

Debugger

Beschreibung

Der Debugger erlaubt dem Nutzer, die Schritte einer High-Level- Definition darzustellen.

Er wird in der Form 'DEBUG name' eingeschaltet. Dabei ist 'name' das Dictionary-Wort, welches verfolgt werden soll. Bei der Ausfuehrung des Wortes werden die einzelnen Schritte und der Parameterstack auf dem Bildschirm angezeigt.

Sonderbefehle waehrend der Anzeige:

<F> ermoeglicht die zusaetzliche Eingabe von Forth-Befehlen und deren Interpretation, bis mit RESUME die Bearbeitung fortgesetzt wird

<C> schaltet auf kontinuierlichen Lauf um

<Q> bricht die Bearbeitung des Wortes ab und stellt dessen Standardabarbeitung wieder ein

Hinweis: das Paket ist unabhangig von der Lage des Forth83 im Speicher; wird das Forth83 auf einen anderen Adressbereich gelinkt, braucht das Paket nicht angepasst zu werden!

GLOSSAR

BUG (-) Forth

macht das Vokabular BUG zum ersten durchsuchten

(DEBUG) (adr1 adr2-) Bug

setzt die obere und die untere Grenze zu dem gegebenen Wert und patcht NEXT (adr1 niedere und adr2 obere Adr)

DEBUG (name-) Forth

schaltet den Bearbeitungsvorgang fuer das Wort 'name' ein. Patcht NEXT zur debuggierten Version von NEXT und setzt die Grenzen.

RES (-adr) Bug

Variable- falls wahr, Debugarbeit wird wieder aufgenommen

RESUME (-) Bug

schaltet RES ein, und setzt damit die Niederschrift fort.

SLOW (-adr) Bug

Variable- falls wahr, kontinuierlicher Betrieb

TRACE (ip-) Bug

zeigt den Inhalt des Parameterstacks und den Namen des naechsten Wortes, welches zur Ausfuehrung vorgesehen ist. Es wartet auf Tastenbetaetigung, <ET> falls SLOW nicht wahr ist

'UNNEST (pfa-'pfa) Bug

sucht das Ende des bearbeiteten Wortes und uebergibt diese pfa

Bedienung des Debuggers

Für die Programmtestung ist der Debugger ein unentbehrliches Hilfsmittel und jeder, der sich ernsthaft mit Programmierung in FORTH beschäftigt, sollte sich mit diesem Debugger vertraut machen.

Nach dem F83-Standard haben wir die folgenden Worte zur Verfügung:

DEBUG und RESUME.

Nach Aufruf des Vokabulars BUG auch

UNBUG, SLOW ON, SLOW OFF.

Mit DEBUG 'name' wird ein Wort zum Test eingerichtet, wobei dabei noch nichts sichtbares passiert. Wird jedoch das Wort 'name' direkt oder indirekt, das heißt wenn das Wort Bestandteil eines weiteren Wortes ist, gerufen, wird das Wort im Einzel-Schritt-Betrieb abgearbeitet. Zu Beginn der Single-Step-Abarbeitung steht der aktuelle Stackinhalt sowie auf der nächsten Zeile das nächste Wort aus dem Code-Body. Nach Betätigen einer beliebigen Taste erscheint der neue Stackeintrag sowie das folgende Wort.

Am Beispiel von COPY möchte ich das verdeutlichen:

Wir geben

DEBUG COPY /ret/ 1 2 COPY /ret/ ein:

```

      1      2
2      --
      1      2      2
?ENOUGH  --
      1      2
SWAP      --
      2      1
BLOCK      --
      2      32256
DROP      --
      2
BLK      --
      2      4637
empty
UPDATE    --
empty
SAVE-BUFFERS --
empty
UNNEST    -- ok
```

Während des Debug-Betriebes ist ein Abbruch mit 'Q' erreichbar. Es erscheint die Ausschrift „unbug“.

Mit der Taste 'C' ist ein Ausschalten des Single-Step-Modes möglich. Der Debug-Betrieb bleibt erhalten, aber der Rest des Wortes wird hintereinander abgearbeitet bis zu UNNEST bzw. bis zu einer beliebigen Tastenbetätigung. 'F' bewirkt den vorübergehenden Abbruch des Debug-Betriebes. Mit Hilfe des Forth-Systems kann eine weitere Untersuchung erfolgen.

So kann man z. B. falsche Werte auf dem Parameterstack ändern.

Eine andere Möglichkeit ist, mit DEBUG 'neuname' die Untersuchung eines Bestandteils der Definition zu erreichen (in unserem Beispiel etwa BLOCK oder SAVE-BUFFERS).

RESUME setzt dann den Debug-Betrieb dort fort, wo er mit 'F' unterbrochen wurde.

Nach Aufruf von dem Vokabular BUG können auch die folgenden Worte genutzt werden, welche nur nach Erreichen des Wort-Endes wirksam sind.

Mit UNBUG wird der Debug-Betrieb nach Erreichen des Wort-Endes ausgeschaltet, wie mit 'Q' während der Abarbeitung des Wortes.

SLOW ON bedeutet einen kontinuierlichen Step-Betrieb, wie 'C' während der Abarbeitung. SLOW OFF ist das Gegenteil und bewirkt den Step-Betrieb.

Ein Experimentieren mit dem Debugger lohnt sich auf jeden Fall und es wird sich sehr bald die Unentbehrlichkeit dieses Werkzeuges beweisen.

(Michael Scholz)

internes

BC ist der instruction pointer (IP) und IY der return stack pointer (RP).

BUG: Das Vokabular, das die Trace-Wörter auf höherer Ebene enthält.

FNEXT: Eine Maschinensprachen-Subroutine, die NEXT wieder in den ursprünglichen Zustand zurückversetzt.

DNEXT: Eine Kopie von next, die anstelle der normalen ausgeführt wird.

DEBNEXT: Die Debugger-Version von next.

Wenn die IP zwischen <IP und IP> liegt, wird der Inhalt der Ausführungsvariable 'DEBUG ausgeführt. Zuerst wird die IP auf den Parameterstapel gelegt. Das Wort, auf das 'DEBUG zeigt, kann ein beliebiges Wort auf höherer oder niedriger Ebene sein, solange die zuvor abgelegte IP verworfen wird. Es muss durch den Aufruf von PNEXT beendet werden, um next erneut zu patchen und so die Trace-Funktion weiter auszuführen.

PNEXT patcht Forth's Next, um zu DEBNEXT zu springen. Dies versetzt uns in den DEBUG-Modus und ermöglicht die Trace-Funktion.

```
VARIABLE 'DEBUG    5517
VARIABLE <IP        5521
VARIABLE IP>        552b
VARIABLE CNT        5535
```

FNEXT: (restauriert orig. NEXT)

```

5541 21 22 04      LD      HL,>NEXT
5544 36 0A         LD      (HL),0AH
5546 23           INC      HL
5547 36 03         LD      (HL),03H
5549 23           INC      HL
554A 36 6F         LD      (HL),6FH
554C C9           RET
---- orig. >NEXT beginnt mit folgenden drei Bytes
0422 0A         LD      A,(BC)
0423 03         INC      BC
0424 6F         LD      L,A

DNEXT:
5557 0A         LD      A,(BC)
5558 03         INC      BC
5559 6F         LD      L,A
555A 0A         LD      A,(BC)
555B 03         INC      BC
555C 67         LD      H,A      ; HL := (IP)
555D 5E         LD      E,(HL)
555E 23         INC      HL
555F 56         LD      D,(HL)
5560 EB         EX      DE,HL    : DE = ((IP))
5561 E9         JP      (HL)

DEBNEXT:
556E 2A 21 55     LD      HL,(<IP)
5571 B7          OR      A
5572 ED 42       SBC      HL,BC
5574 30 23       JR      NC,5599H
5576 2A 2B 55     LD      HL,(IP>)
5579 B7          OR      A
557A ED 42       SBC      HL,BC
557C 38 1B       JR      C,5599H
557E 3A 35 55     LD      A,(CNT)
5581 3C          INC      A
5582 32 35 55     LD      (CNT),A
5585 FE 02       CP      02H
5587 20 10       JR      NZ,5599H
5589 97          SUB      A
558A 32 35 55     LD      (CNT),A
558D CD 41 55     CALL    FNEXT
5590 C5          PUSH     BC
5591 2A 17 55     LD      HL,('DEBUG)
5594 5E         LD      E,(HL)
5595 23         INC      HL
5596 56         LD      D,(HL)
5597 EB         EX      DE,HL
5598 E9         JP      (HL)
5599 C3 57 55     JP      DNEXT

```

```
PNEXT is
55A6:  LD    HL,>NEXT
55A9:  LD    DE,DEBNEXT
55AC:  LD    (HL),C3
55AE:  INC   HL
55AF:  LD    (HL),E
55B0:  INC   HL
55B1:  LD    (HL),D
55B2:  JP    >NEXT
55B5:  END
```

```
UNBUG is
55BF:  CALL  FNEXT
55C2:  JP    >NEXT
55C5:  END
```

From:
<https://hc-ddr.hucki.net/wiki/> - **Homecomputer DDR**

Permanent link:
<https://hc-ddr.hucki.net/wiki/doku.php/forth/fgforth/debug>

Last update: **2025/07/21 08:55**

