

Parla più FORTH

di Raffaello De Masi
Sesta parte

Le strutture di controllo

Molte applicazioni del computer comprendono l'utilizzazione delle cosiddette strutture di controllo o strutture decisionali, per cui ordini diversi vengono eseguiti a seconda del valore assunto da una o più variabili o, più semplicemente, in base allo stato di vero o falso di alcuni registri (flag ovvero segnalatori). In effetti, nel caso di una struttura di controllo viene effettuata una comparazione tra due entità (confronto logico AND) e, in base al valore assunto dalla variabile booleana, vengono o meno eseguite determinate istruzioni.

Evidentemente, prima della decisione,

viene eseguita una comparazione, utilizzando un flag vero-falso presente in TOS. Le word che governano lo stato di tale flag (che rappresenta a tutti gli effetti il risultato di una sottrazione, eccetto per il fatto che produce un risultato qualitativo (vero o falso) invece di uno quantitativo) sono chiamate word di comparazione.

Semplici word di confronto

Alcune sono di comprensione immediata (si tenga conto che il confronto viene eseguito sempre con valori presenti in stack), come:

Come si vede esistono due gruppi di word: quelle che comparano il TOS con 0 e quelle che comparano i primi due valori dello Stack.

In ogni caso, comunque, i valori vengono effettivamente tolti dallo stack e sostituiti col valore del flag.

Per evitare ciò esiste appunto la word ?DUP che duplica il TOS solo se diverso da 0, consentendo, se necessario, la conservazione di tale valore se esso viene confrontato con lo 0 (word 0<, 0 = 0 0>). Simpatico, no?

La sequenza BEGIN-UNTIL

Il loop BEGIN-UNTIL è semplicemente un DO-LOOP nel quale il numero di ripetizioni non è specificato; esso ripeterà una sequenza continuamente finché si verificheranno condizioni predefinite.

La struttura BEGIN-UNTIL possiede due forme principali:

```
BEGIN ..... flag UNTIL
BEGIN ..... flag WHILE ..... REPEAT
```

dove, come al solito, flag rappresenta l'indicatore vero-falso in TOS (per curiosità il flag è tra l'altro il nome del volantino degli indicatori degli scambi ferroviari). La forma BEGIN-UNTIL ripete l'operazione rappresentata da finché il flag diviene pari ad 1. Al contrario, la forma BEGIN-WHILE-REPEAT esegue continuamente l'operazione tra BEGIN e REPEAT finché il flag diviene falso (nel qual caso il programma continua con la word seguente REPEAT).

Come appare evidente il loop BEGIN-UNTIL esegue sempre una operazione almeno una volta, poiché il suo flag non è testato finché (UNTIL) non si arriva alla fine della struttura. Al contrario un loop BEGIN-WHILE-REPEAT consente di bypassare l'operazione principale (quella compresa tra WHILE e REPEAT) senza eseguirla, poiché il flag è testato prima del-

0<	n --- flag	setta il flag ad 1 se n < 0
0=	n --- flag	idem se = 0
0>	n --- flag	idem se > 0
<	n1 n2 --- flag	setta il flag se n1 < n2
=	n1 n2 --- flag	i.c.s.
>	n1 n2 --- flag	i.c.s.
< >	n1 n2 --- flag	setta il flag se n1 <> n2
U<	un1 un2 --- flag	su numeri senza segno
D0=	d --- flag	*
D<	d1 d2 --- flag	*
D =	d1 d2 --- flag	*
DU<	ud1 ud2 --- flag	* come i rispettivi precedenti tranne che su numeri in doppia precisione.
NOT	flag 1 --- flag 2	Rovescia il valore del flag. Identico a 0=
?DUP (-DUP in Fig)	n --- n (n)	Duplica n se e' diverso da 0


```

: RAQU
  ( calcola la radice quadrata di un numero senza )
  ( segno (un) ,usando il metodo di Newton )
  ( un - - radice quadrata (un) )
  ( nella definizione si incontrano due nuove word )
  ( PICK (n1 - - n2) copia l'iesimo numero in TOS )
  ( U/MOD = /MOD tranne che si opera su un numero doppio )
  1
  ( si assume A = 1 )
  BEGIN
  OVER 0 3 PICK U/MOD      ( A1=(n/A+A) / 2 )
  SWAP DROP
  OVER + 2 U/MOD
  SWAP DROP
  SWAP OVER =             ( testa se A1 = A )
  UNTIL
  SWAP DROP              ( cancella il valore iniziale ; questo )
                        ( passaggio puo' essere abolito qualora )
                        ( si desideri conservare il valore di )
                        ( partenza )
                        ( mostra il risultato )
                        ( fine compilazione )
  ;

```

Figura 1 - Definizione della word RAQU che consente la radice quadrata di un numero col metodo di Newton.

```

:
  ( definisce la word prescelta )
  ( eleva il numero n1 alla potenza n2 . Se n2 )
  ( e' negativo il risultato e' dato pari a 0 )
  ( n1 n2 - - - n1^n2 )
  ( tale word e' gia' esistente in FORTH 79 )
  ( ed e' definita come ** )
  DUP 1 >
  IF
  OVER SWAP              ( n1 n2 - - - n1 n1 n2 )
                        ( valore di partenza del DO-LOOP = 1 )
  1
  DO OVER * LOOP        ( DO legge il valore finale del loop )
                        ( OVER riporta in TOS n1 )
                        ( viene eseguita la moltiplicazione )
                        ( a fine loop viene cancellato n1 )
  SWAP DROP
  ELSE
  ?DUP                  ( duplica il TOS solo se diverso da 0 )
  0=
  IF DROP 1             ( se n2=0 , il risultato e' 1 )
  ELSE 0<
  IF DROP 0             ( se n2<0 , il risultato e' 0 )
  THEN
  THEN
  THEN
  .
                        ( mostra il risultato )
                        ( fine definizione ) ;

```

Figura 2 - Elevamento di numero a potenza.

l'operazione, al momento del WHILE.

La struttura BEGIN-UNTIL è ideale per eseguire calcoli ripetitivi convergenti verso un risultato desiderato. Un esempio è il classico metodo di Newton per il calcolo della radice quadrata di un numero. Essenzialmente esso stabilisce che, se A è un'approssimazione della radice quadrata di n, allora

$$A1 = (n/A + A) / 2$$

(semisomma del rapporto numero/valore di A + A)

rappresenta una approssimazione migliore.

Se n = 625 ed il primo valore approssimato è 1

$$A1 = (625/1 + 1) / 2 = 313$$

$$A1 = (625/313 + 313) / 2 = 157.49$$

$$A1 = (625/157.49) / 2 = 80.72$$

$$A1 = (625/80.72 + 80.72) / 2 = 44.23$$

$$A1 = (625/44.23 + 44.23) / 2 = 29.18$$

$$A1 = (625/29.18 + 29.18) / 2 = 25.29$$

$$A1 = (625/25.29 + 25.29) / 2 = 25.00$$

$$A1 = (625/25 + 25) / 2 = 25$$

vale a dire che ogni successiva approssimazione si avvicina sempre di più al valore della radice quadrata.

Scrivere un programmino per calcolare la radice quadrata di un numero diviene, in FORTH, molto semplice. Immaginiamo di partire con un valore approssimato iniziale (il valore 1, come nel precedente esempio, è buono come un altro); useremo il loop BEGIN-UNTIL per ricercare le approssimazioni successive. Il solo problema è sapere quando fermarsi, vale a dire occorre scegliere una approssimazione tale che non sia troppo piccolo (per non rendere l'operazione eccessivamente lunga) per numeri grandi, né troppo grande per numeri piccoli. Si può risolvere tale nodo gordiano continuando il loop finché un calcolo produce lo stesso risultato di quello precedente, vale a dire che l'approssimazione rimane la stessa.

La figura 1 mostra la definizione della word RAQU (RADice QUadrata) che con-

sente di eseguire tale funzione col metodo di Newton. I numerosi OVER presenti nel programma rendono difficile da seguire lo sviluppo della word ma, per chi ha pazienza, basta verificare volta per volta lo stato dello stack per rendere il tutto molto più chiaro.

Prima del loop BEGIN-UNTIL, lo stack contiene il numero di base ed il primo valore di base approssimato, (un), assunto arbitrariamente eguale ad 1 (va bene, come già detto qualsiasi altro numero). La successiva approssimazione, A1, è ottenuta copiando un ed A in TOS (OVER 0 3 PICK), con lo 0 necessario per rendere un numero doppio. Viene eseguita la divisione (U/MOD) e cancellato il resto (SWAP DROP). Il secondo termine, A, è aggiunto al risultato (OVER +) e la somma mediata (0 2 U/MOD). Cancellato questo secondo resto con la sequenza SWAP DROP saranno presenti nello stack 3 valori (un, A ed A1) e siamo pronti a verificare se i due di testa, A ed A1, sono eguali.

Bisognerebbe, a questo punto, operare un confronto tra di essi, ma la word = li cancellerebbe dallo stack sostituendo ad essi il valore del flag condizionale. Bisognerebbe eseguire una duplicazione di A ed A1 ma, all'atto pratico, risulta più conveniente duplicare solo A1 in quanto dopo il confronto, la approssimazione precedente o è uguale ad A1 o è maggiore; ed in ambedue i casi A ci è inutile. La sequenza SWAP OVER = sposta A in TOS, riloca in TOS una copia di A1 e li sostituisce con un flag vero-falso.

Se A ed A1 hanno valori differenti, UNTIL cancella il flag e trasferisce il controllo alla linea seguente BEGIN. Altrimenti cancella il flag, scambia un ed A1 ponendo il valore iniziale di un in TOS, lo cancella e mostra il risultato.

Il FIG-FORTH contiene un'altra struttura, della forma BEGIN ... AGAIN, nella quale AGAIN forza il sistema operativo a ritornare a BEGIN ogni volta, incondi-

zionatamente. Questo loop è molto utile in quanto pone il programma in stallo, nel quale il processore aspetta un interrupt da qualche periferica (ricordate l'INKEYS o statement simili?).

La sequenza IF-THEN

L'altro tipo di struttura di controllo si presenta in una delle due forme seguenti:

- flag IF ... (vero) ... THEN
- flag IF ... (vero) ... ELSE ... (falso) ... THEN

In ambedue i casi IF estrae un flag dal TOS ed effettua una decisione in base al valore del flag stesso.

Se il flag vale 1 (vero), la struttura a) esegue le istruzioni (la o le word) tra IF e THEN mentre la b) eseguirà quelle tra IF ed ELSE. Se il flag vale 0 (falso) la struttura a) salta l'istruzione (word) mentre la b) eseguirà quella o quelle presenti tra ELSE e THEN, prima di continuare il programma.

Ad esempio la sequenza

```
DUP 10 > IF CR . " il numero in TOS è maggiore di 10" THEN
```

esegue un test sul TOS per accertare se esso è maggiore di 10 (Notare il DUP iniziale, necessario in quanto la word > preleva dal TOS un operando).

Allo stesso modo la sequenza

```
DUP 10 >
CR
IF . "Il numero è maggiore di 10"
ELSE . "Il numero è minore od eguale a 10"
THEN
```

esegue un test completo sul TOS nei riguardi del valore 10.

Una possibilità estremamente utile (non sempre presente in altri linguaggi ben più quotati), è quella di modificare le strutture condizionali. In tal caso, però, bisogna stare molto attenti agli operandi presenti in stack, tenendo conto che ogni operazione di IF preleva questo o questi dallo stack

stesso e li sostituisce col valore del flag.

L'esempio seguente è un miglioramento di quello precedente, consentendo di testare il valore del numero in TOS con 10 e di darne l'esatto risultato di confronto. Si noti il solito DUP iniziale, ed il DROP che cancella la copia del numero nel caso esso sia più grande di 10.

```
: COMPARAZIONECOL10
(compara il numero in TOS con 10, stampando
un messaggio di)
```

```
(risposta appropriato)
DUP 10 > CR
IF
. "Il numero è maggiore di 10"
DROP
ELSE
10 =
IF
. "Il numero è eguale a 10"
ELSE
. "Il numero è minore di 10"
THEN
THEN
(fine definizione) ;
```

Il valore in TOS viene cancellato. Nel caso lo si voglia conservare, basta eliminare il DROP ed aggiungere un DUP prima del secondo IF.

Un utile esempio di utilizzo delle strutture IF-THEN integrata da quella DO-LOOP è mostrato nella figura 2. In essa viene definita la word ^ che consente l'elevamento a potenza di un numero. È previsto anche il caso che l'esponente sia negati-

```
: .S ( elenca il contenuto dello Stack , senza alterarlo )
( - - - )
DEPTH 2 * ( duplica il numero di byte occupati in stack )
?DUP 0= ( controlla che lo stack non sia vuoto )
IF ( nel qual caso )
CR ."STACK VUOTO" ( lo indica )
ELSE
SP@ 2+ ( inizio del DO-LOOP = puntatore stack originale )
DUP ROT + ( fine del DO-LOOP = puntatore + bytes )
SWAP ( valore di partenza , valore finale - - - )
( valore finale , valore di partenza )
DO
CR I ? ( mostra il numero successivo )
2
+LOOP
THEN
( fine definizione )
;
```

Figura 3a - Definizione della word .S in FORTH 79.


```
: .S ( elenca il contenuto dello stack , senza alterarlo )
( - - - )
SP@ SO @ - ( numero di byte occupati = SP@ - (SO) )
-DUP 0= ( controlla che lo stack non sia vuoto )
IF
CR ."STACK VUOTO"
ELSE
SO @ ( limite di DO-LOOP = (SO) )
DUP ROT +,
DO
CR I ? ( mostra il numero successivo )
2
+LOOP
THEN
( fine definizione )
;
```

Figura 3b - idem ma in FIG-FORTH.

vo nel qual caso (ricordate le regole e le limitazioni della notazione in virgola fissa) il risultato è dato pari a 0.

Come utile applicazione, viene esemplificato nell'esempio pubblicato qui sopra (figg. 3a e 3b) l'uso di una word .S, estremamente utile che, sebbene non menzionata nei manuali del FIG FORTH e del FORTH-79, è presente in molti pacchetti FORTH. Essa mostra il contenuto dell'intero stack dall'alto in basso senza in alcun modo alterarlo, per cui appare evidente la sua utilità in fase di debug.

Data la sua grande utilità, ne vengono fornite le versioni sia in FORTH-79 che in FIG-FORTH. La prima si basa sulla word DEPTH (che lascia in TOS il numero di valori presenti nello stack), la seconda sulla variabile SO (che già conosciamo e che contiene il valore iniziale del puntatore dello stack) mentre ambedue utilizzano la word SPC, che consente di conoscere l'indirizzo iniziale dello stack. Nella versione FIG compare il -DUP identico al ?DUP (duplica il TOS solo se diverso da 0) e la @ che ricerca il numero all'indirizzo mentre la word I inserisce il valore corrente dell'indice di DO-LOOP nello stack.

Cari forthisti, ormai abbiamo a disposizione tanti di quei mezzi da poter abbandonare la Panda del Basic per passare alla Kawasaki del Forth. Forse non sarà altrettanto comoda, e non perdonerò gli errori in curva, ma nessuno vi aveva promesso rose e fiori, ma solo sgommate in partenza... Comunque consolatevi, il peggio è passato! Tenete conto che, comunque, il Forth abbisogna, al contrario di altri linguaggi più intuitivi e, diciamo, discorsivi, di un continuo esercizio, per cui carta e penna, e cominciate a fare allenamento (ed attenti alla manetta del gas). 

Su richiesta del Forth Interest Group Italia, pubblichiamo il seguente comunicato

Alcune note sul F.I.G. Italia

Dopo quindici anni dalla creazione di questo splendido linguaggio, le persone che utilizzano il FORTH in tutto il mondo sono diventate decine di migliaia. Il FORTH riscuote ovunque un notevole successo divenendo spesso il linguaggio preferito di hobbisti e professionisti operanti nei più svariati settori. Persino una delegazione di soci F.I.G. in visita nella Repubblica Popolare Cinese ha avuto una accoglienza entusiasta nelle locali Università.

La Associazione che riunisce gran parte dei programmatori FORTH, il FORTH Interest Group, ha conosciuto dei tassi di crescita del 100% sin dall'anno della sua formazione nel 1978. Attualmente è presente in quasi tutti gli stati degli U.S.A. con nuclei attivi e in quasi tutti i paesi del mondo mediante Associazioni Nazionali.

In Italia è presente da circa un anno il FORTH Interest Group Italia, una Associazione senza fini di lucro avente come fine lo scambio di informazioni ed esperienze tra utilizzatori del FORTH che conta ormai un centinaio di soci.

Il 1 di giugno si è aperto il suo nuovo anno sociale con molte iniziative in programma per tutti i soci:

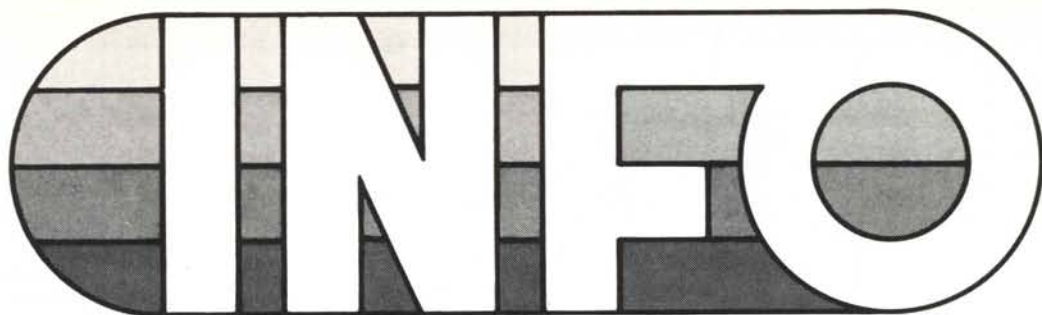
- distribuzione della rivista bimestrale di oltre 40 pagine FORTH DIMENSIONS edita dal F.I.G. unitamente ad un inserto in lingua italiana, contenente applicazioni particolari, routine, articoli divulgativi, realizzato grazie al contributo dei soci.
 - Distribuzione di pubblicazioni inerenti il FORTH, compresi i listati sorgente del linguaggio per quasi tutte le CPU più usate a prezzi particolarmente vantaggiosi.
 - Realizzazione di un corso pilota presso la Biblioteca Comunale di Novate Milanese con possibilità di usare cinque macchine FORTH-dotate.
 - Incontri informativi presso il Politecnico di Milano, Istituto di Elettronica.
 - Riunioni regolari per scambi e dibattiti a Milano e in varie altre città.
- L'iscrizione al FORTH Interest Group Italia dura un anno a partire da giugno e la quota associativa è di Lit. 40.000.

Per iscriversi basta spedire una richiesta di iscrizione scegliendo per il pagamento una delle seguenti forme:

- assegno bancario non trasferibile intestato a FORTH Interest Group Italia
- bonifico bancario a favore del c/c n. 11705/1 presso la CARIPLO - agenzia 33 - viale Lunigiana 22 - 20125 Milano, intestato a: FORTH Interest Group Italia.

Per informazioni ed iscrizioni l'indirizzo della Associazione è:
FORTH INTEREST GROUP ITALIA - Via G. Forni 48 - 20161 Milano

OGGI C'E'



E' UN MARCHIO INFOTEL

AL COMPLETO SERVIZIO DEI RIVENDITORI

agente esclusivo per il Lazio:



◦ personal computer a 16 bit **APRICOT, VICTOR**



- floppy **MAXELL**
- stampanti ad aghi **MITSUI**
- stampanti low cost **CP/JP-80**
- stampanti a margherita **JUKI**
- accoppiatori acustici **NOVATION CAT, ANDERSON-JACOBSON**
- plotter **YEW, ENTER C**
- digitizer **GTCO**
- mouse **MOUSE SYSTEM**

agente esclusivo per Lazio e Umbria:



◦ software **J.soft, MICROSOFT, HABADDEX, FILEVISION....**



**GRUPPO
EDITORIALE
JACKSON**

◦ *tutti i libri della casa editrice*