

IL TURBO DEBUGGER IN 10 MINUTI

(Brevissima introduzione al TD)

1. Come preparare i file

- File C e C++: `bcc -v nomefile.c`
- File ASM: `tasm /zi nomefile`
`tlink /v nomefile`
- File C e ASM: supponendo di avere un file in C che contiene il main (`pippo.c`) e un file in Assembler (`etabeta.asm`) e volendo ottenere un eseguibile (`pluto.exe`) con il modello di memoria small, è necessario digitare:

```
bcc -v -ms -epluto pippo.c etabeta.asm
```

Per i file C e C++ è possibile creare l'eseguibile dall'ambiente integrato e passare al TD direttamente, dal menu System (-). Accertarsi che sia contrassegnata la voce "Debug info in OBJs" in Options/Compiler/Advanced_code_generation e la voce "Source debugging" in Options/Debugger/ sia posta a "Standalone".

Per accedere al TD dalla linea di comando, digitare **td** (seguito, eventualmente, dal nome del file)

Una volta entrati nell'ambiente integrato del TD, caricare il file con la solita procedura (File/Open....). Il TD richiede un file .exe, ma utilizza anche il file con il codice sorgente.

2. View

Le finestre principali:

- **Breakpoints** - vengono visualizzati i punti di interruzione inseriti e le loro caratteristiche.
- **Stack** - visualizza le chiamate a procedura.
- **Watches** - visualizza la lista di variabili che si vogliono mantenere sotto osservazione.
- **Variables** - elenco di tutte le variabili del programma, con corrispondente indirizzo in memoria e valore corrente.
- **Module** - visualizza il codice sorgente.
- **CPU** - composta da 5 riquadri: code, registri, registro flag, stack, data. Il riquadro selezionato automaticamente è quello del codice; ci si può spostare agli altri riquadri premendo TAB.

Riquadro code: mostra il contenuto del code segment: istruzioni Assembler del programma caricato.

Premendo ALT+F10 (o premendo il tasto destro del mouse) si accede ad una finestra che presenta alcune funzioni per agire sul codice.

Per esempio:

- ◇ *Goto...*: permette di spostarsi all'istruzione di cui si inserisce l'indirizzo.
- ◇ *Assemble*: viene presentato un riquadro nel quale inserire una qualunque istruzione Assembler che sarà eseguita al posto dell'istruzione corrente.

Riquadro registri: mostra il contenuto dei registri. Premendo ALT+F10 (o premendo il tasto destro del mouse) si apre una finestra che presenta alcuni comandi coi quali è possibile modificare il valore del registro selezionato; per esempio si può incrementare, decrementare, azzerare o modificare direttamente.

ATTENZIONE: se è selezionato Options/Language/Source i valori numerici introdotti vengono considerati in esadecimale se il sorgente è in Assembler, in decimale se è in C. Se, invece, è selezionato Options/Language/Assembler i valori sono considerati in esadecimale; con Options/Language/C in decimale.

Riquadro registro flag: visualizza il contenuto del registro flag. Con ALT+F10 (o premendo il tasto destro del mouse) si accede alla finestra che permette di modificare il bit selezionato.

Riquadro Data: mostra il contenuto di un'area di memoria. E' automaticamente visualizzato il data segment (riferito da DS).

Premendo ALT+F10 (o premendo il tasto destro del mouse) si apre la finestra che permette alcune azioni. Le principali sono:

- ◇ *Goto...*: si posiziona all'indirizzo inserito. E' possibile specificare l'indirizzo di una variabile inserendo semplicemente il nome della stessa.
- ◇ *Change*: modifica il valore del byte su cui è posizionato il cursore.

Riquadro stack: analogo al precedente, ma riferito a SS. Nella posizione inferiore è visualizzato (indicato da un triangolino nero) l'elemento sulla cima dello stack. Sopra di esso vi è il penultimo elemento dello stack.

3. Run

Questo menu mette a disposizione le funzioni per l'esecuzione del programma (in varie modalità).

- **Run** - esecuzione del programma. Si arresta nei breakpoint o al termine del programma.
- **Go to the cursor** - esegue il programma fino alla posizione corrente del cursore.
- **Trace into (F7)** - esegue una singola riga sorgente (se la finestra attiva è Module), o un'istruzione Assembler (se la finestra attiva è CPU). Se la riga corrente contiene una chiamata a funzione, si entra in essa.
- **Step over (F8)** - simile alla precedente, ma non entra nelle funzioni (se un'istruzione contiene una chiamata ad una funzione, questa viene eseguita "in blocco").
- **Back trace** - ritorna indietro di un'istruzione. ATTENZIONE: affinché questa funzione abbia successo è necessario attivare l'opzione di Full history in questo modo: entrare in View/Execution history, premere ALT+F10, premere F per rendere attiva Execution history. Con il mouse: aprire la finestra di Full history, premere il tasto destro del mouse e premere il tasto sinistro in corrispondenza della voce Full history.
- **Arguments** - permette di inserire gli argomenti del main.
- **Program reset**: ricarica il programma e consente di ripartire dall'inizio.

4. Data

Funzioni relative alla visualizzazione e modifica delle variabili (tipi elementari, array, strutture, oggetti).

- **Inspect**: visualizza le caratteristiche di una variabile (indirizzo in memoria, tipo, valore). Premendo ALT+F10 (o premendo il tasto destro del mouse) si apre una finestra che fornisce alcuni comandi per agire sul dato; per esempio, è possibile modificarne il valore.
- **Evaluate/modify**: permette di valutare un'espressione (inserita nell'apposito riquadro) e di modificarne il valore.
- **Add watch**: aggiunge alla finestra Watches una variabile. Durante l'esecuzione il valore delle variabili di tale finestra viene aggiornato e mantenuto in osservazione.

5. Breakpoints

Un breakpoint è punto di interruzione del programma.

Il metodo più semplice per **inserire un breakpoint** consiste nel posizionare il cursore sulla riga in corrispondenza della quale il programma si deve arrestare (il programma si arresta prima di avere eseguito l'istruzione della riga) e usare il comando Toggle (Breakpoints/Toggle). Per **eliminare** il breakpoint seguire la stessa procedura.

E' possibile anche **inserire delle interruzioni globali**; per esempio, desideriamo che l'esecuzione si fermi se è verificata una certa condizione su un dato (p.e.: $a > 10$) o su un'espressione (p.e.: $x - y < 0$). Per inserire questi breakpoint, richiamare Breakpoints/Expression_True_Global... ed inserire nel riquadro l'espressione.

Per **eliminare tutti i breakpoint**: Breakpoints/Delete_all .

6. Gestione delle finestre

Portare in primo piano una finestra: premere ALT+w (per entrare nella finestra del menu Window) e selezionare la finestra digitando il numero corrispondente, o selezionandola col cursore.

Spostare una finestra: portare in primo piano la finestra, poi premere ALT+w, premere S (o selezionare il comando Size/move) e spostare la finestra con i tasti di spostamento del cursore. In alternativa, premere Cntrl+F5.

Con il mouse: posizionare l'indicatore sul lato sinistro o sul lato superiore della finestra ed eseguire lo spostamento.

Ridimensionare una finestra: seguire la procedura precedente per rendere attivo il comando Window/Size/move; tenendo premuto il tasto Shift, usare i tasti del cursore per ridimensionare la finestra.

Con il mouse: posizionare l'indicatore sul lato destro o sul lato inferiore della finestra ed eseguire il dimensionamento.

Chiudere una finestra: Window/Close. Con il mouse: premere il tasto sinistro in corrispondenza del quadratino posto nell'angolo in alto a sinistra della finestra.

7. Scorciatoie e uso del mouse

Per la maggior parte dei comandi, sono previste combinazioni di tasti per ottenere direttamente l'effetto del comando. Vedere le indicazioni a lato del comando citato nella finestra.

Tutti i comandi sono ottenibili anche con l'uso del mouse, premendo il tasto sinistro (come in un normale ambiente a finestre). Per aprire le finestre di modifica è necessario selezionare il riquadro desiderato e premere il tasto destro del mouse.