Bob Loth

Jeremy Satram

Baylee Bobele

ADDR pushes the offset of a parameter to the stack following an invoke keyword. Not using the invoke keyword, as in the case of count, will just push the actual value. Values are pushed in reverse order, so that the last value or offset after invoke will be the bottom value on the stack, and will be popped last.

INVOKE PromptForIntegers,

ADDR prompt1,

ADDR array,

Count

00000005 6A 03 \* push +000000003h

00000007 68 00000036 R \* push OFFSET array

0000000C 68 00000000 R \* push OFFSET prompt1

00000011 E8 00000000 E \* call PromptForIntegers

The function prototypes are contained in this listing file, but the functions’ implementation instructions are contained in another file.

C PromptForIntegers PROTO,

C ptrPrompt:PTR BYTE, ; prompt string

C ptrArray:PTR DWORD, ; points to the array

C arraySize:DWORD ; size of the array

Function calls have an ‘E’ following where the memory location and the bytes used values are. This seems to indicate the program is looking for the implementation in a different location.

00000027 FF 35 00000042 R \* push sum

0000002D 68 00000019 R \* push OFFSET prompt2

00000032 E8 00000000 E \* call DisplaySum

The include file’s contents are inserted in the listing file, just above the instructions for main.

PromptForIntegers PROTO,

C ptrPrompt:PTR BYTE, ; prompt string

C ptrArray:PTR DWORD, ; points to the array

C arraySize:DWORD ; size of the array

C

C ArraySum PROTO,

C ptrArray:PTR DWORD, ; points to the array

C count:DWORD ; size of the array

C

C DisplaySum PROTO,

C ptrPrompt:PTR BYTE, ; prompt string

C theSum:DWORD ; sum of the array

C

All of the implementations for the functions that are used in main are contained in their respective .asm files. They all include sum.inc.