

Introduction to Malware Analysis

Assignment 4 – A Crash Course in X86 Disassembly

10 points

LAB – ASSEMBLY CODE

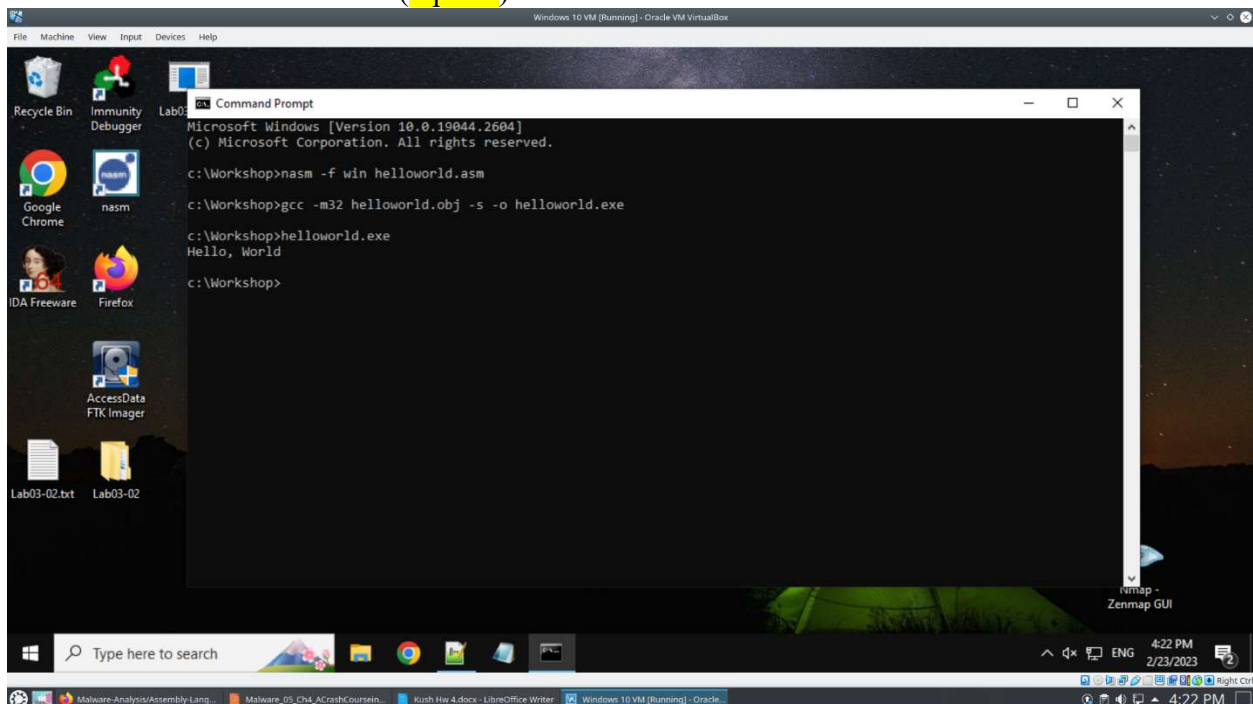
Read [this tutorial](#) on assembly language if needed.

Please note: If students copy and paste the assembly code from the slides, it may contain invisible formatting symbols and the compilation may fail.

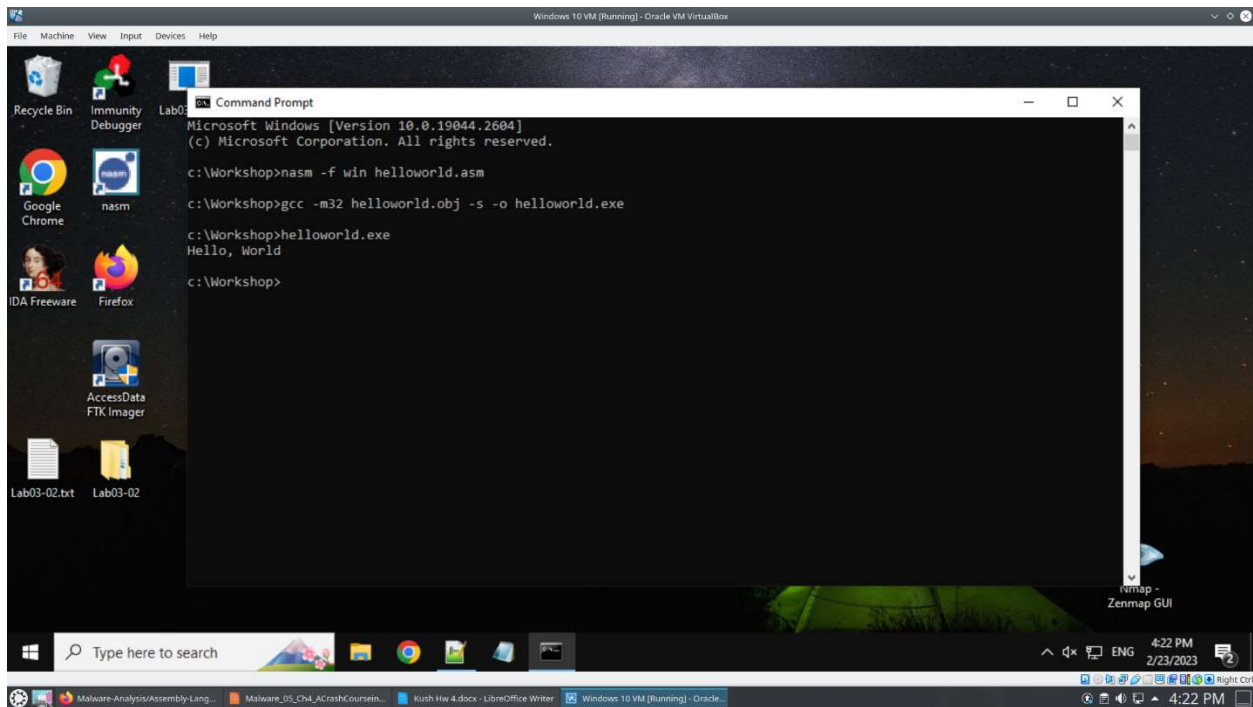
1. Copy the code in helloWorld.asm in the slides.
 - a. Please give a note (comment, denoted by a semicolon ;) on each instruction (each line) explaining what it does and paste the code with the notes below. (2 points)

1. extern _printf	;functions from C library
2. section .text	; section called .text
3. _main:	; functions
4. push message	;push message onto stack
5. call _printf	;print function is called
6. add esp, 4	;add 4 to esp
7. Ret	;return
8. message:	;section called message
9. db 'Hello, World', 10, 0	;string Hello World

- b. Provide a screenshot of the commands compiling the code and linking to create the executable. (1 point)



- c. Provide a screenshot of the results running the executable. (1 point)

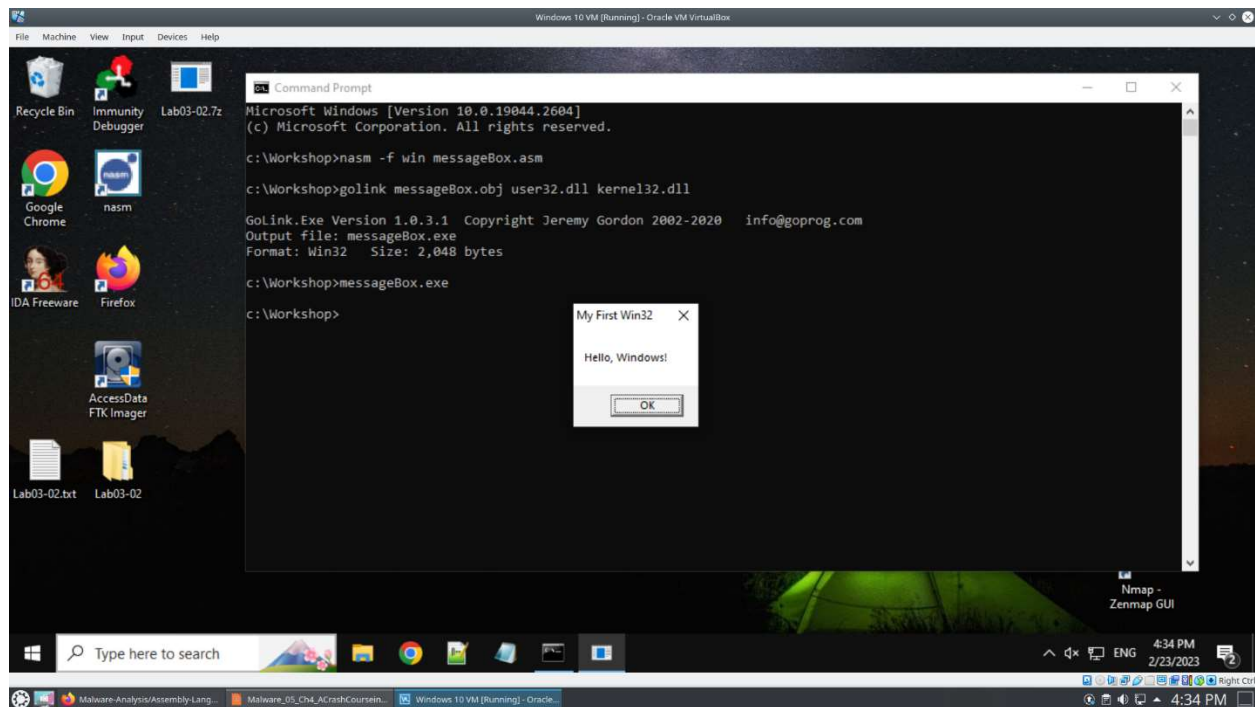


2. Copy the code in messageBox.asm in the slides.
- Please give a note (comment, denoted by a semicolon ;) on each instruction (each line) explaining what it does and paste the code with the notes below. (1 point)

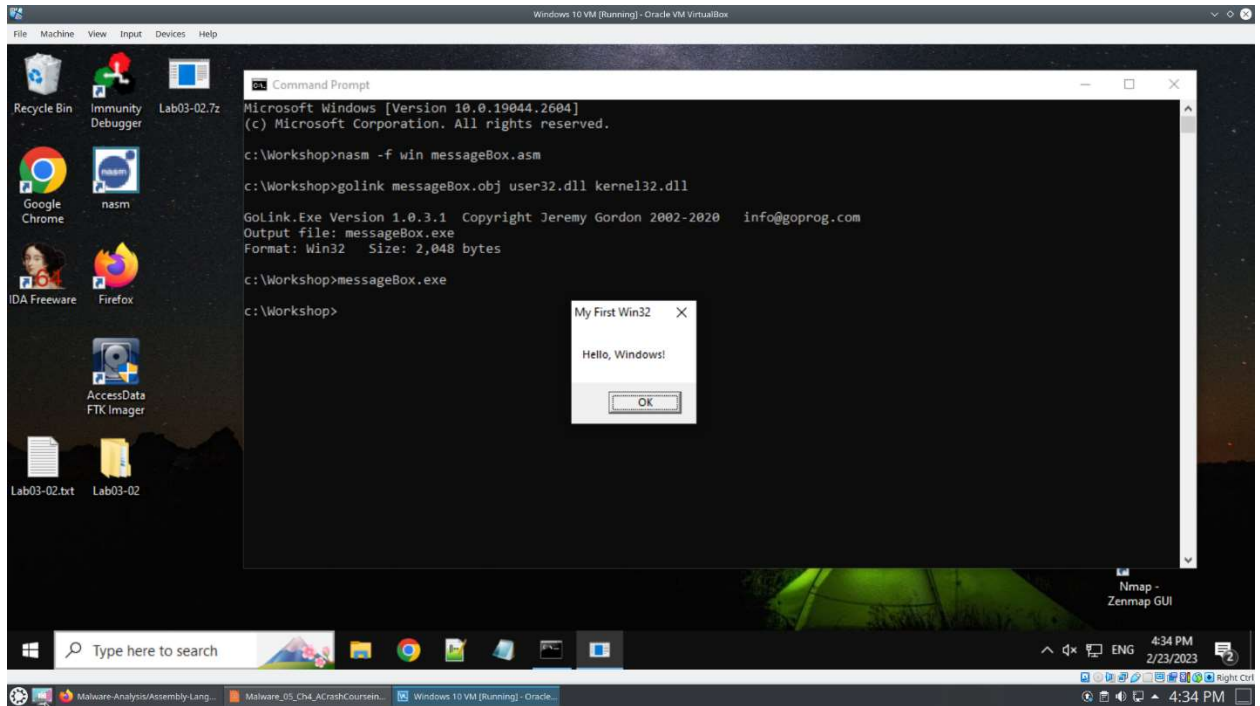
1. global start	
2. NULL equ 0	;equation set to 0
3. MB_OK equ 0	
4. extern MessageBoxA ;from user32	;function from user32
5. extern ExitProcess ;from kernel32	;function from kernel32
6. section .data	;section called .data
7. hello: db 'Hello, Windows!',0	; string Hello, Windows!
8. title: db 'My First Win32',0	;string My First Win32
9. section .text	;section called .text
10. start:	;start functions
11. push MB_OK	;push MB_OK onto stack
12. push title	;push title onto stack
13. push hello	;push hello onto stack

- | | | |
|-----|------------------|-----------------------------|
| 14. | push NULL | ;push NULL onto stack |
| 15. | call MessageBoxA | ;call function MessageBoxA |
| 16. | push 0 | ;push 0 onto stack |
| 17. | call ExitProcess | ;call functions ExitProcess |

- b. Provide a screenshot of the commands compiling the code and linking to create the executable. (1 point)



- c. Provide a screenshot of the results running the executable. (1 point)



3. Copy the code in exitProcess.asm in the slides.
- Please give a note (comment, denoted by a semicolon ;) on each instruction (each line) explaining what it does and paste the code with the notes below. (1 point)

- | | |
|-------------------------------------|---|
| 1. global start | |
| 2. extern printf | ;printf function from msvcrt |
| 3. extern scanf | ;scanf function from msvcrt |
| 4. extern ExitProcess | ;ExitProcess functionsfrom kernel32 |
| 5. section .bss | ; section called .bss |
| 6. name: resb 100 | |
| 7. section .data | ;section called .data |
| 8. prompt: db 'Enter your name: ',0 | ;prompts you to enter name |
| 9. frmt: db '%s',0 | ;format of something |
| 10. greet: db 'Hello, %s!',0ah,0 | ;program greets you after entering name |
| 11. section .text | ;section called .text |
| 12. start: | ;start function |
| 13. push prompt | ;push prompt onto stack |
| 14. call printf | ;called printf function |
| 15. add esp,4 | ;4 is added to esp |

16.	push	name	;push name onto stack
17.	push	frmt	;push frmt onto stack
18.	call	scanf	;calls scanf function
19.	add	esp,8	; 8 is added to esp
20.	push	name	;push name onto stack
21.	push	greet	;push greet onto stack
22.	call	printf	;calls printf onto stack
23.	add	esp,8	; 8 is added to esp
24.	push	0	; 0 is pushed onto stack
25.	call	ExitProcess	; calls ExitProcess function

- b. Provide a screenshot of the commands compiling the code and linking to create the executable. (1 point)

The screenshot shows a Windows 10 virtual machine environment. A Command Prompt window is open, displaying the following commands and output:

```

c:\Workshop>nasm -f win exitProcess.asm
c:\Workshop>golink /console exitProcess.obj msvcrt.dll kernel32.dll

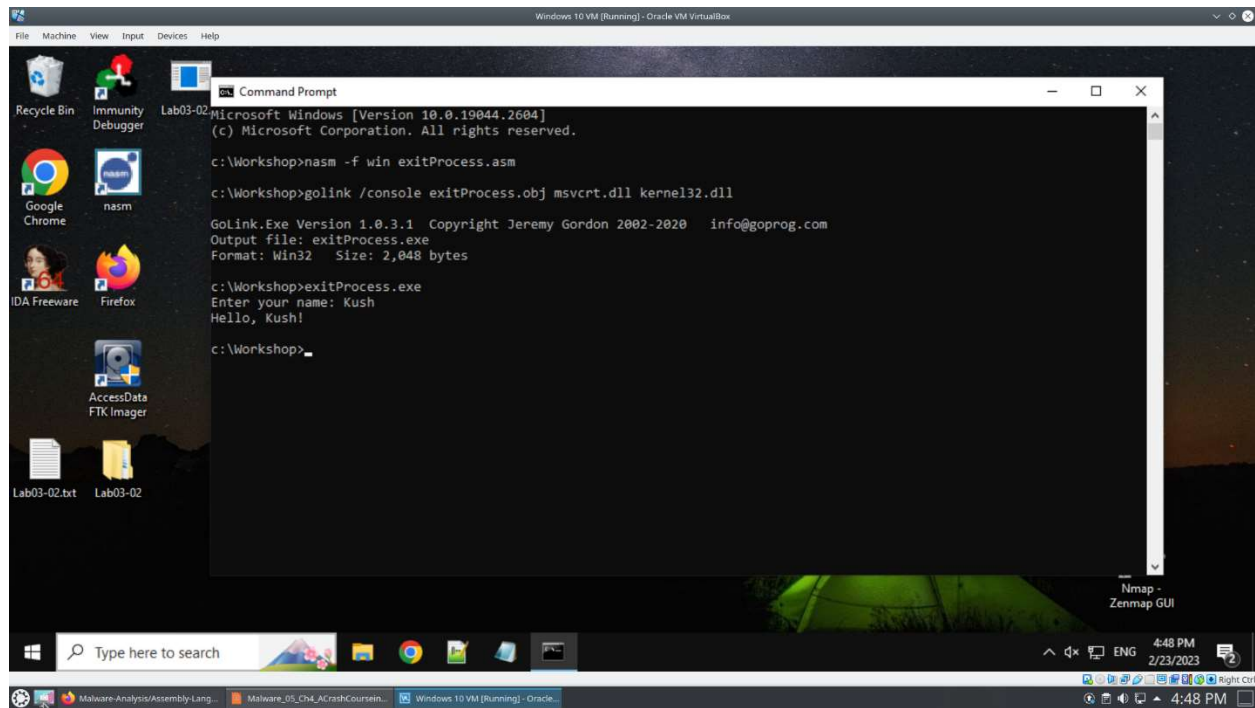
GoLink.Exe Version 1.0.3.1 Copyright Jeremy Gordon 2002-2020 info@goprog.com
Output file: exitProcess.exe
Format: Win32 Size: 2,048 bytes

c:\Workshop>exitProcess.exe
Enter your name: Kush
Hello, Kush!

c:\Workshop>
  
```

The background shows a desktop with various icons including Recycle Bin, Immunity Debugger, Google Chrome, nasm, IDA Freeware, Firefox, AccessData FTK Imager, and Lab03-02.txt. The taskbar at the bottom shows the Windows Start button, a search bar, and several open applications including Malware Analysis/Assembly Lang... and Windows 10 VM (Running) - Oracle VM VirtualBox.

- c. Provide a screenshot of the results running the executable. (1 point)



```
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

c:\Workshop>nasm -f win exitProcess.asm
c:\Workshop>golink /console exitProcess.obj msvcrt.dll kernel32.dll

GoLink.Exe Version 1.0.3.1 Copyright Jeremy Gordon 2002-2020 info@goprog.com
Output file: exitProcess.exe
Format: Win32 Size: 2,048 bytes

c:\Workshop>exitProcess.exe
Enter your name: Kush
Hello, Kush!

c:\Workshop>
```