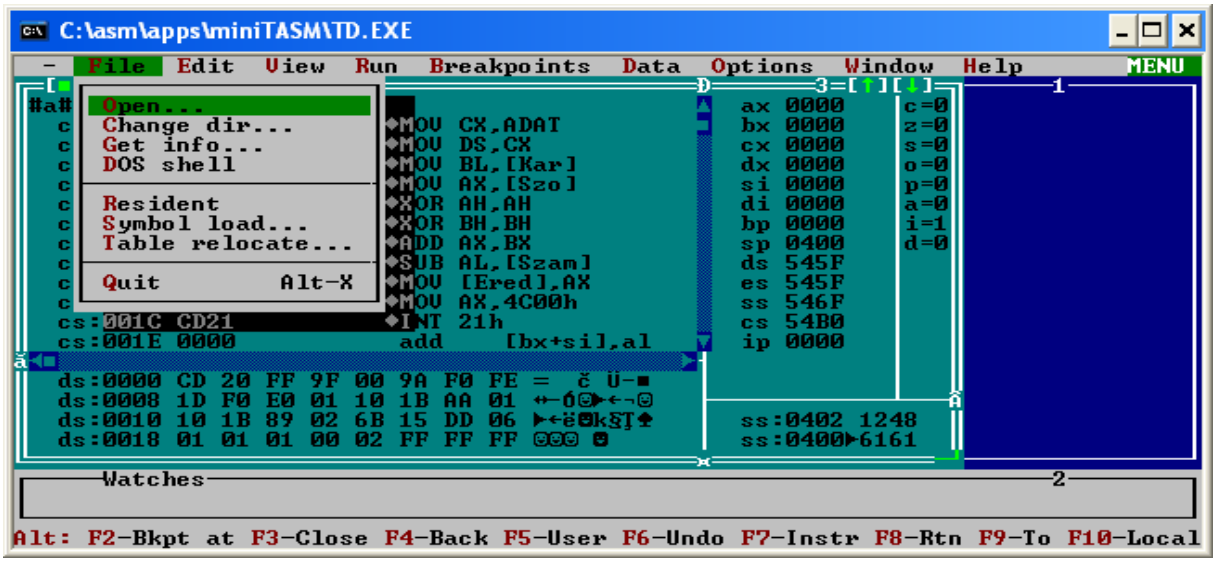



A kép nagy részét elfoglaló munkaasztalon (desktop) helyezkednek el a különféle célt szolgáló ablakok. Meghívása ***view*** menü ablak ***CPU*** sora.

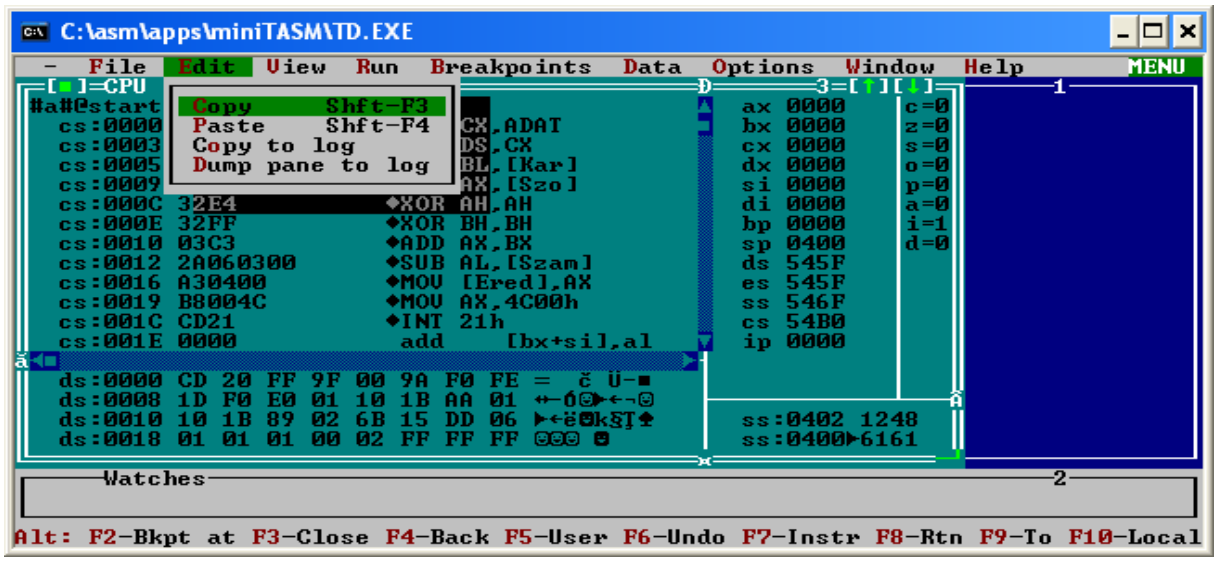


A legfelső ablak (CPU ablak) több részre osztható. A bal felső sarok az aktuális kód disassemblált változatát mutatja, és most éppen úgy van beállítva, hogy a forrás eredeti sorait is megjeleníti. A következő végrehajtandó sor a bal oldalon egy kis jobbra mutató háromszöggel van megjelölve (ez most a CS:0009h című sor).
Emellett az egyes regiszterek és flag-ek tartalma látható. Az előző állapothoz képest megváltozott dolgokat fehér színnel jelöli meg.
A jobb alsó sarok a memória egy adott területének tartalmát mutatja hexadecimális és ASCII alakban (ez az ú.n. memory dump)Végül a jobb alsó sarokban a verem aktuális állapotát szemléltetjük ***(ss:0400 ▸6161)***. A veremmutatót (azaz a verem tetejét) szintén egy jobbra mutató sötét háromszög jelzi.
A kép közepén levő nagyobb ablakban a forrásban követhetjük nyomon a program futását. Itt mindig azt a fájlt látjuk, amihez az éppen végrehajtott kód tartozik. A következő végrehajtandó sort a bal oldalon kis sárga háromszög mutatja.
Az alatta látható, *Stack* címkéjű ablak a különböző eljárás- és függvényhívások során a verembe rakott argumentumokat mutatja.
A legalsó, keskeny ablak a *Watches* címkét viseli. Ennek megfelelően az általunk óhajtott változók, memóriaterületek aktuális értékét követhetjük itt figyelemmel.
A felső sorban található menürendszeren kívül minden ablakban előhívható egy helyi menü (***local menu***) az **<ALT>+<F10>** billentyűkombinációval, ahol az aktuális ablakban (vagy részablakban) rendelkezésre álló plusz szolgáltatásokat érhetjük el. Ilyenek pl. az utasítás assemblálása, regiszter tartalmának megváltoztatása, flag törlése, megfigyelendő változó felvétele a Watches listába stb.
Az ablakok között az **<*F6*>** és **<*Shift*>+<*F6*>** billentyűkkel mozoghatunk, míg az aktuális ablakon belül a **<*Tab*>** és a **<*Shift*>+<*Tab*>** kombinációkkal lépkedhetünk a részablakok között.

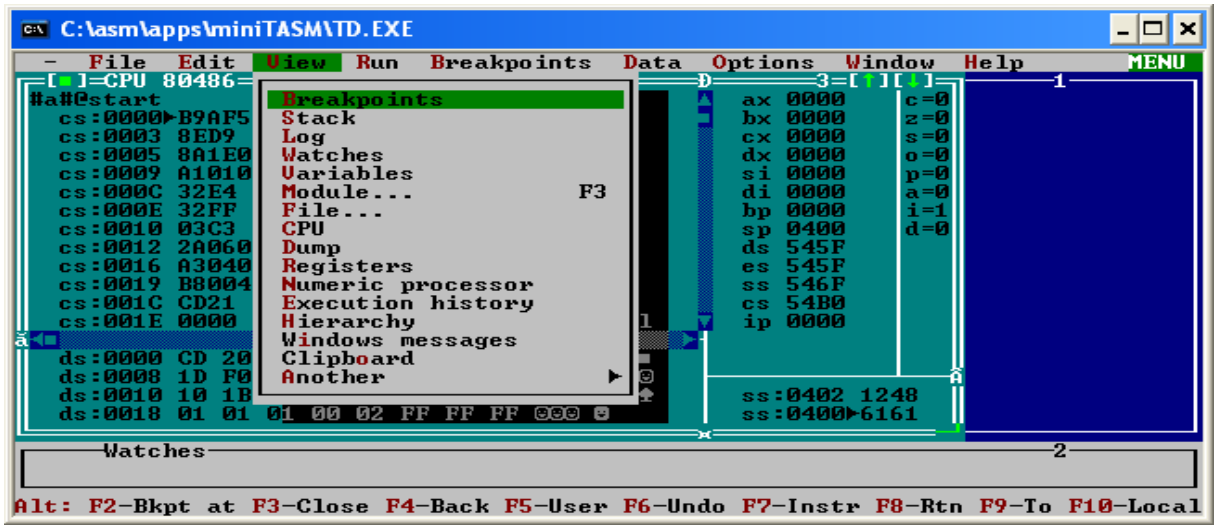
**Most áttekintjük az egyes menük funkcióit:**



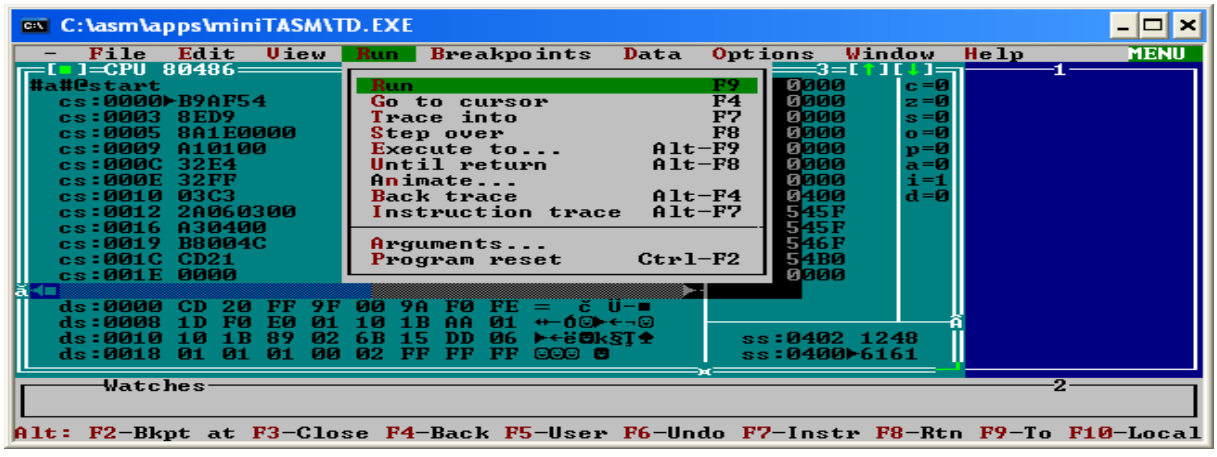

A ***File*** menü a szokásos állományműveleteket tartalmazza, de itt kaphatunk információt a betöltött programról is.




Az ***Edit*** menü szintén a gyakori másolás-beszúrás típusú funkciókat rejti.



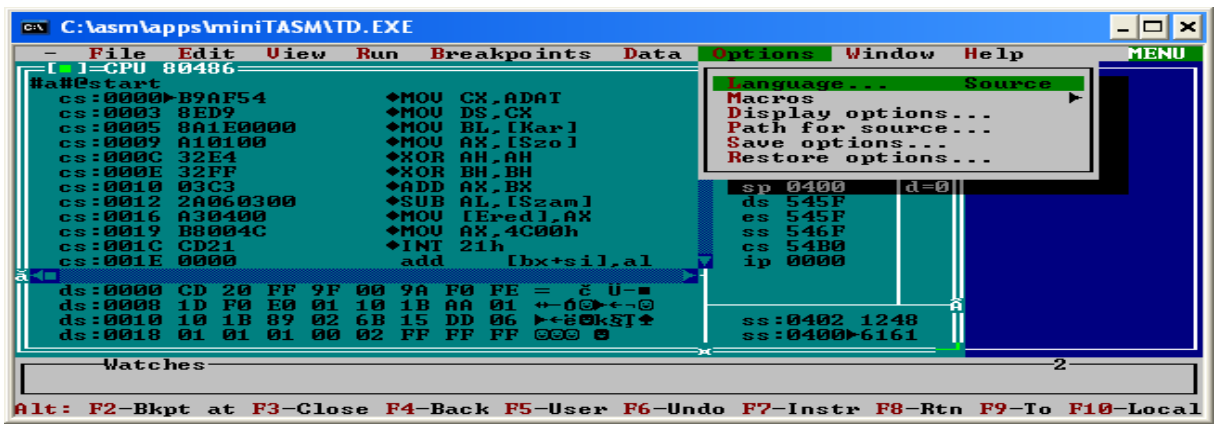

A ***View*** menü készlete igen gazdag. Innen nézhetjük meg a változókat (*Variables)*, a *CPU* ablakot, másik programmodult, tetszőleges állományt és még sok minden mást.



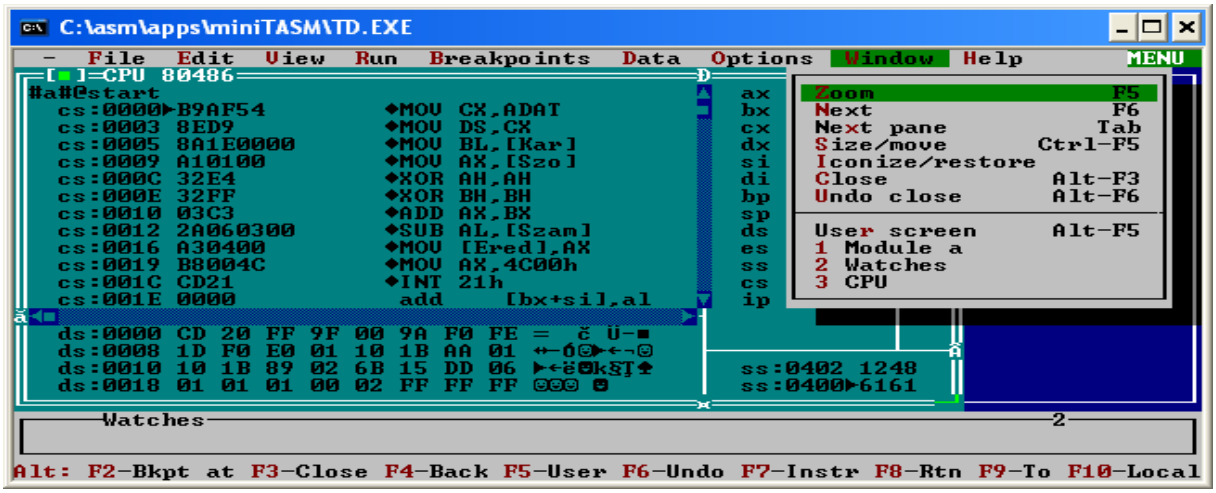
A ***Run*** menüben, mint neve is sejteti, a futtatással kapcsolatos tevékenységek vannak összegyűjtve, de itt állíthatók be a program futtatási (parancssoros) argumentumai is. Szinte az összes itt lévő szolgáltatáshoz tartozik valamilyen gyorsbillentyű (*hot key*) is. Néhány ezek közül: *futtatás <F9>, újraindítás <Ctrl>+<F2>, adott sorig végrehajtás <F4>, lépésenkénti végrehajtás <F7>, CALL és INT utasítások átugrása <F8>*.


A ***Breakpoints*** menüvel a töréspontokat tarthatjuk karban. A töréspont egy olyan hely a kódban, ahol a program végrehajtásának valamilyen feltétel teljesülése esetén (vagy mindenképpen) meg kell szakadnia. Ilyenkor a vezérlést ismét visszakapjuk a TD képernyőjével együtt, és kedvünk szerint beavatkozhatunk a program menetébe.


A ***Data*** menü az adatok, változók manipulálását segíti.



Az ***Options*** menüben találhatók a TD beállításai. Ilyenek pl. a forrás nyelve (Language), helye (Path for source), képernyővel kapcsolatos dolgok (Display options).




A ***Window*** menü az ablakokkal való megjelenítésre jó, a Help menü pedig a szokásos súgót tartalmazza.

A TD egyébként nemcsak Assembly, de Pascal és C nyelvű programot is képes nyomon követni, és az ezekben a nyelvekben meglevő összes adattípust is képes kezelni.