FORTRAN: Pascal External Functions and Procedures

When you use assembly language functions and procedures from FORTRAN, all parameters are passed to these functions and procedures as formal parameters. This means that when a function requires an integer, real number, boolean or character, it is the address of the variable being passed that resides on the stack, not the contents of the variable. Therefore, assembly language functions and procedures should load the address of the variable from the stack and use this address to retrieve the data in the variable.

Because of this small detail, it may not be possible to use some assembly language functions and procedures developed in Pascal from FORTRAN, since Pascal will pass data that requires two to four bytes of space on the stack. Data types requiring more space than two words (such as long integers, arrays, strings, and so on) are sent as formal parameters; therefore, the address in memory where the variable resides is passed on the stack, and the data is accessed through that address. When an external function or procedure in Pascal uses formal parameters, it can easily be used in FORTRAN; otherwise, it will have to be rewritten to handle the formal parameter stack format of FORTRAN.

If you wish to see a specific example of this difference, compare the assembly language listings of the paddle functions on page 120-121 of the Apple FORTRAN reference manual with the same listing on page 143-144 of the Apple Pascal Operating System Reference Manual.

NOTE: A FORTRAN host program calling a Pascal REAL subroutine with a VAR
      parameter may cause a "Stack Overflow".
Apple Technical Communications

Keywords: <None>

===

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 133