Klaus Kohl:

Vokabulare in KK-FORTH

Folgender Auszug aus dem KK-FORTH für PC zeigt den Aufbau der Vokabulare in diesem FORTH. Die Struktur ist so gewählt, daß sie kompatibel mit dem volksFORTH und FORTH-83 ist. Es werden bis zu 6 CONTEXT-Vokabulare für die Befehlssuche unterstützt. Ich hoffe, Ihnen helfen diese Kurzinformationen.

```
  User voc-link         ( Vokabular-Verkettung )
  User current          ( Compiler-Vokabular )
    $12 uallot          ( 6 Adressen für Context-Vokabulare )
  : context     ( -- addr ) \ Adresse des aktuellen CONTEXT-Vokabulars
    current 2+  dup @  + ;
  : definitions ( -- addr ) \ CONTEXT-Vokabular als Zielvokabular
    context @  current ! ;
  : also        ( -- ) \ CONTEXT erweitern
    current 2+ @  $0c  >  ERROR" too many Vocs"
    context @   2  current 2+  +!   context ! ;
  : seal        ( -- ) \ CONTEXT kürzen
    current 2+  dup @  2-  0 max  swap ! ;
  : onlyforth   ( -- ) ( FORTH als CURRENT und 2*CONTEXT )
    2  current 2+  !   Forth   also   Definitions ;
| : >.id        ( cfa -- )
    >name ?dup IF .id exit THEN ." ???" ;
| : (@order      ( pfa+2 -- )
    @  2-  >.id  space ;
  : order       ( -- ) ( Anzeige aktiver Vokabulare )
    context 2+ current 4 + ?DO i (@order 2 +LOOP  space
    current                      (@order  ;
  : vocs        ( -- ) ( Anzeige aller Vokabulare )
    voc-link
    BEGIN @ ?dup WHILE dup body> >.id space REPEAT ;
  : Vocabulary  ( name ; -- ; -- )
    Create  here voc-link @ ,
                 voc-link !        ( VOC-Verkettung )
            0 ,                    ( Wortliste )
    Does>   2+  context ! ;

  Vocabulary Forth
```

Bei der Definition neuer Befehle werden die Header wie folgt in die Befehlsliste aufgenommen:

```
HERE                                    \ Adresse merken
CURRENT  @  @  ,                        \ Verkettung zum letzten Wort
CURRENT  @  !                           \ neuer Anfang merken
```

Die Befehlssuche gestaltet sich im Prinzip wie folgt (f enthält normalerweise Immediate-Flag):

```
  : find         ( csa -- cfa f | csa 0 )
    context 2+  current 4 +
    ?DO  i @ @
       BEGIN ?dup
       WHILE dup >r  2+
             ( CSA mit aktuellem Namen vergleichen. Ende wenn gefunden )
             ( Ende: nip  name>  -1   rdrop  unloop  exit )
             r>
       REPEAT
    2 +LOOP
    0 ; ( nicht gefunden )
```