**Assembling, Linking, and Debugging a 32-bit Program with a Batch File**

Many people like to use a *Windows batch file* to assemble and link programs. A batch file is a text file containing a sequence of commands that execute as if they had been typed at the command prompt. In fact, batch files are powerful enough to contain variables, loops, IF statements, and so on.

The easiest way to run a batch file is to first open a Command window and then type the name of the batch file (along with any arguments) at the command prompt. To open a Command window, you must execute a program named **cmd.exe**. We will make that step easier for you.

**Step 1:** Download the ZIP file named BatchFiles.zip, it contains the following items:

* **A shortcut to cmd.exe,** which opens a Command window in the current directory
* **asm32.bat**, a batch file for assembling and linking programs
* **main.asm**, a sample assembly language program

**Step 2:** Extract the ZIP file into the C:\Irvine\Examples directory on your computer.

**Step 3:** If you're running Windows, open the file with a text editor (such as NotePad++) and find this line:

set VS\_HOME=C:\Program Files\Microsoft Visual Studio 12.0

And change it to the correct path for Visual Studio 2017 Community:

set VS\_HOME=C:\Program Files (x86)\Microsoft Visual Studio 14.0

**Step 4:** Do the following:

* Copy asm32.bat to any directory on your system path. By doing this, you make it possible for MS-Windows to recognize **asm32** as a valid command when typed at the MS-Windows command prompt. (If you want to find out which directories are on the current system path, type **path** and press Enter at the system command prompt.)
* Double-click the shortcut to **cmd.exe**. A Command window should appear.
* At the command prompt in this window, type **asm32** and press Enter. This will execute the asm32 batch file and display help information.

|  |
| --- |
| This file assembles, links, and debugs a single assembly language  source file. Before using it, install Visual Studio 2017 Community in the following  directory:  C:\Program Files\Microsoft Visual Studio 14.0  Next, install the Irvine link libraries and include  files in the following directory: C:\Irvine  Finally, copy this batch file to a location on your system path.  We recommend the following directory:  C:\Program Files\Microsoft Visual Studio 14.0\VC\bin  Command-line syntax:  asm32 [/H | /h | -H | -h] -- display this help information  asm32 filelist -- assemble and link all files  asm32 /D filelist -- assemble, link, and debug  asm32 /C filelist -- assemble only  <filelist> is a list of up to 5 filenames (without extensions),  separated by spaces. The filenames are assumed to refer to files  having .asm extensions. Command-line switches are case-sensitive. |

Type the following command to assemble and link a source file named **main.asm**:

asm32 main

You should see the following messages:

|  |
| --- |
| Assembling: main.asm  The file main.obj was produced.  ..................................  Linking main.obj to the Irvine32, Kernel32, and User32 libraries.  The file main.exe was produced.  .................................. |

In fact, several files were produced.

* main.exe – the executable main program
* main.obj - the object file
* main.ilk - incremental link status file
* main.pdb - debug symbol file

If there were syntax errors in your program, you would see error messages generated by the assembler. Here is an example:

Assembling: main.asm  
main.asm(9) : error A2008: syntax error : myMessage  
main.asm(15) : error A2006: undefined symbol : myMessage

You would then open the main.asm file with a text editor (such as NotePad++), fix the errors, and run the asm32 batch file again.

|  |
| --- |
| Although we used a file named main.asm in this example, the asm32.bat batch file will work for any assembly language file, regardless of the name. The only requirement is that your assembly language source file have a **.asm** filename extension. |

**Assembling Programs in Other Directories**

No doubt, you will want to assemble programs in a variety of different disk folders, not just the folder used in the foregoing example. All you need to do is copy the **cmd.exe** shortcut we gave you to your working directory, where your assembly language source files are located. When you double-click to run the shortcut, it will open a Command window in the current folder.

**Assembling, Linking, and Debugging**

In addition to assembling and linking, you can use the asm32.bat file to launch your program in the Visual Studio debugger. Try the following command:

asm32 /D main

If the program assembles and links with no errors, your program should load in Visual Studio. The first time you do this with a new program, the source code will not appear. All you must do is press the **F10 key** to begin debugging, and your program should appear with a yellow band across the first executable line.

(Depending on how Visual Studio is configured, you might have to press F8 to do the same thing.)

From here, you can step through the program. When you get to the call to WriteString, you can even trace into its code by pressing the F11 key (trace to). When you finish, close Visual Studio.

From this time on, when you load the same program in the Visual Studio debugger, your source code will appear right away.

**Assembling without Linking**

Occasionally, you may want to assemble programs but not link them. This happens, for example, when you are creating a Multimodule project and you want to assemble each asm file into an obj file separately before linking them into the final exe program. Or, you might be assembling a module to be inserted into a link library (like Irvine32.lib).

To assemble a source file only, insert the /C option before the name of the file being assembled:

asm32 /C main

You should see the following output:

Assembling: main.asm

The file main.obj was produced.

..................................

If you are interested in learning more about how batch file commands work, here are some reference links we found:

* Microsoft TechNet article: [Creating Truly Powerful Batch Files](http://technet.microsoft.com/en-us/library/cc750054.aspx), by Brien Posey
* Microsoft TechNet article: [Using Batch Files in Windows NT](http://technet.microsoft.com/en-us/library/cc750056.aspx), by Troy Thompson

Links go out of date quickly, but you can do a web search for *Windows batch files* and get plenty of hits.