Tech Info Library

Macintosh Pascal: Memory Use by Programs

Revised: 7/30/87
Security: Everyone

Macintosh Pascal: Memory Use by Programs

===

About 35K of memory is available not only for your program text, but for program variables, the program's pseudo-code representation, debugging information and libraries. The majority of your Macintosh memory is used to keep as much of Macintosh Pascal resident in RAM as possible in order to minimize disk I/O. On a 512K Macintosh or Lisa, only 64K is available, due to pointer size limitations.

Version 2.0 of Macintosh Pascal lets you write larger programs, depending on the contents of the program, of approximately 750 lines on a 128K Macintosh and 2,000 on a 512K Macintosh.

Inside Macintosh provides a brief introduction to memory management in the Macintosh in the chapter "Macintosh Memory Management: An Introduction". Macintosh Pascal uses the Macintosh memory in a slightly different way.

The 2.0 version of Macintosh Pascal includes new predefined procedures for manipulating relocatable objects. (A relocatable object is an object that can be moved within memory and is referred to indirectly by means of a handle.) Anyone using these procedures needs to have a basic understanding of how Macintosh Pascal uses memory.

Macintosh Pascal uses two main run-time data structures: the machine stack and the general stack. The machine stack grows downward from high memory. From the bottom of the application heap just above the master pointer area, the general stack grows upward. The second stack is actually a pointer object that has a base fixed throughout a Macintosh Pascal session. When either the interpreter or the user interface is running, its code segments are moved to the top of the application heap and locked. Thus, programmers who cause objects to be allocated in the heap must be very careful not to cause the objects to be locked down just above the general stack. This will almost always cause the interpreter to run out of memory and exit to the Finder. Similarly, objects should not be allowed to be locked down just below the code segments that are locked high.

Tech Info Library Article Number:999