

## Lab 8: Using Jasmin to run x86 Assembly Code

**Course Name:** Malware Analysis and Reverse Engineering (IAM302)

**Student Name:** Nguyễn Trần Vinh – SE160258

**Instructor Name:** Mai Hoàng Đình

**Lab Due Date:** 15/2/2023

### Purpose

To practice writing and running basic x86 assembly code, using the Jasmin interpreter

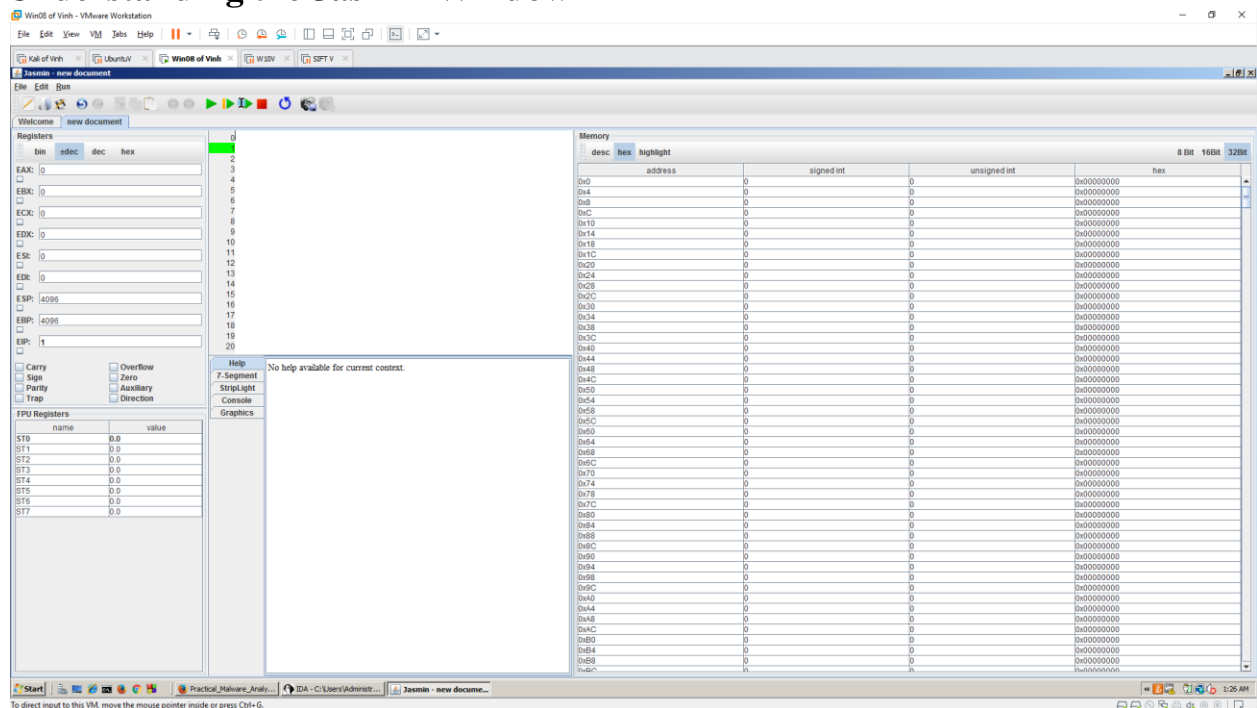
### What You Need for This Project

- Any computer, running any OS

### Install Java

### Download Jasmin

### Understanding the Jasmin Window



### Registers

Registers	
	<div>bin±decdec hex</div>
EAX:	<input type="text" value="0"/>
<input type="checkbox"/>	
EBX:	<input type="text" value="0"/>
<input type="checkbox"/>	
ECX:	<input type="text" value="0"/>
<input type="checkbox"/>	
EDX:	<input type="text" value="0"/>
<input type="checkbox"/>	
ESI:	<input type="text" value="0"/>
<input type="checkbox"/>	
EDI:	<input type="text" value="0"/>
<input type="checkbox"/>	
ESP:	<input type="text" value="4096"/>
<input type="checkbox"/>	
EBP:	<input type="text" value="4096"/>
<input type="checkbox"/>	
EIP:	<input type="text" value="1"/>
<input type="checkbox"/>	

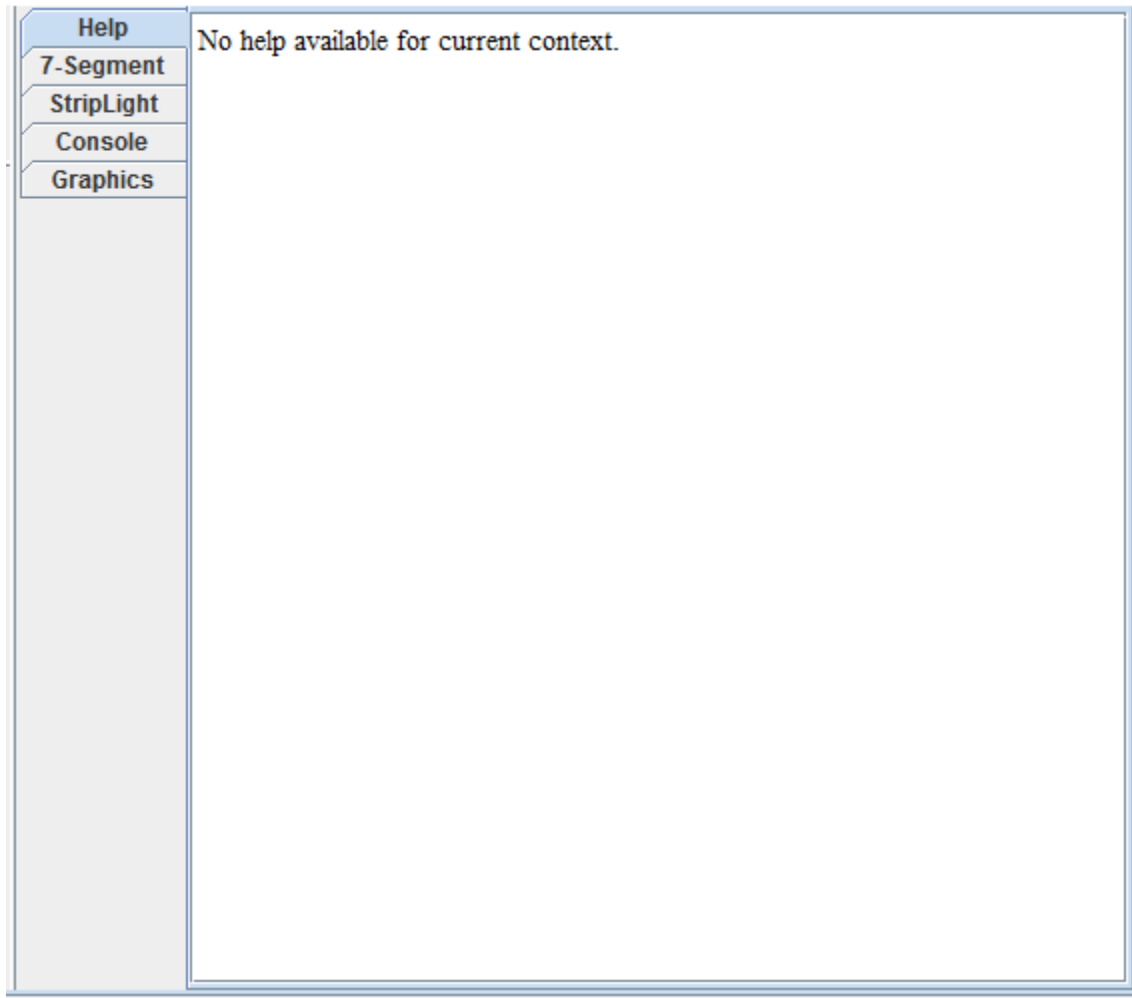
## Flags

<input type="checkbox"/> Carry	<input type="checkbox"/> Overflow
<input type="checkbox"/> Sign	<input type="checkbox"/> Zero
<input type="checkbox"/> Parity	<input type="checkbox"/> Auxiliary
<input type="checkbox"/> Trap	<input type="checkbox"/> Direction

## Code

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

**Help**



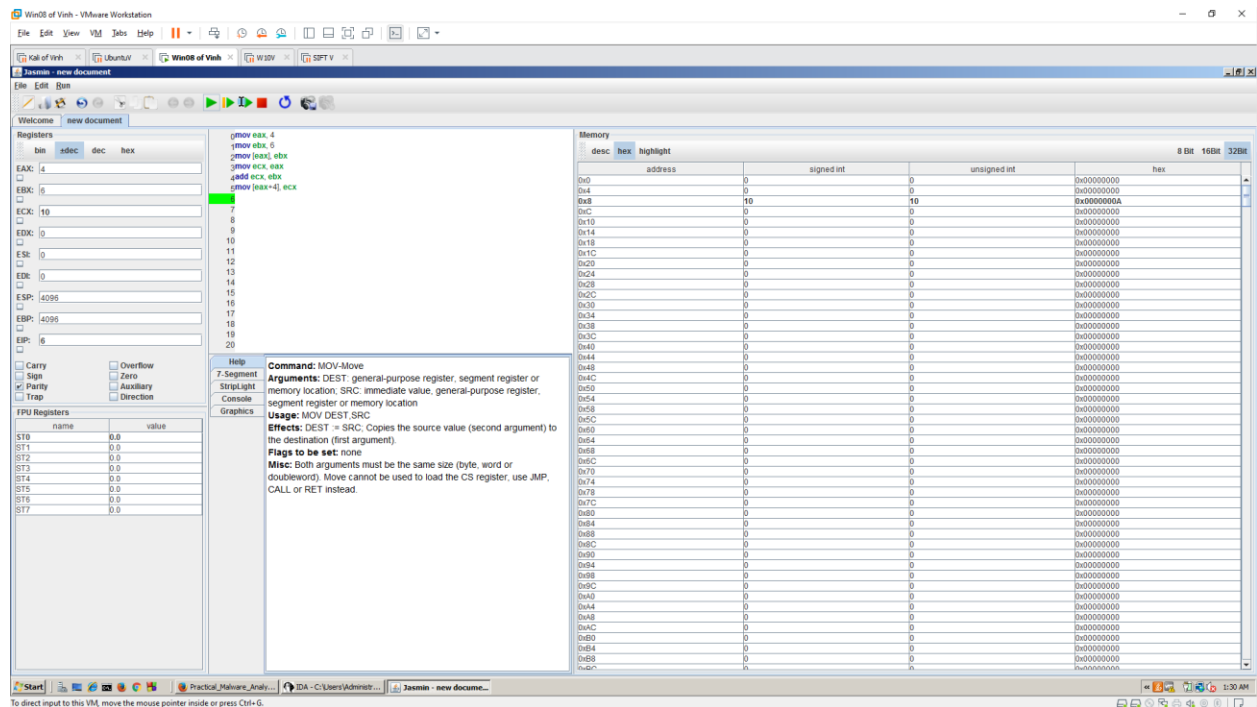
## Memory

Memory				
desc	hex	highlight	8 Bit 16Bit 32Bit	
address	signed int	unsigned int	hex	
0x0	0	0	0x00000000	
0x4	0	0	0x00000000	
0x8	0	0	0x00000000	
0xC	0	0	0x00000000	
0x10	0	0	0x00000000	
0x14	0	0	0x00000000	
0x18	0	0	0x00000000	
0x1C	0	0	0x00000000	
0x20	0	0	0x00000000	
0x24	0	0	0x00000000	
0x28	0	0	0x00000000	
0x2C	0	0	0x00000000	
0x30	0	0	0x00000000	
0x34	0	0	0x00000000	
0x38	0	0	0x00000000	
0x3C	0	0	0x00000000	
0x40	0	0	0x00000000	
0x44	0	0	0x00000000	
0x48	0	0	0x00000000	
0x4C	0	0	0x00000000	
0x50	0	0	0x00000000	
0x54	0	0	0x00000000	
0x58	0	0	0x00000000	
0x5C	0	0	0x00000000	
0x60	0	0	0x00000000	
0x64	0	0	0x00000000	
0x68	0	0	0x00000000	
0x6C	0	0	0x00000000	
0x70	0	0	0x00000000	
0x74	0	0	0x00000000	
0x78	0	0	0x00000000	
0x7C	0	0	0x00000000	
0x80	0	0	0x00000000	
0x84	0	0	0x00000000	
0x88	0	0	0x00000000	
0x8C	0	0	0x00000000	
0x90	0	0	0x00000000	
0x94	0	0	0x00000000	
0x98	0	0	0x00000000	
0x9C	0	0	0x00000000	
0xA0	0	0	0x00000000	
0xA4	0	0	0x00000000	
0xA8	0	0	0x00000000	
0xAC	0	0	0x00000000	
0xB0	0	0	0x00000000	
0xB4	0	0	0x00000000	
0xB8	0	0	0x00000000	
0xBC	0	0	0x00000000	

# Using mov Instructions

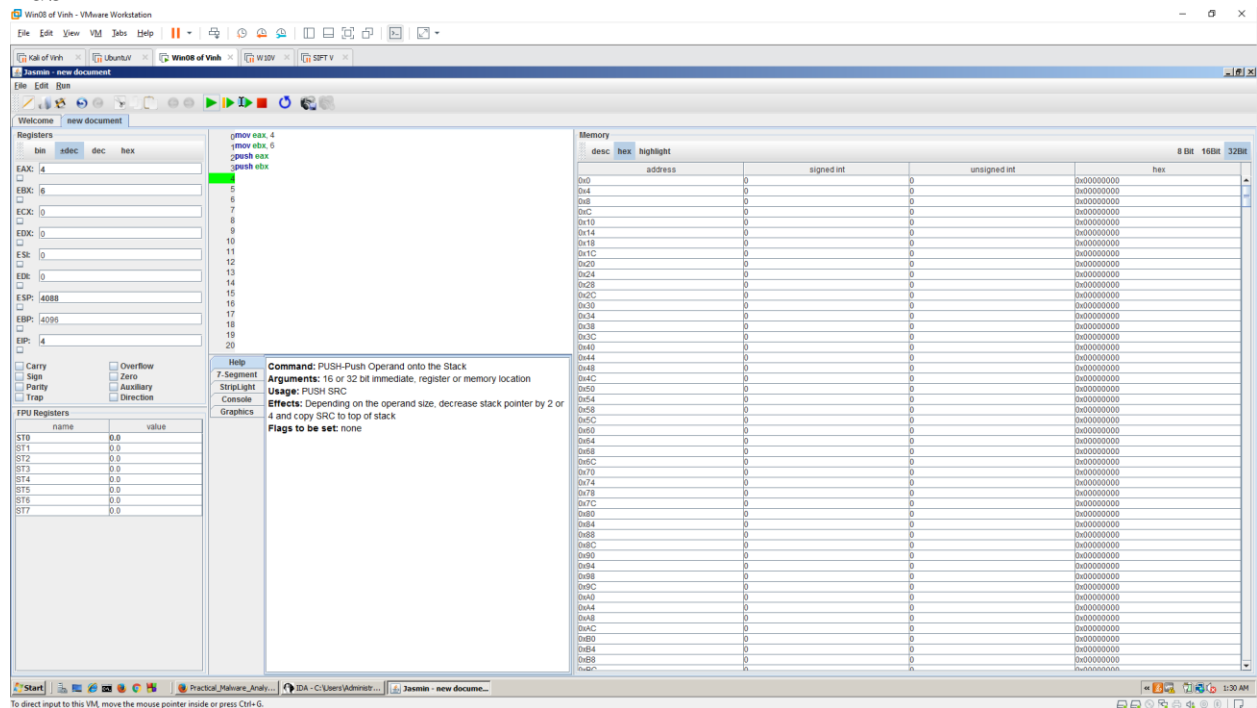


## Storing Results in Memory



## Using the Stack

## Push



## Pop

Win08 of Vmsh - VMware Workstation

File Edit View VM Tools Help

Kali of Vmsh Ubuntu Vmsh Win08 of Vmsh W32V SPT V

new document

Registers

bin adoc dec hex

EAX: 4

EBX: 6

ECX: 6

EDX: 0

ESI: 0

EDI: 0

ESP: 4092

EBP: 4096

EBX: 5

Carry Sign Parity Trap

Overflow Zero Auxiliary Direction

FPU Registers

name	value
ST0	0.0
ST1	0.0
ST2	0.0
ST3	0.0
ST4	0.0
ST5	0.0
ST6	0.0
ST7	0.0

Help

7 Segment StripLight register Console Graphics

Command: POP-POP a Word from the Stack

Arguments: DEST 16 or 32 bit memory location or register or segment

Usage: POP DEST

Effects: POP stores the contents of DEST with the word on the top of the 80386 stack, addressed by SS:SP (address-size attribute of 16 bits) or SS:ESP (address-size attribute of 32 bits). The stack pointer is then increased accordingly.

Flags to be set: none

Memory

desc	hex	highlight	address	signed int	unsigned int	hex
0x0	0	0	0x00000000	0	0	0x00000000
0x4	0	0	0x00000000	0	0	0x00000000
0x8	0	0	0x00000000	0	0	0x00000000
0xC	0	0	0x00000000	0	0	0x00000000
0x10	0	0	0x00000000	0	0	0x00000000
0x14	0	0	0x00000000	0	0	0x00000000
0x18	0	0	0x00000000	0	0	0x00000000
0x1C	0	0	0x00000000	0	0	0x00000000
0x20	0	0	0x00000000	0	0	0x00000000
0x24	0	0	0x00000000	0	0	0x00000000
0x28	0	0	0x00000000	0	0	0x00000000
0x2C	0	0	0x00000000	0	0	0x00000000
0x30	0	0	0x00000000	0	0	0x00000000
0x34	0	0	0x00000000	0	0	0x00000000
0x38	0	0	0x00000000	0	0	0x00000000
0x3C	0	0	0x00000000	0	0	0x00000000
0x40	0	0	0x00000000	0	0	0x00000000
0x44	0	0	0x00000000	0	0	0x00000000
0x48	0	0	0x00000000	0	0	0x00000000
0x4C	0	0	0x00000000	0	0	0x00000000
0x50	0	0	0x00000000	0	0	0x00000000
0x54	0	0	0x00000000	0	0	0x00000000
0x58	0	0	0x00000000	0	0	0x00000000
0x5C	0	0	0x00000000	0	0	0x00000000
0x60	0	0	0x00000000	0	0	0x00000000
0x64	0	0	0x00000000	0	0	0x00000000
0x68	0	0	0x00000000	0	0	0x00000000
0x6C	0	0	0x00000000	0	0	0x00000000
0x70	0	0	0x00000000	0	0	0x00000000
0x74	0	0	0x00000000	0	0	0x00000000
0x78	0	0	0x00000000	0	0	0x00000000
0x7C	0	0	0x00000000	0	0	0x00000000
0x80	0	0	0x00000000	0	0	0x00000000
0x84	0	0	0x00000000	0	0	0x00000000
0x88	0	0	0x00000000	0	0	0x00000000
0x8C	0	0	0x00000000	0	0	0x00000000
0x90	0	0	0x00000000	0	0	0x00000000
0x94	0	0	0x00000000	0	0	0x00000000
0x98	0	0	0x00000000	0	0	0x00000000
0x9C	0	0	0x00000000	0	0	0x00000000
0xA0	0	0	0x00000000	0	0	0x00000000
0xA4	0	0	0x00000000	0	0	0x00000000
0xA8	0	0	0x00000000	0	0	0x00000000
0xAC	0	0	0x00000000	0	0	0x00000000
0xB0	0	0	0x00000000	0	0	0x00000000
0xB4	0	0	0x00000000	0	0	0x00000000
0xB8	0	0	0x00000000	0	0	0x00000000
0xBC	0	0	0x00000000	0	0	0x00000000

Start

Practical\_Malware\_Analy...

IDA - C:\Users\Admini...

Jasmin - new docume...

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

## Reversing a Sequence

Win08 of Vmsh - VMware Workstation

File Edit View VM Tools Help

Kali of Vmsh Ubuntu Vmsh Win08 of Vmsh W32V SPT V

new document

Registers

bin adoc dec hex

EAX: 4

EBX: 3

ECX: 2

EDX: 1

ESI: 0

EDI: 0

ESP: 4096

EBP: 4096

EBX: 12

Carry Sign Parity Trap

Overflow Zero Auxiliary Direction

FPU Registers

name	value
ST0	0.0
ST1	0.0
ST2	0.0
ST3	0.0
ST4	0.0
ST5	0.0
ST6	0.0
ST7	0.0

Help

7 Segment StripLight register Console Graphics

Command: POP-POP a Word from the Stack

Arguments: DEST 16 or 32 bit memory location or register or segment

Usage: POP DEST

Effects: POP stores the contents of DEST with the word on the top of the 80386 stack, addressed by SS:SP (address-size attribute of 16 bits) or SS:ESP (address-size attribute of 32 bits). The stack pointer is then increased accordingly.

Flags to be set: none

Memory

desc	hex	highlight	address	signed int	unsigned int	hex
0x0	0	0	0x00000000	0	0	0x00000000
0x4	0	0	0x00000000	0	0	0x00000000
0x8	0	0	0x00000000	0	0	0x00000000
0xC	0	0	0x00000000	0	0	0x00000000
0x10	0	0	0x00000000	0	0	0x00000000
0x14	0	0	0x00000000	0	0	0x00000000
0x18	0	0	0x00000000	0	0	0x00000000
0x1C	0	0	0x00000000	0	0	0x00000000
0x20	0	0	0x00000000	0	0	0x00000000
0x24	0	0	0x00000000	0	0	0x00000000
0x28	0	0	0x00000000	0	0	0x00000000
0x2C	0	0	0x00000000	0	0	0x00000000
0x30	0	0	0x00000000	0	0	0x00000000
0x34	0	0	0x00000000	0	0	0x00000000
0x38	0	0	0x00000000	0	0	0x00000000
0x3C	0	0	0x00000000	0	0	0x00000000
0x40	0	0	0x00000000	0	0	0x00000000
0x44	0	0	0x00000000	0	0	0x00000000
0x48	0	0	0x00000000	0	0	0x00000000
0x4C	0	0	0x00000000	0	0	0x00000000
0x50	0	0	0x00000000	0	0	0x00000000
0x54	0	0	0x00000000	0	0	0x00000000
0x58	0	0	0x00000000	0	0	0x00000000
0x5C	0	0	0x00000000	0	0	0x00000000
0x60	0	0	0x00000000	0	0	0x00000000
0x64	0	0	0x00000000	0	0	0x00000000
0x68	0	0	0x00000000	0	0	0x00000000
0x6C	0	0	0x00000000	0	0	0x00000000
0x70	0	0	0x00000000	0	0	0x00000000
0x74	0	0	0x00000000	0	0	0x00000000
0x78	0	0	0x00000000	0	0	0x00000000
0x7C	0	0	0x00000000	0	0	0x00000000
0x80	0	0	0x00000000	0	0	0x00000000
0x84	0	0	0x00000000	0	0	0x00000000
0x88	0	0	0x00000000	0	0	0x00000000
0x8C	0	0	0x00000000	0	0	0x00000000
0x90	0	0	0x00000000	0	0	0x00000000
0x94	0	0	0x00000000	0	0	0x00000000
0x98	0	0	0x00000000	0	0	0x00000000
0x9C	0	0	0x00000000	0	0	0x00000000
0xA0	0	0	0x00000000	0	0	0x00000000
0xA4	0	0	0x00000000	0	0	0x00000000
0xA8	0	0	0x00000000	0	0	0x00000000
0xAC	0	0	0x00000000	0	0	0x00000000
0xB0	0	0	0x00000000	0	0	0x00000000
0xB4	0	0	0x00000000	0	0	0x00000000
0xB8	0	0	0x00000000	0	0	0x00000000
0xBC	0	0	0x00000000	0	0	0x00000000

Start

Practical\_Malware\_Analy...

IDA - C:\Users\Admini...

Jasmin - new docume...

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.