Il Debug di MsDos

Assembly con Debug

Il modo più immediato di scrivere programmi in Assembly per x-86 è tramite il programma di Sistema Operativo DEBUG.COM fornito su tutte le piattaforme Microsoft tramite la Shell di MsDos (invocabile a sua volta usando il comando cmd).
Con questo programma si scrivono programmi assembly x-86 in formato COM e con indirizzi assoluti numerici, imponendo una rigorosa disciplina nella scrittura delle istruzioni e mostrando realmente la pratica della programmazione x-86.
Inoltre, il fatto che DEBUG.COM sia sempre disponibile su ogni piattaforma Microsoft x-86 (ovvero da MsDos fino a Win32), ne garantisce la continuità didattica e la reperibilità immediata per gli studenti.

Formato degli eseguibili

MsDos riconosce come programmi eseguibili due tipi di formati che corrispondono a due tipi di files a loro volta distinti da una estensione caratteristica: files eseguibili di tipo ***COM*** (con estensione .COM) e files eseguibili di tipo ***EXE*** (con estensione .EXE). Inoltre la shell è in grado di eseguire in serie una lista di comandi (interni o esterni) scritti, riga per riga, su un file di testo denominato ***batch*** e di estensione ***BAT*** (.BAT).
Pur essendo eseguito**, un file .BAT non è un file eseguibile**, ma è semplicemente una sequenza di comandi scritti in forma testuale.
Gli eseguibili di tipo COM non possono essere più ampi di 64KB, mentre gli eseguibili di tipo EXE possono essere di qualsiasi dimensione, purchè inferiore alla dimensione della memoria convenzionale disponibile (tipicamente 640 KB).

L'eseguibile di tipo COM è molto semplice: contiene direttamente la lista di istruzioni assembly da eseguire, senza alcuna informazione ulteriore.
Quando la shell deve eseguire un file COM - su ordine dell'utente o perché presente in una riga di un file batch, decide a quale indirizzo di Memoria caricarlo – operazione detta di ***rilocazione statica*** (o caricamento rilocante statico) di MsDos.
Nei **primi 256 byte** (da 0 a FF_{16}), a partire da quell'indirizzo, scrive una serie di informazioni canoniche dette ***PSP*** (Program Segment Prefix), quindi legge il file .COM da disco byte per byte e lo ricopia nello stesso ordine in Memoria, subito dopo il PSP (quindi dall'indirizzo $\mathbf{100_{16}}$). Terminato il caricamento, MsDos cede il controllo alla prima istruzione del programma COM in memoria, impostando il Program Counter (registro IP) con l'indirizzo iniziale del programma. **Quando il programma terminerà**, dovrà preoccuparsi di **riconsegnare il controllo a MsDos**, affinche il calcolatore ritorni nella condizione di partenza.

Comandi

Debug è un **ambiente a carattere**, con un prompt (il trattino) che attende un comando dell'utente.
Tutti i numeri utilizzati con Debug sono sempre considerati in formato *esadecimale*; per usare una diversa rappresentazione va specificato il formato in coda al numero.
Debug **non è *case sensitive***, ovvero non distingue se il comando è scritto in maiuscolo o in minuscolo, comportandosi allo stesso modo in entrambi i casi.
I comandi dell'utente sono singoli caratteri, di facile comprensione. In evidenza, i comandi principali.

Alcuni dei comandi principali

- ? mostra i comandi del Debug

- q esce dal Debug chiudendo la finestra DOS

- r mostra il contenuto dei registri (register)

r	mostra tutti i registri
r nome_registro	mostra e consente, eventualmente, di variare il contenuto del registro specificato
r f	mostra e consente, eventualmente, di variare il contenuto dei singoli bit del registro dei flags

- a consente di scrivere istruzioni Assembly in memoria (assembla)

a	propone il primo indirizzo utile (per default l'offset vale :0100)
a indirizzo	consente di scrivere in memoria a partire dall'indirizzo specificato (associato al byte) considerato come offset all'interno del segmento indicato dal registro CS

- u mostra il contenuto della memoria interpretando il codice binario come istruzioni Assembly (disassembla)

u	mostra il contenuto a partire dal byte successivo all'ultimo comando u dato
u indirizzo	consente di mostrare le istruzioni a partire dal byte che si trova all'indirizzo specificato (visualizza il contenuto di circa 32 byte)
u ind1 ind2	consente di mostrare le istruzioni dall'indirizzo di partenza ind1 a quello finale ind2 specificati
u ind1 Ln	consente di mostrare le istruzioni dall'indirizzo di partenza ind1 per il numero n di byte specificato dopo L (n è interpretato in base 16)

- t consente di eseguire un'istruzione alla volta (trace)

t	esegue l'istruzione che si trova all'indirizzo di memoria contenuto nel registro IP
t =indirizzo	esegue l'istruzione che si trova all'indirizzo di memoria specificato

- e consente di mostrare e modificare il contenuto esadecimale della memoria byte per byte (enter)

e indirizzo	mostra un byte alla volta a partire da quello specificato dall'indirizzo e consente di digitare il nuovo valore esadecimale
e indirizzo elenco	a partire dall'indirizzo specificato vengono scritte in memoria le coppie di cifre esadecimali presenti nell'elenco (le coppie devono essere separate tra loro da uno spazio)

- d **mostra il contenuto in esadecimale della memoria, in righe di 16 byte alla volta (dump)**

d	**mostra 8 righe di 16 byte ciascuna a partire dal byte successivo all'ultimo comando d dato**
d indirizzo	**mostra 8 righe di 16 byte ciascuna a partire dal byte che si trova all'indirizzo di memoria specificato**
d ind1 ind2	**mostra il contenuto della memoria dall'indirizzo di partenza ind1 a quello finale ind2 specificati**
d ind1 Ln	**mostra il contenuto della memoria dall'indirizzo di partenza ind1 per il numero n di byte specificato dopo L (n è interpretato in base 16)**

ESEMPIO

Consultare i registri e il registro dei flags

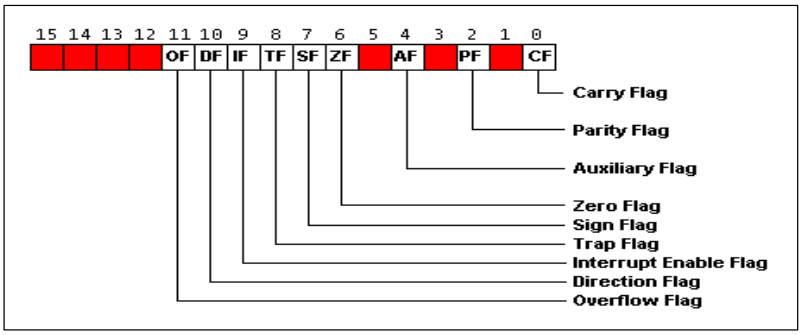
Con il **Comando R** del Debug consente di visualizzare il contenuto e lo stato dei registri x-86, nonché impostarne il valore.
Naturalmente i valori visualizzati sono espressi in esadecimale, mentre il registro dei Flags è mostrato attraverso delle sigle convenzionali.
Al prompt di Debug (il trattino) si può quindi usare **R registro**, specificando un registro se si vuole impostarne il valore. Infine, si può modificare un singolo bit del registro dei flag con il comando **r f**, specificando la lista dei nuovi valori.

C:\>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CED ES=0CED SS=0CED CS=0CED IP=0100 NV UP EI PL NZ NA PO NC
0CED:0100 27 DAA
-r cx
CX 0000
:12
-r
AX=0000 BX=0000 CX=0012 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CED ES=0CED SS=0CED CS=0CED IP=0100 NV UP EI PL NZ NA PO NC
0CED:0100 27 DAA
-r f
NV UP EI PL NZ NA PO NC -pe
-r f
NV UP EI PL NZ NA PE NC -zr po
-r
AX=0000 BX=0000 CX=0012 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CED ES=0CED SS=0CED CS=0CED IP=0100 NV UP EI PL ZR NA PO NC
0CED:0100 27 DAA
-q (per uscire dal debug)

Si nota che oltre ai registri, il comando R mostra sempre l'indirizzo, l'op. code e il codice mnemonico dell'istruzione puntata dal Program Counter (CS:IP).

Registro dei flags

Per quanto riguarda il registro Flags, vengono mostrati solo otto stati di otto bit significativi, con una coppia di caratteri.



La decodifica delle coppie di caratteri è la seguente:

Flag	Codice Debug	bit	Descrizione	Codice Debug	Bit	Descrizione
Overflow	OV	1	si	NV	0	no
Direzione	DN	1	decremento	UP	0	incremento
Interruzione	EI	1	abilitato	DI	0	disabilitato
Segno	NG	1	negativo	PL	0	positivo
Zero	ZR	1	si	NZ	0	no
Ausiliario	AC	1	si	NA	0	no
Parità	PE	1	pari	PO	0	dispari
Carry	CY	1	si	NC	0	no