Subject: Re: Compilation on Mac
Posted by fudadmin on Tue, 03 Jan 2023 01:27:35 GMT
View Forum Message <> Reply to Message

brown wrote on Tue, 03 January 2023 00:56Thank you, it works with an earlier theide...
I got the point, added your branch as an assembly, using the default arch settings, then I got an error for crypto due to the different arch:
 (): ld: warning: ignoring file /opt/homebrew/Cellar/openssl@3/3.0.7/lib/libcrypto.a,
 building for macOS-x86_64 but attempting to link with file built for macOS-arm64

When I change the build config to add -target arm64-apple-macos11, then I get again the previous problem (sysctl related compiler error).
Therefore I add again the #include <sys/sysctl.h> onto your source tree, and the compilation + link gets success then. (no error, 32 warning)
Furtheremore the newly built "ide" app is running on m2:
% file ./ide
./ide: Mach-O 64-bit executable arm64
while the downloaded previous upp theide is x86
file ./theide
./theide: Mach-O 64-bit executable x86_64

There was an .so / lib problem popup regarding libClang with the new ide, also only an external debugger can be run (lldb) but internal gdb method is not working.
Obviously, I have to set the CLANG.bm correctly. by adding openssl include and lib as well.
I will soon retest my projects based on arm64 and the latest possible sources...

Fantastic news! Congratulations! I suspect we have a VERY promising member... Would you mind to share your *bm file. And/or test if -target arm64-apple-macos13 works?

P.S. Tested. Mine working CLANG_ARM.bm is as follows (might need to clean some surplus -target...):

```
BUILDER = "CLANG";
COMPILER = "clang++";
COMMON_OPTIONS = "-mmacosx-version-min=13 -DTARGET_CPU_ARM64 -target
arm64-apple-macos13";
COMMON_CPP_OPTIONS = "-std=c++17 -target arm64-apple-macos13 -Wall
-Wno-logical-op-parentheses -Wno-deprecated-anon-enum-enum-conversion
-Wno-deprecated-declarations";
COMMON_C_OPTIONS = "-target arm64-apple-macos13";
COMMON_LINK = "-target arm64-apple-macos13";
COMMON_FLAGS = "";
DEBUG_INFO = "2";
DEBUG_BLITZ = "0";
DEBUG_LINKMODE = "1";
DEBUG_OPTIONS = "-O0";
DEBUG_FLAGS = "";
```

DEBUG_LINK = "";
RELEASE_BLITZ = "0";
RELEASE_LINKMODE = "1";
RELEASE_OPTIONS = "-O3 -ffunction-sections -fdata-sections";
RELEASE_FLAGS = "";
RELEASE_LINK = "";
DEBUGGER = "gdb";
ALLOW_PRECOMPILED_HEADERS = "0";
DISABLE_BLITZ = "0";
PATH = "";
INCLUDE = "/opt/X11/include;/opt/X11/include/freetype2";
LIB = "/opt/X11/lib";
LINKMODE_LOCK = "0";

P.S. It is better to use -arch arm64 flags for compiling and linking than the mentioned above if you do not want restrict your users.
