Subject: Re: Compilation on Mac
Posted by fudadmin on Mon, 02 Jan 2023 17:51:22 GMT
View Forum Message <> Reply to Message

brown wrote on Mon, 02 January 2023 17:03In case of the sysctl.h include is missing from Cpu.cpp, I got 3 compilation errors on Cpu.cpp lines 168,169,173 like undeclared identifier 'CTL_HW', 'HW_MEMSIZE' and sysctl() fn prototype.
Perhaps, this is just a side effect of something else, which I didn't find yet... But there is a GetSystemMemoryStatus(r,r) fn implementation in this file, which have multiple "body" regarding to the PLATFORM_MACOS "switch", the referenced fn and sys ctrl codes are came from the sys/sysctl.h include file according to man 3 sysctl in this platform...
I compiled the umk here (because the normal make was getting failed earlier - known fixed reason). So, if I run the
make -f umkMakefile
command, this was the remaining issue, which can be fixed by adding the specific include to the file.
Yeah, probably it should be included from somewere else, I can imagine...

Back to the previous thread (compile mm files), I just wanted to build the umk and theide from its source on mac. There are .mm files in example: Draw/FontCoco.mm.
Some reason, my make still not able to link, because of the Makefile doesn't contains any *.mm files at all... I guess those are missing from my Makefile :).
How could you compile the upp itself on Mac with the actual Makefile.in content?

You are trying a very very complicated way... :) Try 3 simple steps:
1. Please download working macos version from here:
https://sourceforge.net/projects/upp/files/upp/2022.2/
2. It should contain an older working ide. Then, with the ide, try to build at least one reference package. (Report errors here on forums, if any)
3. If that goes ok, create a new uppsrc assembly and point to a folder from my fork branch. Build/execute.

P.S. Yes. You are right . There are problems for the macos version 13 because of sysctl.h. And that include should contain version checking.
Then the problems arise with linking upp x86_64 with macos arm libs...
To solve it would be needed to find how to switch compiling upp into correct CPU_ARM or __aarch64__ or just __arm__ for macos? Or some changes in Core.h would be needed?

P.P.S Putting into *.bm (Build methods) -target arm64-apple-darwin. or arm64-apple-macos* switches upp core/config.h aarch64 correctly. But the compiled *.o files are x86_64. I suspect theide internal builder but need more investigation. Commmand line check clang --version gives arm64. I think, Mirek already has the answers. :)