

Kevin Sekuj
934-333-648
1/21/2021
Project 2
CS-271_400_W2021

Part 1 – Initial Setup

1.

EAX = FFFFFFFF

2.

CY/Carry = 0

OV/Overflow = 0

ZR/Zero = 0

PL/Sign = 0

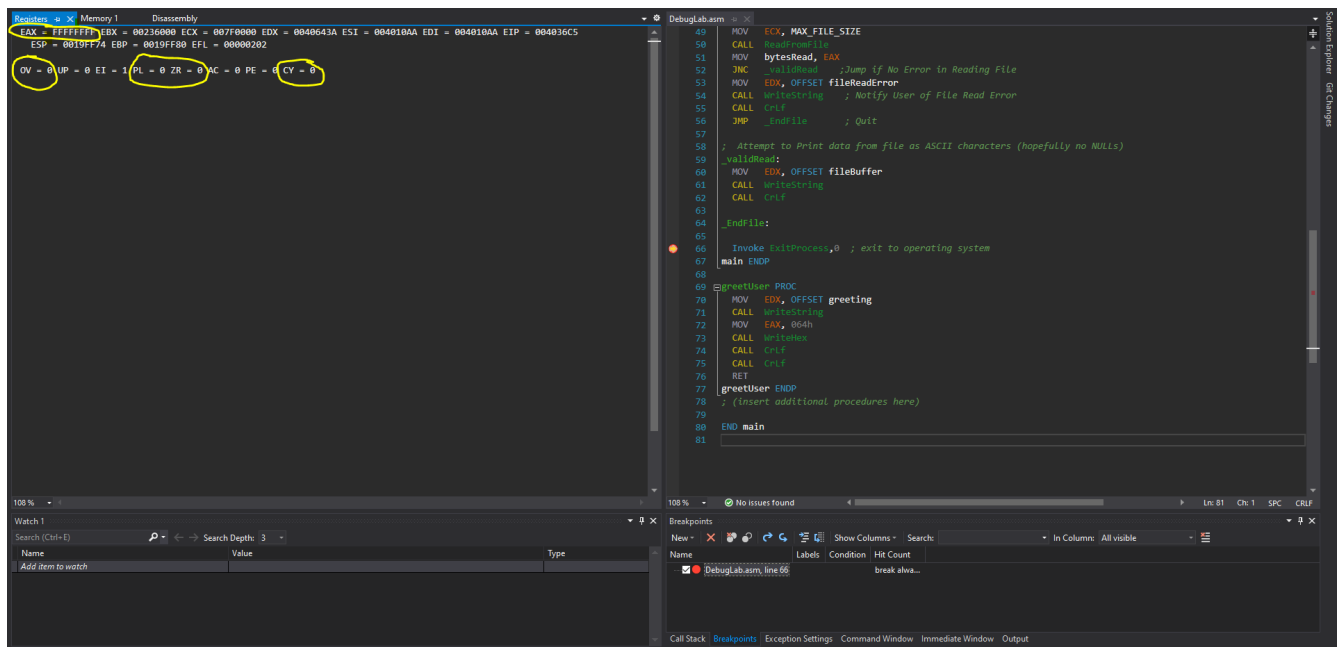


Figure 1: Part 1: Value of EAX (hex) and state of Carry, Overflow, Zero, and Sign

Part 2 – Navigating Code and Procedures

1. User number in terminal window, along with registers and editor windows.

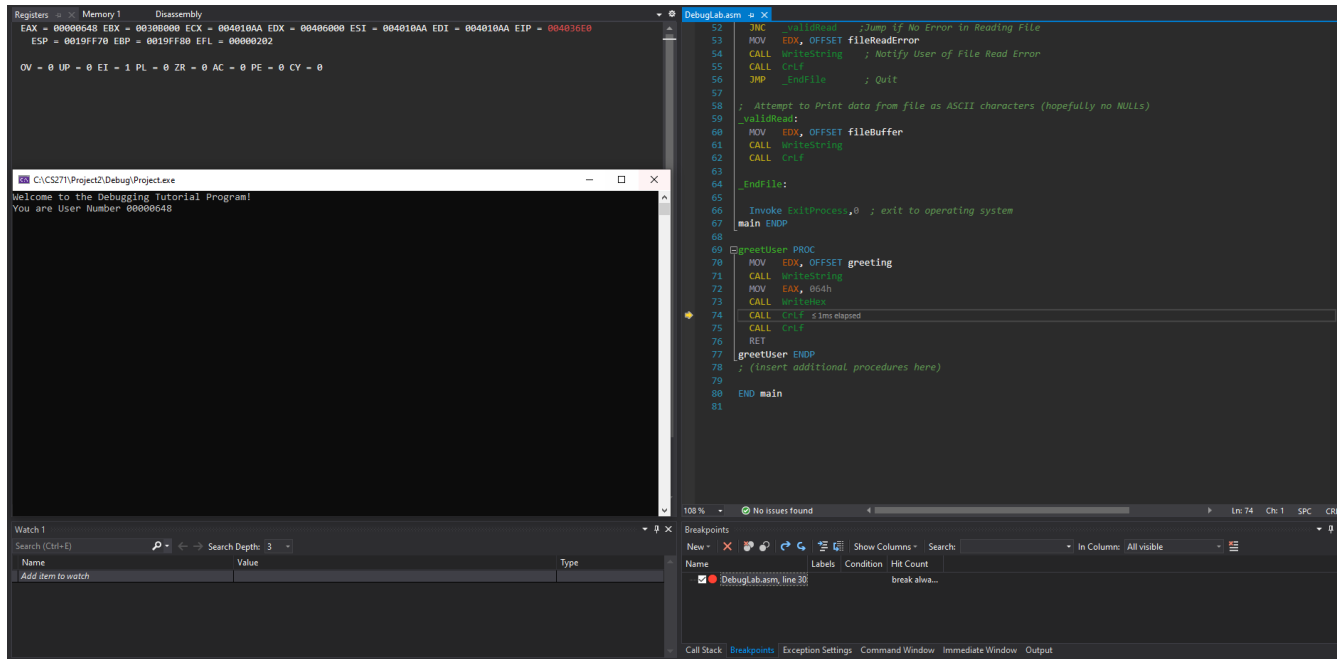
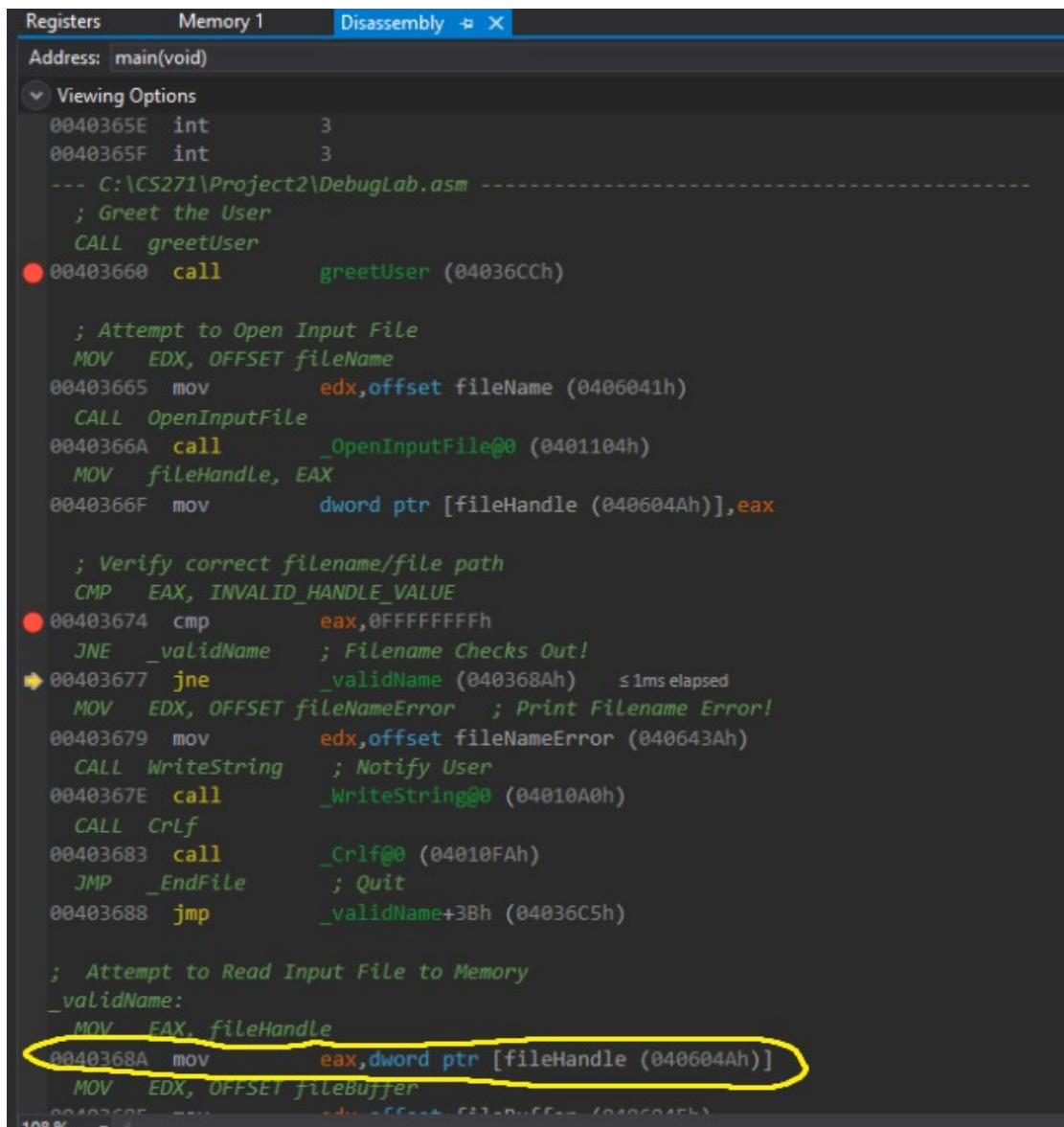


Figure 2: User Number in EAX along with Registers/Editor on screen

Part 3 – Disassembly View

1. Memory address of the instruction located at `_validName` label = 0040368Ah
2. The instruction `mov` is stored along with its operands – `eax` and `fileHandle`.



```
Registers    Memory 1    Disassembly
Address: main(void)
Viewing Options
0040365E int 3
0040365F int 3
--- C:\CS271\Project2\DebugLab.asm ---
; Greet the User
CALL greetUser
00403660 call greetUser (004036CCh)
; Attempt to Open Input File
MOV EDX, OFFSET fileName
00403665 mov edx,offset fileName (00406041h)
CALL OpenInputFile
0040366A call _OpenInputFile@0 (00401104h)
MOV fileHandle, EAX
0040366F mov dword ptr [fileHandle (0040604Ah)],eax
; Verify correct filename/file path
CMP EAX, INVALID_HANDLE_VALUE
00403674 cmp eax,0FFFFFFFFh
JNE _validName ; Filename Checks Out!
00403677 jne _validName (0040368Ah) ≤ 1ms elapsed
MOV EDX, OFFSET fileNameError ; Print Filename Error!
00403679 mov edx,offset fileNameError (0040643Ah)
CALL WriteString ; Notify User
0040367E call _WriteString@0 (004010A0h)
CALL CrLf
00403683 call _CrLf@0 (004010FAh)
JMP _EndFile ; Quit
00403688 jmp _validName+3Bh (004036C5h)
; Attempt to Read Input File to Memory
_validName:
MOV EAX, fileHandle
0040368A mov eax,dword ptr [fileHandle (0040604Ah)]
MOV EDX, OFFSET fileBuffer
0040368F mov edx,offset fileBuffer (00406045h)
```

Figure 3: Instructions and operands circled at the code segment address from question 1.

3. The EIP is the 32bit instruction pointer register, and holds the address of the next instruction to be executed. Thus, the leftmost value on any given line is the memory address of the instruction to be executed, as such, it will be stored in the EIP in the Register window.

Part 4 – Spelunking through Memory

1.

Last 3 digits of ONID id = 648

$n = 648$

$(n + 1)$ st byte (index n) = 649th byte of TestText.txt = s

2.

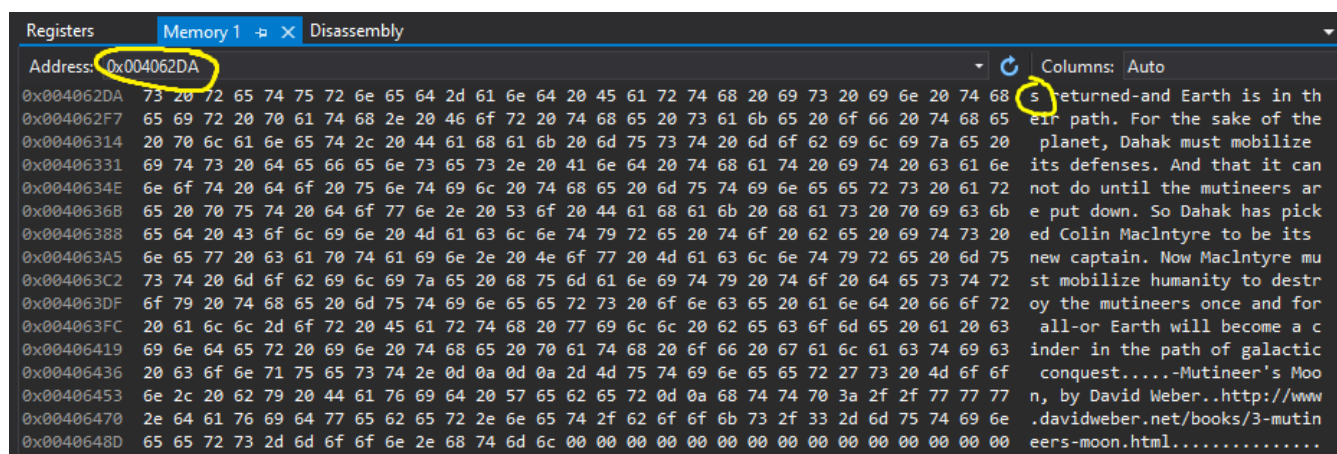


Figure 4: Memory window with value in address bar circled, as well as its ASCII representation

Part 5 – Keeping Careful Watch

1. Unsigned value of fileHandle = 240

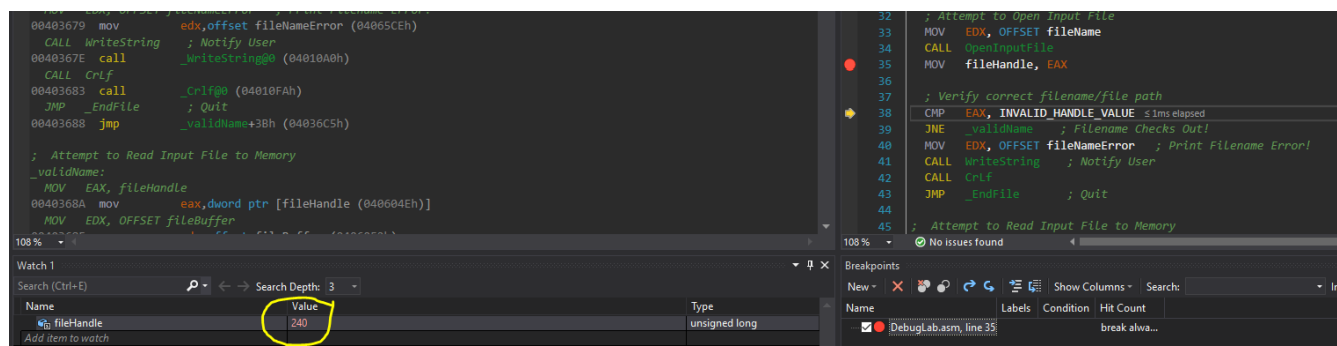


Figure 5: Unsigned value of fileHandle after opening TestText.txt