Subject: Re: Do we have support for Windows on ARM64?
Posted by mirek on Fri, 05 Dec 2025 10:06:11 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Fri, 05 December 2025 10:53Hi Mirek,

OK, now I have the new laptop with Windows 11 ARM64 up and running. The nightly upp installed easily and runs in the Prism emulation (embedded in Windows 11 ARM by default) very well and does not feel sluggish at all.

The next step I took was downloading ARM64 version of the llvm tools from:

https://github.com/mstorsjo/llvm-mingw/releases/download/20251202/llvm-mingw-20251202-ucrt-aarch64.zip

Then I extracted the contents and replaced the entire C:\upp\bin\clang -directory with the contents from this archive. I opened theide and the UWord from examples. As I tried to compile it, the compiler first complained about the 'Common option -mpopcnt' which is not supported by aarch64. So I removed that option.

I also changed the PATH executable directory: C:\upp\bin\clang\x86_64-w64-mingw32\bin path to point at C:\upp\bin\clang\aarch64-w64-mingw32\bin instead.

I made those changes to a new build method 'CLANGARM64' using 'CLANG' builder. I know this is not complete and probably needs more changes for at least the other paths, includes and libraries, depending on the program being built.

Next issue was found on Core\Debug.cpp around line 270.

Slightly off topic, you do know about "Misc/Copy current position" ?

Quote:
The 'qword esp = ep->ContextRecord->Rsp;' and 'dword esp = ep->ContextRecord->Esp;' do not exist here. I do not know how to really fix this, but as a quick and dirty workaround I decided to replace that code with just 'qword esp = 0;' and consequently the UWord compiled, linked and runs OK!

That whole thing can go away I guess, it is fossil...

Quote:
So, I guess all this is doable, but requires some tuning in TheIDE so that everything runs natively out of the box. Can you take a more professional look at this based on this description?

I guess there 2 hard problems ahead:

- libraries that we ship with (esp OpenSSL). that is even more sad as updating libraries takes awful amount of time, now there would be another architecture...
- debugging. That might possibly work (test please), but at minimum the dissassembler only supports x86.

Mirek