Tech Info Library

Pascal II: Operand Formats (1 of 4)

Revised: 12/5/84
Security: Everyone

Pascal II: Operand Formats (1 of 4)

==

When you need to send data (especially complex data formats, such as strings) to an assembly routine from a Pascal host program, it can be very useful to be familiar with the internal structure of Pascal variables. This article describes a few of the more commonly used variable types; for a complete description of the more complex variables, including records and arrays, see pp. 227-228 of the Apple Pascal Operating System Reference Manual.

Machine language (assembly) routines are commonly used either when (a) speed is critical, or (b) when the code must access other assembly routines (such as PROMs or I/O drivers) that can't be reassembled as part of the program. Also, bit manipulations such as right-shift are much easier to do in assembly than in Pascal.

In the UCSD Pascal system, it's fairly easy to create short assembly programs which can be linked into a Pascal host program. In some cases, it may be sufficient to merely call the assembly routine; most routines require that data be passed to them, though. Data is passed to or from routines by means of a "parameter", a temporary variable created by Pascal specifically for that purpose. The term "Var parameter" implies that the address of the actual variable is passed to the routine as a parameter instead of its value.

Certain types of variables may be passed by value, but any variable may be passed by name by simply declaring it to be a Var parameter. Pascal does not allow parameters of variable length (with the exception of certain sets and long integers) to be passed on the CPU stack, since doing so could end up filling the stack to capacity and thereby crashing the operating system. These parameters, therefore, are automatically used as if defined as Var parameters. A good explanation of the various methods of passing parameters may be found in Peter Grogono's book, "Programming in Pascal".

Before delving into the details, let's define some terms and conventions which we'll use later on:

Bit = a binary digit (0 or 1). A bit is the smallest unit of information which can be stored in a computer.
Nybble = 4 bits (half a byte). A hexadecimal digit is one nybble

```
                 (pronounced "nibble").
        Byte =   8 bits (2 nybbles).  This is the unit of storage which the
                 6502 processor uses.
        Word =   2 bytes (16 bits).  A word is the unit of information which
                 Pascal uses.
        LSB =    least significant bit
        MSB =    most significant bit

        decimal 65535                                    0
    hexadecimal $FFFF <--------memory---------> $0000 addresses
                  MSB                                  LSB
```

This diagram of memory structure is useful for understanding the format of variables: although we're used to writing numbers from left to right, Pascal reads data from memory FROM RIGHT TO LEFT, starting at the least significant byte.

Apple Tech Notes

Tech Info Library Article Number:686