Installing HLA Under Linux, Mac OSX, or FreeBSD (*NIX)

HLA is a compiler that translates source code into either object code or a lower-level assembly language that Gas (GNU's *as* assembler) must process. After compilation, you must link the object code output using a linker program such as the GNU ld linker. Typically, you will link the object code produced by one or more HLA source files with the HLA Standard Library (hlalib.a). Most of this activity takes place transparently whenever you ask HLA to compile your HLA source file(s). However, for the whole process to run smoothly, you must have installed HLA and all the support files correctly. This section will discuss how to set up HLA on your system.

These instructions assume that you are using the BASH command-line shell program. If you are using a different command-line shell interpreter, you may need to adjust some of the following instructions accordingly. Note that you can run the BASH interpreter from just about any command-line shell by typing "bash" at the command line.

Mac OSX users note: the terminal window, by default, does not run the BASH shell command interpreter. You should explicitly run BASH by typing "bash" at the command-line prompt when you open up a terminal window.

First, you will need an HLA distribution for Linux, Mac OSX, or FreeBSD. Please see Webster or the previous section if you're attempting to install HLA on a different OS such as Windows. The latest version of HLA is always available on Webster at http://webster.cs.ucr.edu. You should go there and download the latest version if you do not already possess it.

Under Linux, Mac OSX, and FreeBSD, HLA will produce a low-level assembly language output file that you can assemble using the Free Software Foundation's Gas assembler. The HLA package contains the HLA compiler, the HLA Standard Library, and a set of include files for the HLA Standard Library. If you write an HLA program want Gas to process it, you'll need to make sure you have a reasonable version of Gas available (Gas is available on most *NIX distributions, so this shouldn't be a problem). Note that the HLA Gas output can only be assembled by Gas v2.10 or later (so you will need the 2.10 or later binutils distribution).

Here's the steps I went through to install HLA on my Linux, Mac OSX, and FreeBSD systems:

- First, if you haven't already done so, download the HLA executables file (for Linux, Mac OSX, or FreeBSD) from Webster at http://webster.cs.ucr.edu. On Webster you can download several different tar.gz files associated with HLA from the HLA download page. The "Linux Executables", "Mac Executables", or "FreeBSD executables" is the only one you'll absolutely need; however, you'll probably want to grab the documentation and examples files as well. If you're curious, or you want some more example code, you can download the source listings to the HLA Standard Library. If you're *really* curious (or masochistic), you can download the HLA compiler source listings to (this is *not* for casual browsing!).
- I downloaded the linux.hla.tar.gz (for Linux), mac.hla.tar.gz (for Mac OSX), or bsd.hla.tar.gz (for FreeBSD) file for HLA v2.2 while writing this. Most likely, there is a much later version available as you're reading this. Be sure to get the latest version. I chose to download this file to my "/usr/hla" directory; you can put the file wherever you like, though this documentation assumes that all HLA files wind up in the "/usr/hla/..." directory tree. Note: the .tar.gz file downloads into /usr/hla. If you want the files placed somewhere else, unpack them to this directory and then move them.
- After downloading linux.hla.tar.gz, mac.hla.tar.gz, or bsd.hla.tar.gz to my root directory, I executed the following shell command: "gzip -d linux.hla.tar.gz" ("gzip –d bsd.hla.tar.gz" under FreeBSD; "gzip –d mac.hla.tar.gz" for Mac OSX). Once decompression was complete, I extracted the individual files using the command "tar xvf linux.hla.tar" ("tar xvf bsd.hla.tar" under FreeBSD, "tar xvf mac.hla.tar" under Mac OSX). This extracted several executable files (e.g., "hla" and "hlaparse") along with three subdirectories (include, hlalib, and hlalibsrc). The HLA program is a "shell" program that runs the HLA compiler (hlaparse), gas (as), the linker (ld), and other programs. You can think of hla as the "HLA Compiler". It would be a real good idea, at this point, to set the permissions on "hla" and "hlaparse" so that everyone can read and execute them. You should also set read and execute permissions on the two subdirectories and read permissions on all the files within the directories (if this isn't the default state). Do a "man chmod" from the Linux/Mac OSX/FreeBSD command-line if you don't know how to change permissions.
- If you prefer a more "Unix-like" environment, you could copy the hla and hlaparse (and other executable) files to the "/usr/bin" or "/usr/local/bin" subdirectory. This step, however, is optional

- Next, (logged in as a plain user rather than root or the super-user), I edited the ".bashrc" file in my home directory ("/home/rhyde" in my particular case, this will probably be different for you). I found the line that defined the "path" variable, it originally looked like this on my system:

  ```
  PATH=$DBROOT/bin:$DBROOT/pgm:$PATH
  ```

  I edited this line to add the path to the HLA directory, producing the following:

  ```
  PATH=$DBROOT/bin:$DBROOT/pgm:/usr/hla:$PATH
  ```

  Without this modification, *NIX will probably not find HLA when you attempt to execute it unless you type a full path (e.g., "/usr/hla/hla") when running the program. Since this is a pain, you'll definitely want to add "/usr/hla" to your path. Of course, if you've chosen to copy hla and hlaparse to the "/usr/bin" or "/usr/local/bin" directory, chances are good you won't have to change the path as it already contains these directories.

  Next, I added the following four lines to ".bashrc" (note that *NIX filenames beginning with a period don't normally show up in directory listings unless you supply the "-a" option to ls):

  ```
  hlalib=/usr/hla/hlalib
  export hlalib
  hlainc=/usr/hla/include
  export hlainc
  ```

  These four lines define (and export) environment variables that HLA needs during compilation. Without these environment variables, HLA will probably complain about not being able to find include files, or the linker (ld) will complain about strange undefined symbols when you attempt to compile your programs. Note that this step is optional if you leave the library and include files installed in the /usr/hla directory subtree.

- Optionally, you can add the following two lines to the .bashrc file (but make sure you've created the /tmp directory if you do this):

  ```
  hlatemp=/tmp
  export hlatemp
  ```

  After saving the ".bashrc" shell, you can tell *NIX to make the changes to the system by using the command:

  ```
  source .bashrc
  ```

  Note: this discussion only applies to users who run the BASH shell. If you are using a different shell (like the C-Shell or the Korn Shell), then the directions for setting the path and environment variables differs slightly. Please see the documentation for your particular shell if you don't know how to do this.
- At this point, HLA should be properly installed and ready to run. Try typing "HLA -?" at the command line prompt and verify that you get the HLA help message. If not, go back and figure out what you've done wrong up to this point (it doesn't hurt to start over from the beginning if you're lost).
- Now it's time to try your hand at writing an honest to goodness HLA program and verify that the whole system is working. Here's the canonical "Hello World" program written in HLA. Enter it into a text editor and save it using the filename "hw.hla":

```
program HelloWorld;
#include( "stdlib.hhf" )
begin HelloWorld;

    stdout.put( "Hello, World of Assembly Language", nl );

end HelloWorld;
```

- Make sure you're in the same directory containing the "hw.hla" file and type the following command at the prompt: "hla -v hw". The "-v" option tells HLA to produce VERBOSE output during compilation. This is helpful for determining what went wrong if the system fails somewhere along the line. This command should produce output like the following:

```
HLA (High Level Assembler)
Use '-license' to see licensing information.
Version 2.0 build 411 (prototype)
ELF output
GAS output
-test active

HLA Lib Path:     /usr/hla/hlalib/hlalib.a
HLA include path: /usr/hla/include
HLA temp path:
Files:
1: hw.hla

Compiling 'hw.hla' to 'hw.asm'
using command line:
[hlaparse -LINUX -level=high  -v -sg -test "hw.hla"]

----------------------
HLA (High Level Assembler) Parser
use '-license' to view license information
Version 2.0 build 411 (prototype)
-t active
File: hw.hla
Output Path: ""
hlainc Path: "/usr/hla/include"
hlaauxinc Path: "(null)"
Compiler running under Linux OS
Back-end assembler: GAS
Language Level: high

Compiling "hw.hla" to "hw.asm"
Compilation complete, 25444 lines,   0.122 seconds,  208557 lines/second
----------------------

Assembling "hw.asm" to "hw.o" via [as --32 -o hw.o   "hw.asm"]
Linking via [ld     -o "hw"   "hw.o" "/usr/hla/hlalib/hlalib.a"]
```

Versions of HLA may appear for other Operating Systems (beyond Windows, Linux, FreeBSD, and Mac OSX) as well. Check out Webster to see if any progress has been made in this direction. Note a unique thing about HLA: Carefully written (console) applications will compile and run on all supported operating systems without change. This is unheard of for assembly language! Therefore, if you are using multiple operating systems supported by HLA, you'll probably want to download files for all supported OSes.

Standard Configurations under Linux/FreeBSD/Mac OSX

HLA supports fewer configurations under Linux, FreeBSD, and Mac OSX than under Windows but this is primarily because the main tools available for *NIX (Linux/FreeBSD/MacOSX) are all freely distributable and there is no need to support commercial tools. There are three different ways to generate object code files and only one linker and one librarian option available under Linux. There is no resource compiler (that HLA would automatically use).

HLA can generate object files in one of two different ways under *NIX:

- The hlaparse program can generate an ELF object file directly.
- The hlaparse program can generate a Gas-compatible source file that the FSF Gas assembler can convert to an ELF file.

As this was being written, HLA under Mac OS X only generates Gas-compatible source files that the Gas assembler converta to Mach-o object files. Direct output of mach-o object files should appear in HLA v2.3.

Under *NIX you don't get a choice of linkers. Everyone uses the FSF/GNU ld (load) program as the standard system linker. The HLA package also uses ld. In a similar vein, your only librarian choice is the FSF/GNU ar (archive) program. These tools work great and they're freely distributable, so they're the perfect back ends to the HLA system.