New Technical Notes

Macintosh

®

Developer Support

Debugging Tips

Overview

Revised by: March 1988
Written by: Jim Friedlander April 1986

This presents a few tips which may make your debugging easier.

Setting memory location 0 to something odd

Dereferencing nil handles can cause real problems for an application. If location 0 (`nil`) is something even, the dereference will not cause an address error, and the application can run on for quite a while, making tracing back to the problem quite difficult. If location 0 contains something odd, such as `$50FFC001`, an address error will be generated immediately when a nil handle is dereferenced. On Macintoshes with 68020s, like the Mac II, this same value (`$50FFC001`) will cause a bus error. An address error or bus error will also be generated, of course, when the ROM tries to dereference a nil handle, such as when you call `HNoPurge(hndl)`, where `hndl` is nil.

Some versions of the `TMON` debugger set location 0 to 'NIL!' `($4E494C21)` or `$50FFC001`. If you are using MacsBug, you should include code in your program that sets location 0. Of course, there is no need to ship your application with this code in it—it's just for debugging purposes. Old versions of the Finder used to set location 0 to the value `$464F424A ('FOBJ')`. On newer machines, newly launched applications get location 0 set to `$00F80000` by the Segment Loader.

Checksumming for slow motion mode

Entering the Macsbug command "`SS 400000 400000`" will cause Macsbug to do a checksum of the location `$400000` every time an instruction is executed. Checksum the ROM, because it will not change while your program is executing (the ROM may change in between launches of your application, but that's OK)! This will cause the Macintosh to go into slow motion mode. For example, you will need to hold down the mouse button for about 10 seconds to get a menu to pull down—you can see how the ROM draws menus, grays text, etc.

This technique is very handy for catching problems like multiple updates of your windows, redrawing scroll bars more than once, that troublesome flashing grow icon, etc. To turn slow motion mode off, simply enter MacsBug and type "SS".

TMON performs this function in a different way. Instead of calculating the checksum after each instruction, it only calculates checksums after each trap. You can checksum different amounts of the ROM depending on how much you want things to slow down.

Checksumming MemErr

A lot of programs don't call `MemError` as often as they should. If you are having strange, memory-related problems, one thing that you can do to help find them is to checksum on `MemErr` (the low memory global word at `$220`). In MacsBug, type "`SS 220 221`". In `TMON`, enter `220` and `221` as limits on the '`Checksum (bgn end):`' line and on the line above, enter the range of traps you wish to have the checksum calculated after.

When `MemErr` changes, the debugger will appear, and you can check your code to make sure that you are checking MemErr. If not, you might have found a problem that could cause your program to crash!

Checksumming on a master pointer

Due to fear of moving memory, some programmers lock every handle that they create. Of course, handles need only be locked if they are going to be dereferenced and if a call will be made that can cause relocation. Unnecessarily locking handles can cause unwanted heap fragmentation. If you suspect that a particular memory block is moving on you when you have its handle dereferenced, you can checksum the master pointer (the handle you got back from NewHandle is the address of the master pointer). Your program will drop into the debugger each time your handle changes—that is, either when the block it refers to is relocated, or when the master pointer's flags byte changes.