
Subject: Re: OpenGL + Linux crashes

Posted by [Karl Schultz](#) on Tue, 09 Mar 2004 20:27:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Michael Wallace" <mwallace.removethismunge@swri.edu.invalid> wrote in message news:104s3g1s2kg8vf1@corp.supernews.com...

>> when attempting to use any of IDL's OpenGL 3D stuff (like the Demo ->

>> ltools, for example), you might try the following:

>>

>> setenv MESA_NO_ASM 1

>>

>> which disables some specific ASM code in the Mesa library which was

>> causing these types of crashes for me. I use an ATI Radeon 7500 +

>> XFree86 4.3.0's radeon drivers. With this fix in place, it seems

>> stable, and is definitely much faster than software rendering. Give a

>> a try.

>

> This is awesome! I also have a Fedora box with an ATI Radeon 7500 and

> it works great. I'm just curious what assembly language optimizations

> are conflicting with IDL's OpenGL stuff. Oh, well. At least it works

> better than using software rendering everywhere.

On recent Linux/XFree86 installations, there are actually two instances of the Mesa library in play when you run IDL. One is the Mesa that is linked directly to IDL that IDL uses to perform software rendering and really isn't involved any further in this discussion. The other instance is over in the X server where it implements OpenGL (via GLX) in software if there is no hardware acceleration support. For systems with the hardware and software support for hardware acceleration, Mesa still serves as the OpenGL implementation and it uses various driver modules to interface with the specific hardware.

When a client (like IDL) connects to an X server with the hardware acceleration support, it is really slow to send the GL commands through the X server via X/GLX protocol. So, there is a module called DRI (Direct Rendering Infrastructure) that is used to connect the client application "directly" to the hardware, or more accurately, to the Mesa interface layer that implements OpenGL for the device.

When this Mesa module starts up, it attempts to see if the SSE (Streaming SIMD (Single Instruction Multiple Data) Extensions) instructions are available on the CPU. Unfortunately, one of the steps that must be taken in order to do this reliably on Linux is to force an exception and poke around in the CPU status registers to see if it did what it was supposed to if the SSE instructions are present. The code that does this installs its own exception handler and then removes it when finished. Apparently there is something wrong with this particular code sequence that causes an exception to occur anyway. After reading the code, my *guess* is that the exception

handler code isn't clearing the exception state out of the CPU and then IDL is tripping over it later. It turns out that another application (SDL) has the same problem. I'll be installing Fedora Core 1 soon to investigate further and try to get it fixed in Mesa.

Because it is so tricky to detect and use SSE, the Mesa developers wisely made it possible to avoid it all by setting the env var mentioned above.

Anyway, the SSE instructions are used to speed up common math operations in Mesa, like coordinate transforms and matrix multiplies. These would be more important if Mesa was doing most of the rendering work in software. If you have a good graphics card that does transforms and lighting in hardware, much of this math would occur on the graphics card and so turning off SSE won't hurt performance as much.

If I learn anything more, I'll post it.

Karl
