|  |  |  |
| --- | --- | --- |
| Week | Reverse Engineering Malware | Duration |
| 3 | Data Transfers, Addressing, and Arithmetic | 120 mins |

Marks allocation: 8/100 for CA practical submission

**Lesson Objectives**

* Write and execute assembly language programs using Arrays and Loops

1. Integer Expression Calculation

Using the AddTwo program as a reference, write a program that calculates the following expression, using registers: A = (A + B) − (C + D). Assign integer values to the EAX, EBX, ECX, and EDX registers.

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

.code

main PROC

mov eax, 1h

mov ebx, 2h

mov ecx, 5h

mov edx, 6h

add eax, ebx

add ecx, edx

sub eax, ecx

INVOKE ExitProcess,0

main ENDP

END main

2. Exchanging Pairs of Array Values

Write a program with a loop and indexed addressing that exchanges every pair of values in an array with an even number of elements. Therefore, item I will exchange with item i+1, and item i+2 will exchange with item i+3, and so on.

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

.data

array DWORD 1h,2h,3h,4h,5h,6h,7h,8h

.code

main PROC

mov esi, OFFSET array

mov ecx, LENGTHOF array / 2

Loops:

mov eax, [esi]

mov ebx, [esi + 4]

mov [esi], ebx

mov [esi + 4], eax

add esi, TYPE array \* 2

loop Loops

INVOKE ExitProcess,0

main ENDP

END main

3. Copying a Word Array to a DoubleWord array

Write a program that uses a loop to copy all the elements from an unsigned Word (16-bit) array into an unsigned doubleword (32-bit) array.

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

.data

array WORD 1h,2h,3h,4h

newArray DWORD LENGTHOF array DUP(?)

.code

main PROC

mov ecx, LENGTHOF array

mov ESI, 0

L1:

movzx eax, array[esi]

mov newArray[esi\*2], eax

add esi, 2

Loop L1

INVOKE ExitProcess,0

main ENDP

END main

END