Subject: cross compile from win32 to Linux
Posted by Andy2222 on Sat, 01 Dec 2007 23:36:28 GMT
View Forum Message <> Reply to Message

Is there some easy way to setup the U++ IDE to output Linux compatible files/bins? Meaning cross compile on windows for Linux?

As far as i can see only the mingw libs are included in the windows version, so what all is needed for the Linux setup?

thx

Subject: Re: cross compile from win32 to Linux
Posted by forlano on Sun, 02 Dec 2007 14:56:16 GMT
View Forum Message <> Reply to Message

Andy2222 wrote on Sun, 02 December 2007 00:36Is there some easy way to setup the U++ IDE to output Linux compatible files/bins? Meaning cross compile on windows for Linux?

As far as i can see only the mingw libs are included in the windows version, so what all is needed for the Linux setup?

thx

Hi,

as far as I know there is no such possibility.
What you can do is to compile the same source code on a linux platform.

Luigi

Subject: Re: cross compile from win32 to Linux
Posted by zsolt on Sun, 02 Dec 2007 19:12:06 GMT
View Forum Message <> Reply to Message

Maybe an alternative way could be, that AFAIK, you can setup remote builds, so you can edit your sources on Windows and build them on a remote Linux machine.

Subject: Re: cross compile from win32 to Linux
Posted by Andy2222 on Tue, 04 Dec 2007 12:36:32 GMT
View Forum Message <> Reply to Message

Thx for the infos, that was what i was afraid of. Seems realy working cross compilers + ide is still at the state it was 10 yea hrs ago

At least we have MinGW

I guess i will have to work on windows and simply setup a second compile environment on my remote linux server.

At least im happy i stumbled around Ultimate++ i cant wait to test out the UI and Container stuff Im also surprised how easy UTF-8 vs WideChar's are handled, i still have nightmares working on a directshow project with mixed interfaces.

PS: What is the recommend way to have 16 byte aligned memory for a container or all internal storage? Need this for propper SSE intrinsics? The silly way was always globaly override new... and route it to _aligned_malloc or use my own stl aligned_allocator.

Subject: Re: cross compile from win32 to Linux
Posted by forlano on Tue, 04 Dec 2007 12:55:31 GMT
View Forum Message <> Reply to Message

Andy2222 wrote on Tue, 04 December 2007 13:36
PS: What is the recommend way to have 16 byte aligned memory for a container or all internal storage? Need this for propper SSE intrinsics? The silly way was always globaly override new... and route it to _aligned_malloc or use my own stl aligned_allocator.

Please post this in the appropriate forum or your question can be hided and risk no answer.
Moreover this can help the search for other users having your same problem.
Thanks.
Luigi

Subject: Re: cross compile from win32 to Linux
Posted by mr_ped on Wed, 05 Dec 2007 00:24:05 GMT
View Forum Message <> Reply to Message

You may also try virtualization?

VirtualBox + some linux distro + Ultimate = compilation for linux "under windows". (Or vmware is total classic, but I hear the new VirtualBox is quite powerful/stable and easy to use)

If it will work, that is. (I didn't personally try this setup, but I'm sort of interested into it too, just the other way around, with windows virtualized under linux, as kubuntu is my main system)

Subject: Re: cross compile from win32 to Linux

Posted by waxblood on Wed, 05 Dec 2007 12:52:30 GMT
View Forum Message <> Reply to Message

I've tried VirtualBox with the Windows-under-Linux formula (very funny having Windows dwarfed inside a X11 window) and must say I'm happy with it, except that file transfer between the two systems via shared folders is a bit slow.

David

Subject: Re: cross compile from win32 to Linux
Posted by captainc on Tue, 29 Apr 2008 12:08:59 GMT
View Forum Message <> Reply to Message

I also develop for both platforms by using Windows XP in a VirtualBox VM. It works great. I don't like the shared folders of VirtualBox as I had some issues with it, so I set up Samba server on Linux and used regular network sharing, so the host and guest OSes appear to be on the same network (useful for testing cross-platform network apps too).

Also, I set up an SVN server on the Linux (host) machine and check in/out the code from all machines so that I don't have to keep copying files back and forth with the networked folders. This will additionally allow you to branch for testing os-specific features that you may have to implement.