



Q: The DOS Library function FGets() seems to read too many bytes if it doesn't come across a newline character. What's happening?

A: This is an excerpt from a recent Autodoc for FGets:

In V36 and V37, it copies one more byte than it should if it doesn't hit an EOF or newline. In the example above, it would copy 50 bytes and put a null in the 51st. This is fixed in dos V39. Workaround for V36/V37: pass in buffersize-1.

Q: Is there an interaction between DMA devices (such as hard drive controllers) and the 68040?

A: Yes. This interaction is only an issue when the 68040 is in copyback data cache mode and the timing is just right. Note that the chance of this interaction affecting any particular application is rather small. The interaction is as follows:

The 68040 data cache has a mode called copyback. In this mode, when the CPU modifies memory, it does not write the data to memory right away. It waits until the cache is needed to cache some other memory or someone explicitly flushes the cache. This creates a problem for DMA devices

because they can read directly from memory without using the CPU. If a DMA device tries to read memory that happens to be in the CPU's data cache, the DMA device will read the wrong data.

To prevent problems with DMA, a DMA device has to call CachePreDMA() before accessing memory. This function does both address translations (for future MMU issues) and cache flushing as needed. When the transfer is complete, the device has to call CachePostDMA() which performs some cache flushes and whatever cleanup it needed.

Unfortunately, there is a nasty interaction between the copyback mode of the 68040 and flushed data areas.

In copyback mode, the 68040 uses complete cache lines (which are four long words in length) when copying data back and forth from the cache. There are good reasons for this, including that the CPU can do a full burst transfer (which is four long words). However, the 68040 does this even if only 2 of the long words in the cache line are changed. So, what does this mean?

Imagine a program that has two memory buffers right next to each other. The program is running on a 68040 in copyback mode. These buffers meet in the middle of a 128 bit boundary, so a cache line would overlap the end of the first buffer and the

Amiga Mail

Volume II



