BRIAN R. HALL

Division of Information
Technology & Sciences
Champlain College
163 South Willard St.
Burlington, VT 05402

hall (at) champlain (dot) edu
(802) 865-5730

RSS >

Installing SASM on a Mac

posted Jan 31, 2015, 8:51 PM by Brian Hall   [updated Jan 31, 2015, 9:40 PM]

One of the courses I teach at Champlain College is CSI 370 Computer Architecture, a course in which we discuss and write a considerable amount of assembly. Students can decide which assembler they want to use, which is primarily based on what OS they feel like working in throughout the course. The three assemblers I discuss directly as part of the course are GAS, NASM, and MASM. Personally, I am a Mac user, most of my programming is in C++, and I like Xcode. Packaged with Xcode is the GNU assembler (as or GAS), so I go back and forth between working with GNUas in Xcode and NASM in the terminal environment.

For the students, Visual Studio is their primary IDE, but VS is limited to MASM and it doesn't color code assembly. Therefore, many students gravitate to SASM, which is a great multi-assembler IDE for Windows and Linux. SASM allows a programmer to quickly flip between assemblers, debug, inspect registers, and it even color codes assembly…hooray!

Now, I'm perfectly happy using GNUas in Xcode, but I can appreciate SASM as a tool for writing NASM code and providing a visual debugger. And so I decided I would install SASM on my Mac. But this presents several issues:
* There are not any SASM packages for MacOS, just Windows and a few Linux distros.
* SASM relies on gcc and gdb, and Apple has now moved away from the land of GNU in favor of LLVM and clang.
* I found no evidence that anyone else had tried to install SASM on a Mac, I didn't look too hard so it may have happened before, but this was a mostly documentation-less process.

So, I decided to document the steps in case anyone else decides they want to install SASM on a Mac and they go about searching on the Web, perhaps they'll come across this.

1. Download SASM's Linux source code from http://dman95.github.io/SASM/english.html
2. SASM is built on Qt, so if you don't have Qt, download Qt from http://qt-project.org
3. Install Qt
4. In order to be able to run qmake, add the following path to your .bash_profile
       PATH=$PATH:/Users/hall/Qt/5.4/clang_64/bin (with obvious changes for your username and Qt v.)
5. Untar/Unzip the SASM source code you downloaded if you haven't already.
6. Open Terminal and cd to the SASM source code directory, then run:
       $ qmake
       $ make
       $ sudo make install
7. The SASM .app will now be in the directory, you can move or copy the sasm.app wherever you want:
       /Applications for example
8. If you have gdb installed, skip to step 12.
9. If you don't have gdb installed, the easiest way to install it is via MacPorts or Homebrew. I use MacPorts. If you don't have MacPorts then download from https://www.macports.org and install.
10. Then install gdb via Terminal:
       $ sudo port install gdb
11. By default, MacPorts installs gdb as ggdb in /opt/local/bin/
12. Now we have to create a certificate for code signing since we need gdb to have access to all things System, so create the certificate for gdb. There is no need for me to re-document this step you can review one of these two sites to see how to create this cert.
       see http://ntraft.com/installing-gdb-on-os-x-mavericks/
       or http://wiki.lazarus.freepascal.org/GDB_on_OS_X_Mavericks_and_Xcode_5

13. Once created, sign the certificate for gdb:

        $ sudo codesign -s gdbc /opt/local/bin/ggdb

14. SASM needs to be able to find gdb (ggdb) so you can create symlinks to accomplish this:

 $ sudo ln -s /opt/local/bin/ggdb /usr/local/bin/gdb

 $ sudo ln -s /opt/local/bin/ggdb /usr/bin/gdb

15. One last thing is that if you want to allow included files in the assembly code then add the "include" folder:

        - Right-click on SASM app, Show package contents -> MacOS/include

If you want the GAS and NASM assemblers to play completely nice within SASM, then you'll need to go the extra mile and install gcc as well. I use Xcode for GAS, and I just wanted to mess with NASM in SASM so this setup allows for that without installing gcc.

In the SASM preferences and build settings, if you use NASM, you'll want to change the "Assembly options" line to use "macho" instead of "elf32", and for starters remove the $MACRO.OBJ$ portion of the "Linking options."