

Assembly Workspace Install Guide

Barak Gonen

This tutorial will guide you through installation of the assembly workspace and basic operations.

Tools:

Before we begin it should be pointed that assembly language is closely related to the hardware which is used for executing the commands. Assembly can be written for various CPU manufacturers (Intel, MIPS etc) and CPU register sizes (16 bit, 32 bit, 64 bit). On top of that, even for a given CPU and register size there are different assembly syntaxes, which slightly differ from each other... So we have to focus and make some choices. Our course will use a specific set of assembly, which is commonly used and can be easily adapted to other platforms:

- Intel x86 family CPU
- 32 bit register size
- FASM syntax

As editor, we shall use notepad++. If you prefer other editor that is possible, but notepad++ is very slim in size and loads very fast.

For debugger / disassembler we shall use IDA (Interactive DisAssembly).

To conclude, we shall need to download and install:

1. Notepad++, for writing our assembly code
2. FASM, for converting the assembly code to machine code (executable file)
3. IDA, for debugging / disassembling

Install Notepad++

Download notepad++ from the following link:

<https://notepad-plus-plus.org/downloads/>

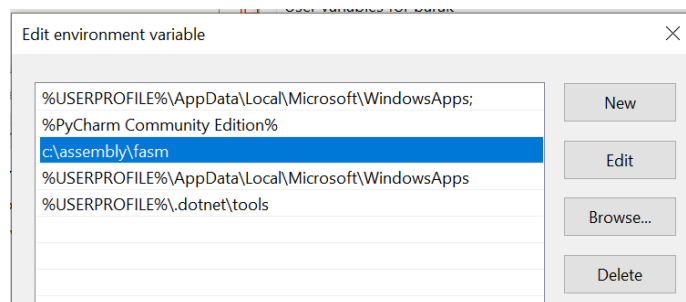
Make sure you select "English" while installing.

Install FASM

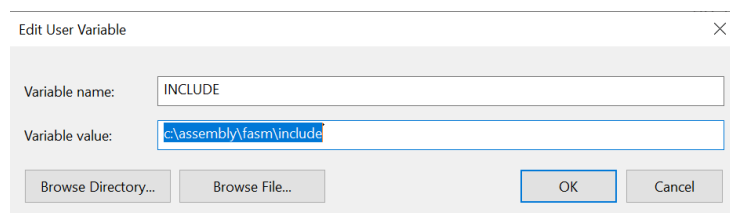
Installations can be downloaded from the following link:

<https://data.cyber.org.il/assembly/32bit/installs.zip>

- Copy the directory **assembly** to **C:** (or any other directory, as long as you change the next steps accordingly)
- Edit environment variables (right click **My Computer**, **Properties**, **Advanced System Settings**)
 - To the end of either user variable or system variable **PATH** add **;C:\assembly\asm** (note the semicolon is a must, it separates this entry from the previous one)



- Add a new user variable or system variable called **INCLUDE** that contains **C:\assembly\asm\include**



Dedicated Files

For our course, we shall use two dedicated files which can be found in the “assembly” directory which you have downloaded:

- **example.asm** is an example file which shall be used to demonstrate basic workspace operations.
- **training.inc** is an include file, which adds input/output capabilities to our assembly code.

Install IDA

<https://hex-rays.com/ida-free/>

Make sure you download the free version of IDA and not the pro, since the pro will allow only a short period of free operation before it gets locked.

Guided Exercise

We shall learn to conduct some basic operations

1. Open the file example.asm with notepad++

```
example.asm
1 include 'win32a.inc'
2
3 format PE console
4 entry start
5
6 section '.data' data readable writeable
7     hi    db  'hi',13,10,0
8     bye   db  'bye',13,10,0
9
10 section '.text' code readable executable
11 ; =====
12
13 start:
14     mov    eax, 1
15     mov    ebx, 2
16     add    eax, ebx
17     ; Show a message to the user:
18     mov    esi, hi
19     call    print_str
20     ; Show a message to the user:
21     mov    esi, bye
22     call    print_str
23
24 ; =====
25
26     push    0
27     call    [ExitProcess]
28
29 include 'training.inc'
30
```

2. Use FASM to convert the assembly code into machine language, by writing "fasm example.asm" in the command line. An executable file will be created – example.exe. Run it simply by writing "example", as in the following screenshot:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\barak>cd c:\assembly

c:\Assembly>fasm example.asm
flat assembler  version 1.71.22  (1048576 kilobytes memory)
3 passes, 0.2 seconds, 2048 bytes.

c:\Assembly>example
hi
bye

c:\Assembly>
```

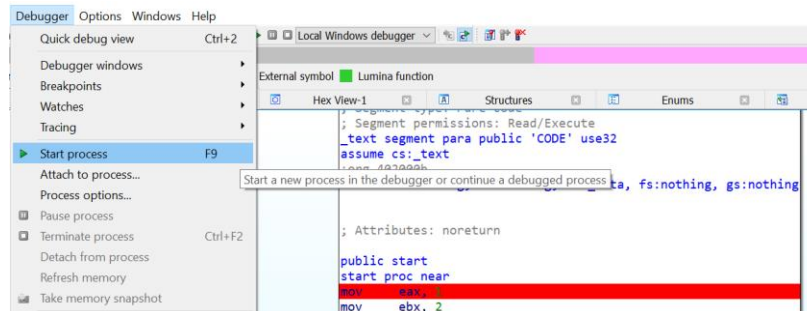
3. Use IDA to open the executable file:
 - a. Open the file (file->open, then select example.exe from the working directory). IDA will automatically find the CPU type, so just approve by clicking "OK".
 - b. Once opened you will see the following:

```
public start
start proc near
mov     eax, 1
mov     ebx, 2
add     eax, ebx
mov     esi, offset aHi ; "hi\r\n"
call    sub_4020CF
mov     esi, offset aBye ; "bye\r\n"
call    sub_4020CF
push    0                ; uExitCode
call    ds:ExitProcess
start endp
```

- c. While your cursor is on the first line, press F2 (or Fn+F2, on some laptops). This will add a breakpoint to the selected line

```
public start
start proc near
mov     eax, 1
mov     ebx, 2
add     eax, ebx
```

- d. Start a new process. IDA will issue a warning to make sure you trust the code. Click "OK".



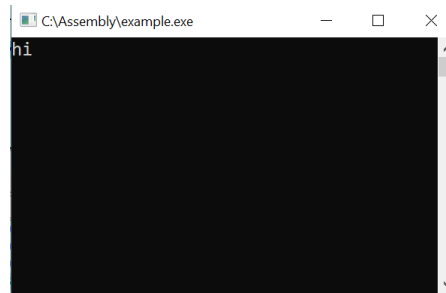
- e. The view will change to debug view with various windows. We shall learn about their contents on our course.

```
.text:00402000 ; Attributes: noreturn
.text:00402000
.text:00402000 public start
.text:00402000 start proc near
.text:00402000 mov     eax, 1
.text:00402005 mov     ebx, 2
.text:0040200A add     eax, ebx
.text:0040200C mov     esi, offset aHi ; "hi\r\n"
.text:00402011 call    sub_4020CF
.text:00402016 mov     esi, offset aBye ; "bye\r\n"
.text:0040201B call    sub_4020CF
.text:00402020 push    0                ; uExitCode
.text:00402022 call    ds:ExitProcess
.text:00402022 start endp
.text:00402022
```

- f. Using F8, advance one instruction. Repeat 5 times until you reach this line:

```
.text:00402000 ; Attributes: noreturn
.text:00402000
.text:00402000 public start
.text:00402000 start proc near
.text:00402000 mov     eax, 1
.text:00402005 mov     ebx, 2
.text:0040200A add     eax, ebx
.text:0040200C mov     esi, offset aHi ; "hi\r\n"
.text:00402011 call    sub_4020CF
.text:00402016 mov     esi, offset aBye ; "bye\r\n"
.text:0040201B call    sub_4020CF
.text:00402020 push    0 ; uExitCode
.text:00402022 call    ds:ExitProcess
.text:00402022 start endp
.text:00402022
```

- g. Look for the command window, where you will see "hi" printed:



Now that you have completed the installation and basic operation of the workspace, you are good to proceed to the assembly training and tasks 😊 Good luck!