Subject: : AmigaOS4
Topic: : Porting apitrace
Re: Porting apitrace
Author: : kas1e
Date: : 2019/4/5 20:15:10
URL:

@Capehill,Hans
I asked Daniel about, so hope he doest mind if i put info there about.

Daniel says he do nothing special for. The only thing is that ogles2's interface is private. Simply because it contains contextual data, naturally, which means that every process gets its own interface, which is probably the reason why we fail.

However, patching inside the respective prog should work.

Unless if we sure, that our ogles2 calls aren't done through function pointers we obtained via aglGetProcAddress (probabaly just like we do in SDL and GL4ES), if that the case, then we can patch the interface (almost) as much as we want and probably won't see any effect.

So, then i go easy route: i just take some pure SDL2/OGLES2 example, and put at begining of main() that patching code which patch just single glCompileShader() , and it works. When i run my test case (which do some rotate cube) i have on serial:

Patched glCompileShader 0x7F686064 with 0x7FA32588
my_glCompileShader: shader 256
my_glCompileShader: shader 257

So from the programm itself patching via SetMethod surely works, and SDL2 didn't break anything there in terms of pointers, etc.

Now need to understand how to patch it in general way, so without needs to touch program's code.

Maybe patch the app in realtime ? I mean as apitrace runs as "apitrace trace --api [gl|egl|w3dn] /path/to/application [args...]" , it mean that we know what application to debug. Then, we can take the "main" section of the elf , and add at begining assembler code which will just execute our patched code, and at end of main, remove them. Sure a bit of hardcore, but just as first idea.