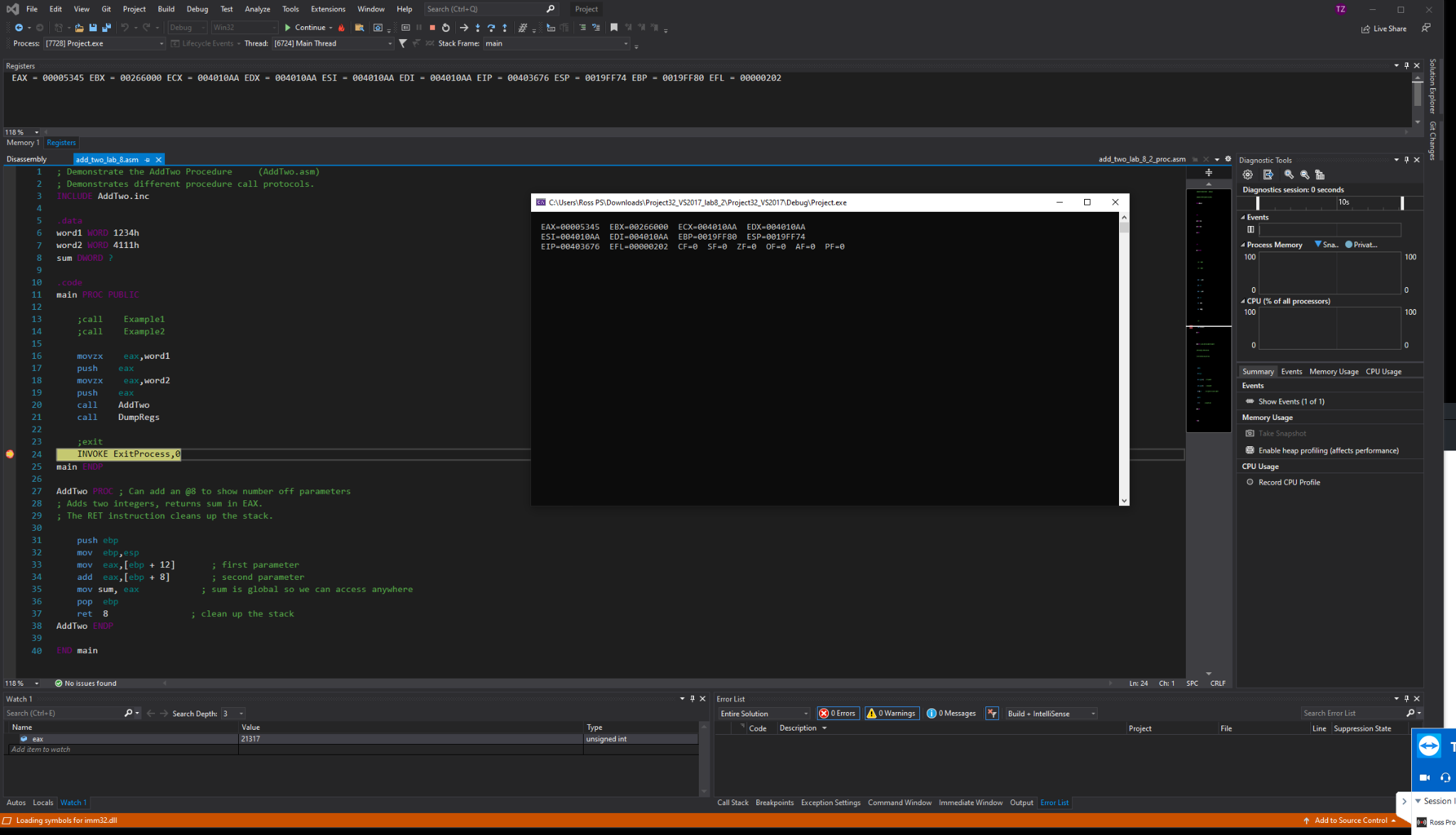
I.

lab-8.2I.asm

| ; Demonstrate the AddTwo Procedure (AddTwo.asm) ; Demonstrates different procedure call protocols. INCLUDE AddTwo.inc  .data word1 WORD 1234h word2 WORD 4111h sum DWORD ?  .code main PROC PUBLIC   movzx eax,word1  push eax  movzx eax,word2  push eax  call AddTwo  call DumpRegs   ;exit  INVOKE ExitProcess,0 main ENDP  AddTwo PROC ; Can add an @8 to show number off parameters ; Adds two integers, returns sum in EAX. ; The RET instruction cleans up the stack.   push ebp  mov ebp,esp  mov eax,[ebp + 12] ; first parameter  add eax,[ebp + 8] ; second parameter  mov sum, eax  pop ebp  ret 8 AddTwo ENDP  END main |
| --- |

AddTwo.inc

| ; header INCLUDE Irvine32.inc .386 .model flat,stdcall .stack 4096 ExitProcess PROTO, dwExitCode:DWORD ; exit PROTO  ; use private proc option OPTION PROC:PRIVATE  ; make AddTwo a proto AddTwo PROTO  DumpRegs PROTO  EXTERNDEF sum:DWORD |
| --- |



II.

AddTwoProc.asm

| INCLUDE AddTwo.inc  .code AddTwo PROC ; Adds two integers, returns sum in EAX. ; The RET instruction cleans up the stack.   push ebp  mov ebp,esp  mov eax,[ebp + 12] ; first parameter  add eax,[ebp + 8] ; second parameter  pop ebp  ret 8 ; clean up the stack AddTwo ENDP  END |
| --- |

Main.asm

| ; Demonstrate the AddTwo Procedure (AddTwo.asm) ; Demonstrates different procedure call protocols. INCLUDE AddTwo.inc  .data  word1 WORD 1234h  word2 WORD 4111h  sum DWORD ?  .code main PROC PUBLIC   ;call Example1  ;call Example2   movzx eax,word1  push eax  movzx eax,word2  push eax  call AddTwo  call DumpRegs   ; exit ; defined in irvine, but not in AddTwo.inc  INVOKE ExitProcess, 0  main ENDP  END main |
| --- |

AddTwo.inc

| ; header INCLUDE Irvine32.inc .386 .model flat,stdcall .stack 4096 ExitProcess PROTO, dwExitCode:DWORD ; exit PROTO  ; use private proc option OPTION PROC:PRIVATE  ; make AddTwo a proto AddTwo PROTO  DumpRegs PROTO  EXTERNDEF sum:DWORD |
| --- |

