Do-Loop Exit Address on Return Stack and ?LEAVE

Allen Anway, Academic Computing, UW-Superior, Superior, WI 54880

Since the Forth Do-Loop "wrecks" the return stack with loop indices, I decided in my Forth to put the loop exit address there. My branch implementation uses absolute addresses, so I borrowed >MARK and >RESOLVE to compile an address after (DO) that will later work as an EXIT type of departure. (DO) execution puts the exit, fixed index, and moving index on the return stack in that order.

Advantages: (a) No extra addresses after LEAVE or ?LEAVE are required. This simplifies the branch type compiler as exit addresses are unstructured with respect to IF THEN structure. (b) ?LEAVE now unnests one with -1 flag, zero with 0 flag, but unnests N with N flag. (c) The words (DO) (LOOP) and (+LOOP) set LEAVAL to value 0. The words LEAVE and ?LEAVE set LEAVAL to N. The programmer calls LEAVAL to check if early loop departure took place.

Unnesting N loops is structured: you know the return site. If one seeks a number in a multi-dimensional array, and upon finding it wishes to quit the nested loops, the (N ---) ?LEAVE with LEAVAL (--- N) trivially solves this otherwise difficult problem.