

# Einführung in CORE WAR

## Teil 4

### Labels vereinfachen das Programmieren

Gegen *Knirps* oder gegen einen Gnom, der im Abstand von vier oder weniger Speicherzellen Bomben wirft, kann sich *Hase* erfolgreich behaupten. Was geschieht jedoch, wenn er etwa gegen einen *Gnom* mit größerem Bombenteppich antreten muss? Die Markierung von *Hase* ist nur vier Speicherzellen lang, so dass sie möglicherweise gar nicht von einer Bombe getroffen wird. Das Frühwarnsystem versagt, und bei *Gnoms* gibt es auch einmal wochentags Hasenbraten. Um dies zu verhindern, wäre es sinnvoll, die Markierung zu verlängern. Dabei stellt sich jedoch das nicht unerhebliche Problem, wo überall etwas in dem Programm *Hase* geändert werden muß. Dieses Problem würde sich nicht stellen, wenn man *Hase* folgendermaßen programmiert hätte:

```
; Redcodeprogramm "Hase" mit Labels und Konstanten:

; Definition von Konstanten

MarkD      EQU      500          ; Entfernung der Markierung vom
                                   ; Programm
MarkL      EQU      8           ; Laenge der Markierung
CodeNr     EQU      42          ; Vergleichszahl
FluchtD    EQU      1000        ; Fluchtweite

; Anweisungsteil von "Hase"

; Aufbau der Markierung

A          MOV      Code        <MarkPos
          DJN      -1           #MarkL
          MOV      #MarkL      -1
          MOV      #-MarkD-5    2
          JMP      Pruef
MarkPos    DAT      #0          #-MarkD-5
Code       DAT      #0          #CodeNr

; Abtasten der Markierung

          DAT      #0            #0
Pruef      MOV      #-MarkD-7    -1
Vergleich  CMP      #CodeNr      <-2
          JMP      KopPrg
          DJN      Vergleich     #MarkL
          MOV      #MarkL      -1
          JMP      Pruef

; Kopierroutine

KopPrg     MOV      #PrgLaenge    7
          MOV      #E-KopZaehl+1  3
          MOV      #A-FluchtD-KopAdr 3
          JMP      KopAnw
KopZaehl   DAT      #0            #0
KopAdr     DAT      #0            #0

KopAnw     MOV      <-2          <-1
          DJN      KopAnw      #0
E          JMP      @-3

PrgLaenge  EQU      E-A+1        ; Anzahl der Anweisungen
          END
```

Lassen Sie zunächst den ersten Teil von *Hase* außer acht. Vor einigen Anweisungen stehen sogenannte Labels, die sich aus alphanumerischen Zeichen zusammensetzen. Sie markieren die Anweisung, der sie vorangestellt sind, so dass etwa durch die Anweisung **JMP pruef** zu der Anweisung gesprungen wird, die mit **pruef** markiert ist.

Am Anfang des Programms ist ein neuer aus **EQU**-Direktiven bestehender Teil hinzugekommen. In ihm werden einige Konstanten definiert. Links am Zeilenanfang steht der Name der zu definierenden Konstanten, der sich aus alphanumerischen Zeichen zusammensetzt. Er wird gefolgt durch das Kürzel **EQU** (nach dem englischen Wort *equal* für gleich). Rechts steht der definierende Ausdruck. Er besteht aus ganzen Zahlen, Konstanten und Labels, die durch die vier Grundrechenarten verknüpft sind; wie üblich geht Punktrechnung vor Strichrechnung. Da man in den ganzen Zahlen rechnet, ist die Division insofern eingeschränkt, als die Nachkommastellen abgeschnitten werden. Ebenso wie bei den **EQU**-Direktiven können Ausdrücke auch bei den Operanden verwendet werden.

Nun zurück zu dem Problem, die Markierung von *Hase* zu verlängern: Am Anfang des Programms wurde die Konstante **mark1** definiert. Sie wurde dann überall dort verwendet, wo die Länge der Markierung eine Rolle spielt, so dass man jetzt nur die Konstantendefinition **mark1 EQU 4** zu **mark1 EQU 8** ändern muss, damit sich *Hase* eine Markierung der Länge acht anlegt.

Quelle: Aus dem Begleitheft zu Core War – Krieg der Kerne, Spektrum der Wissenschaft 1993, S. 30 ff.