
Subject: Re: should ultimate++ support dynamically linked libraries?

Posted by [akspring](#) on Sun, 26 Jan 2014 12:37:46 GMT

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

Novo wrote on Tue, 10 December 2013 22:50

And dynamic libraries is a back door for hackers. They allow to install hooks on Windows and use LD_PRELOAD on Linux.

Upp apps are small. Static linking makes them also fast.

From what I understand, dlls, vlc controls, and anything like chromium and ie controls are equivalent to handing your computer over to military trained cyborg ninja assassins from the future. You may as well forget about security if you include these common place controls. And they are everywhere.

I like to think U++ apps by themselves would be safe to use, but its really just a guess. I use Windows, so security for me and everyone else is really a joke. Same with Linux.

I don't think anyone cares about that much, so whats left is what will make U++ great? - I think its size, speed, and portability. With 7z/LZMA2, I can get a 15mb U++ application with debug info down to 1.8mb.

And 7z is really garbage when compared to costly closed source compression apps, so I bet I could have a self unpacking executable with debug info in under a megabyte. And without debug info my executables are about 3mb, 2mb with Windows compression, and under a megabyte with LZMA2. Just imagine if I paid for a high quality compression app. 500kb?

Its really nice to see something effective and fast, have many nice features like SQL and File Systems and be able to use it on any platform without needing something like I don't know, .NET? What are they doing now 250mb for the framework? I thought 30mb was a lot and that was just with 2.0

With U++ no matter how much I modulate my dlls, the only thing people will have to do is replace their single executable and my next big thing is ready to go. No other advertised big name open source cross platform software can compete like U++. My apps are fast, portable, lightweight, ready to use, and the Core framework gives me plenty of functionality. And my compile times are amazing. I feel it delivers in many ways where standard C++ IDEs fail as well as compensates for many C#/.NET features I want, without the pain of the managed and restrictive CLI. I dropped CodeBlocks and haven't looked back
