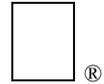


# Apple II Technical Notes



---

Developer Technical Support

## GS/OS

### #9: Interrupt Handling Anomalies

Revised by: Matt Deatherage May  
1990

Written by: Dave Lyons  
January 1990

This Technical Note discusses anomalies in the way GS/OS handles interrupts.

**Changes since January 1990:** Added discussions about calling `UnbindInt` at inopportune times and changes in GS/OS interrupt handling with respect to the Apple II High-Speed SCSI Card.

---

### Problems Installing Interrupt Handlers

If your application calls `ALLOC_INT` to install an interrupt handler for an external interrupt source, it works fine **unless** the SCSI Manager (GS/OS file `SCSI.Manager`) is installed, in which case the system eventually grinds to a halt with a message about 65536 unclaimed interrupts.

### The Problems

If any interrupt handlers are bound (using `BindInt`) to reference number \$17 (`IRQ.OTHER`), the unclaimed interrupt count gets incremented if none of the `BindInt` routines claims the interrupt, even though any handlers installed with `ALLOC_INT` routines still need a chance to claim it. The 5.0.2 `SCSI.Manager` triggers this problem because it calls `BindInt` with vector reference number \$17.

In addition, if one or more interrupt handlers are bound to the `IRQ.OTHER` vector (VRN \$17),

the interrupt is passed to the `ALLOC_INT` handler even if it was already claimed by a `BindInt` routine. If no `ALLOC_INT` routine claims the interrupt, the unclaimed-interrupt count is incremented. As documented in Apple IIGS Technical Note #18, *Do-It-Yourself SCC Interrupts*, you cannot successfully call `BindInt` with vector reference number \$0009.

## The Solution

An application may install **both** a `BindInt` routine and an `ALLOC_INT` routine. If they both claim the external interrupt, the unclaimed count does not get incremented. The solution is compatible with future System Software releases, since it does not depend upon the `ALLOC_INT` routine ever getting called.

Your application's `BindInt` routine sees the interrupt before your `ALLOC_INT` routine does, so the `BindInt` routine should figure out whether the interrupt was caused by your external device, and claim it if so. Your `ALLOC_INT` routine should claim an interrupt it sees if and **only** if your `BindInt` routine claimed the last interrupt it saw.

GS/OS 3.2, which was released with the Apple II High-Speed SCSI Card, no longer treats more than 64K unclaimed interrupts as a fatal error. Note that GS/OS still displays a dialog box asking the user to restart the system, but choosing the Restart button returns control to the application with no ill effects.

## Problems Removing Interrupts Handlers

The *GS/OS Reference* suite says that device drivers may make `BindInt` and `UnbindInt` calls, noting this as an exception to the general rule that drivers may not make GS/OS system calls. What the references fail to note is that these calls may fail for an incredibly annoying reason—the OS may be busy.

GS/OS takes special pains to avoid this while starting and while switching to ProDOS 8, but it does not avoid this condition during an `OSShutdown`—a real shutdown of the OS, not a switch to ProDOS 8.

Driver authors can work around this problem by using a new system service call provided in GS/OS version 3.2 and later. The call, named `UNBIND_INT_VECTOR`, provides the functionality of `UnbindInt` to FSTs and drivers **only** to avoid the OS reentrancy issue. The vector is at `$01/FCD8` and takes an interrupt identification number (as returned from `BindInt`) in the accumulator.

## Further Reference

---

- *GS/OS Reference*
- Apple IIGS Technical Note #18, Do-It-Yourself SCC Interrupts