Understanding C-to-Assembly and Compiled Structures

Dr. Jared DeMott
CTO AND FOUNDER

@jareddemott www.vdalabs.com

Overview

C to Assembly

Lab 2

Structures

C when compiled to x86 .asm

Reversal for standard *if*

```
if( a > 10)
  Do something
```

```
mov eax, ebp+arg_0
mov ebx, 10
cmp eax, ebx
jle loc_around

Do something code…

loc_around:
```

Multi or Compound Statement

```
if( (a > b) || (a == c)
|| (b != c) )

{

    body();

}

…
```

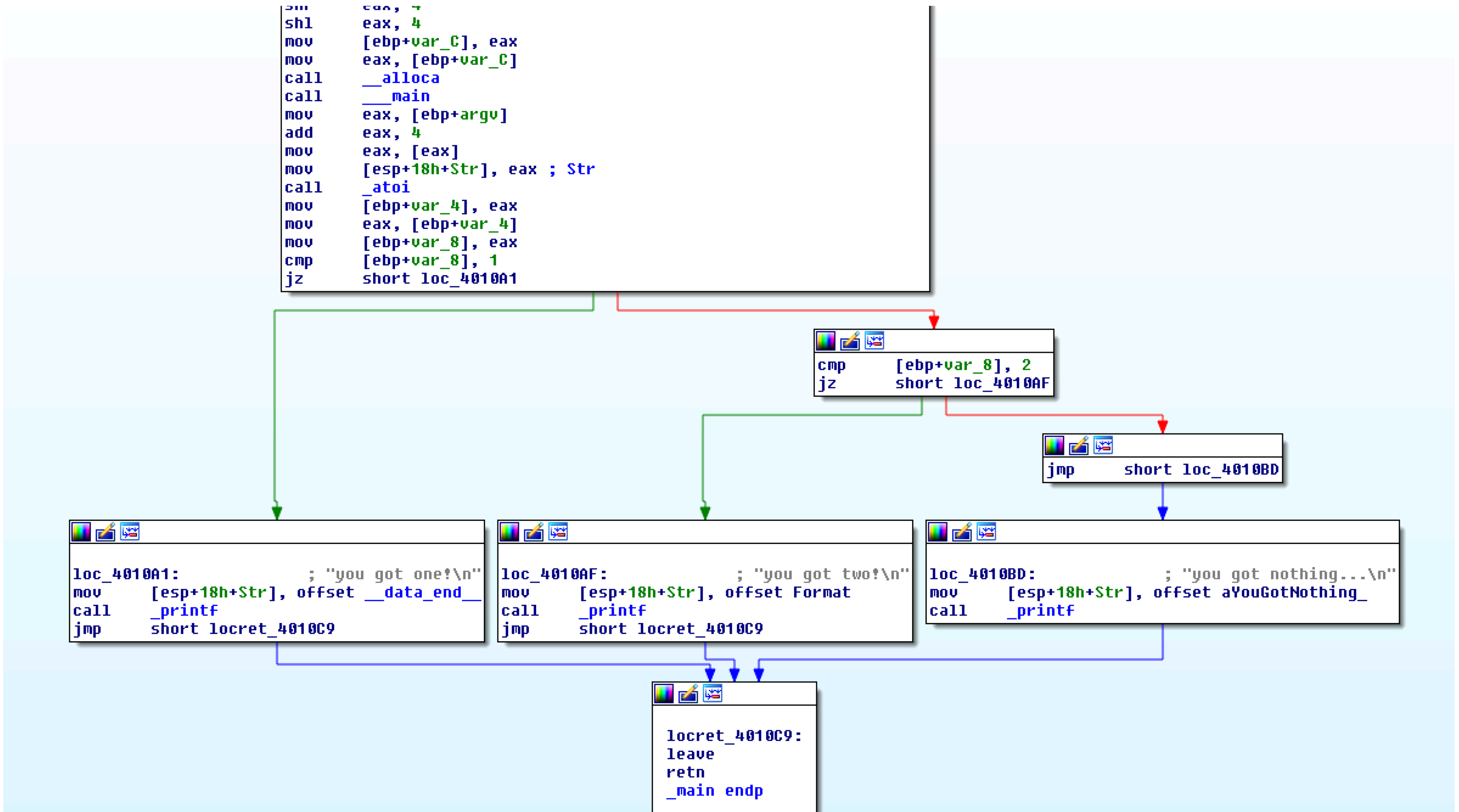
```
cmp eax, ebx

jg body

cmp eax, ecx

je body

cmp ebx, ecx

je endif

body:

    …

endif:
```

Loops

```
for(i=0; i++; i<20)
{
   body();
}
…
```

```
xor ebx, ebx
top:
cmp ebx, 20
jge out
 …     ;loop body
inc ebx
jmp top
out:
```

Small Switch Statement



```
shl     eax, 4
mov     [ebp+var_C], eax
mov     eax, [ebp+var_C]
call    __alloca
call    ___main
mov     eax, [ebp+argv]
add     eax, 4
mov     eax, [eax]
mov     [esp+18h+Str], eax ; Str
call    _atoi
mov     [ebp+var_4], eax
mov     eax, [ebp+var_4]
mov     [ebp+var_8], eax
cmp     [ebp+var_8], 1
jz      short loc_4010A1
```

```
cmp     [ebp+var_8], 2
jz      short loc_4010AF
```

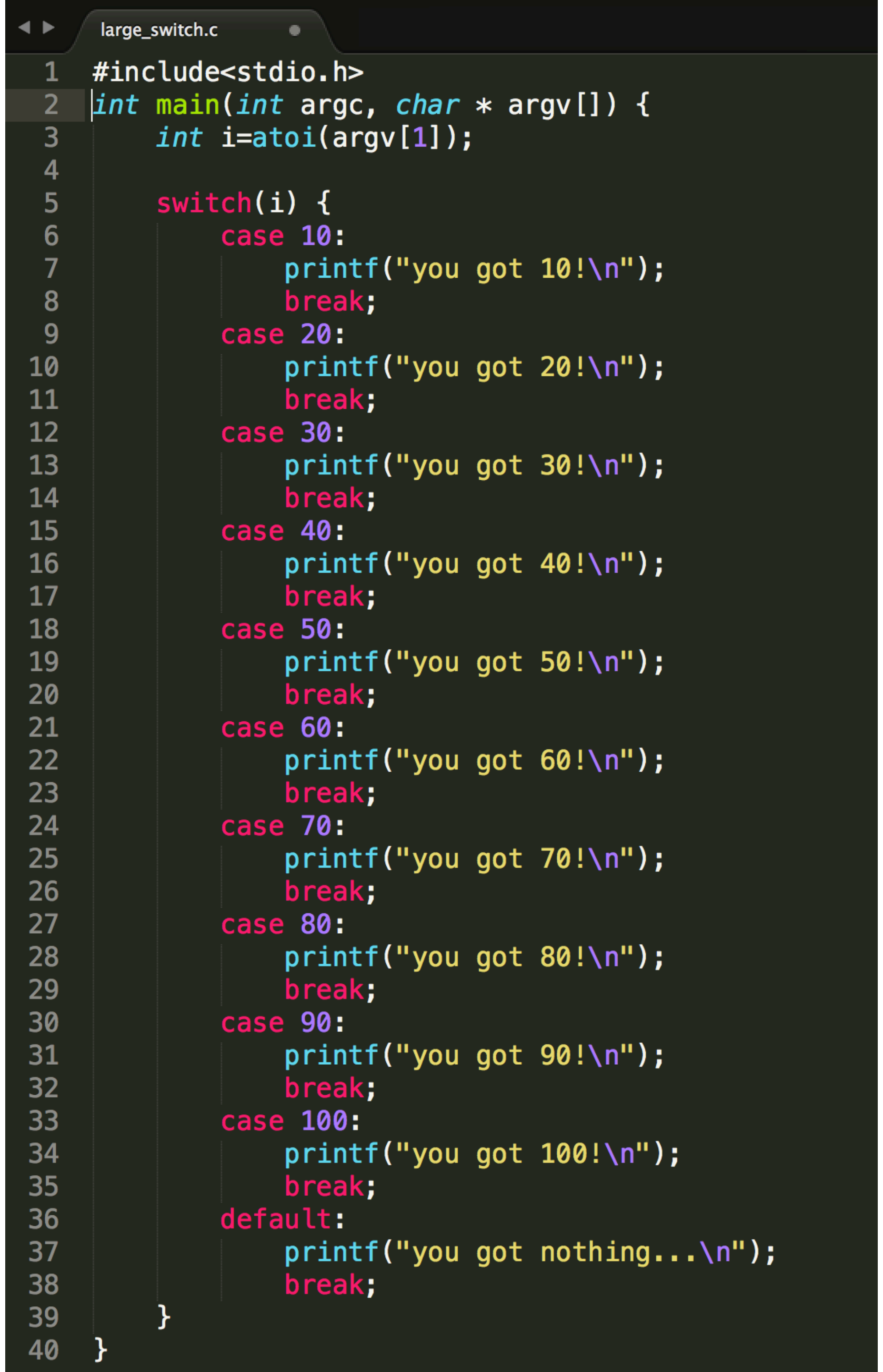
```
jmp     short loc_4010BD
```

```
loc_4010A1:                 ; "you got one!\n"
mov     [esp+18h+Str], offset __data_end__
call    _printf
jmp     short locret_4010C9
```

```
loc_4010AF:                 ; "you got two!\n"
mov     [esp+18h+Str], offset Format
call    _printf
jmp     short locret_4010C9
```

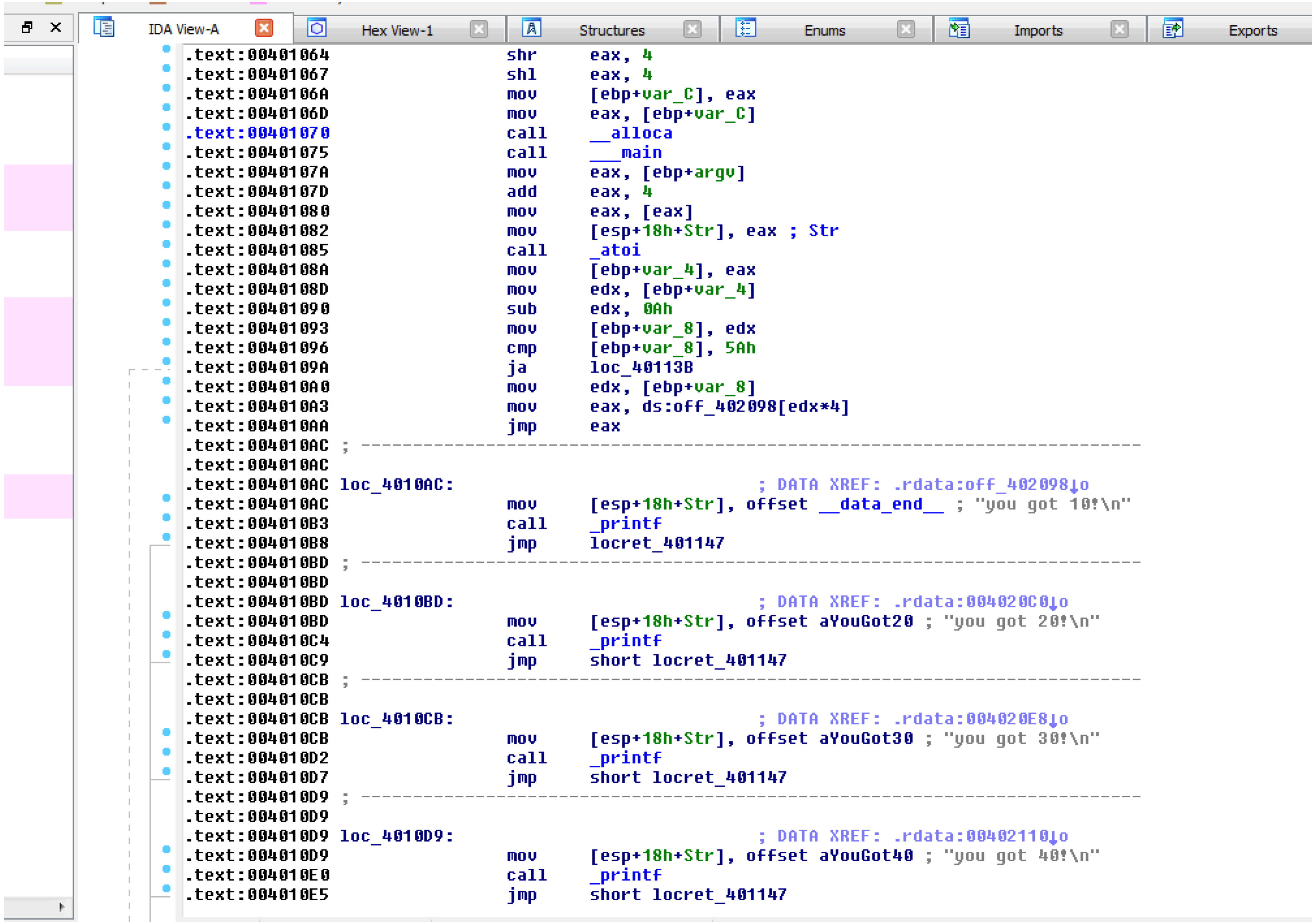
```
loc_4010BD:                 ; "you got nothing...\n"
mov     [esp+18h+Str], offset aYouGotNothing_
call    _printf
```

```
locret_4010C9:
leave
retn
_main endp
```



large_switch.c

```c
#include<stdio.h>
int main(int argc, char * argv[]) {
    int i=atoi(argv[1]);

    switch(i) {
        case 10:
            printf("you got 10!\n");
            break;
        case 20:
            printf("you got 20!\n");
            break;
        case 30:
            printf("you got 30!\n");
            break;
        case 40:
            printf("you got 40!\n");
            break;
        case 50:
            printf("you got 50!\n");
            break;
        case 60:
            printf("you got 60!\n");
            break;
        case 70:
            printf("you got 70!\n");
            break;
        case 80:
            printf("you got 80!\n");
            break;
        case 90:
            printf("you got 90!\n");
            break;
        case 100:
            printf("you got 100!\n");
            break;
        default:
            printf("you got nothing...\n");
            break;
    }
}
```



```
.text:00401064                 shr     eax, 4
.text:00401067                 shl     eax, 4
.text:0040106A                 mov     [ebp+var_C], eax
.text:0040106D                 mov     eax, [ebp+var_C]
.text:00401070                 call    __alloca
.text:00401075                 call    ___main
.text:0040107A                 mov     eax, [ebp+argv]
.text:0040107D                 add     eax, 4
.text:00401080                 mov     eax, [eax]
.text:00401082                 mov     [esp+18h+Str], eax ; Str
.text:00401085                 call    _atoi
.text:0040108A                 mov     [ebp+var_4], eax
.text:0040108D                 mov     edx, [ebp+var_4]
.text:00401090                 sub     edx, 0Ah
.text:00401093                 mov     [ebp+var_8], edx
.text:00401096                 cmp     [ebp+var_8], 5Ah
.text:0040109A                 ja      loc_40113B
.text:004010A0                 mov     edx, [ebp+var_8]
.text:004010A3                 mov     eax, ds:off_402098[edx*4]
.text:004010AA                 jmp     eax
.text:004010AC ; ---------------------------------------------------------------------------
.text:004010AC
.text:004010AC loc_4010AC:                             ; DATA XREF: .rdata:off_402098↓o
.text:004010AC                 mov     [esp+18h+Str], offset __data_end__ ; "you got 10!\n"
.text:004010B3                 call    _printf
.text:004010B8                 jmp     locret_401147
.text:004010BD ; ---------------------------------------------------------------------------
.text:004010BD
.text:004010BD loc_4010BD:                             ; DATA XREF: .rdata:004020C0↓o
.text:004010BD                 mov     [esp+18h+Str], offset aYouGot20 ; "you got 20!\n"
.text:004010C4                 call    _printf
.text:004010C9                 jmp     short locret_401147
.text:004010CB ; ---------------------------------------------------------------------------
.text:004010CB
.text:004010CB loc_4010CB:                             ; DATA XREF: .rdata:004020E8↓o
.text:004010CB                 mov     [esp+18h+Str], offset aYouGot30 ; "you got 30!\n"
.text:004010D2                 call    _printf
.text:004010D7                 jmp     short locret_401147
.text:004010D9 ; ---------------------------------------------------------------------------
.text:004010D9
.text:004010D9 loc_4010D9:                             ; DATA XREF: .rdata:00402110↓o
.text:004010D9                 mov     [esp+18h+Str], offset aYouGot40 ; "you got 40!\n"
.text:004010E0                 call    _printf
.text:004010E5                 jmp     short locret_401147
```

Demo

Solve the password

- Start by running the program and finding main()
- Either trace through start (usually main is called towards the bottom)
- Or use the strings and jump to one
- Notice how compiled ‘math’ looks
- In this example, you can ignore funny compiler things when dividing and such

Identifying Structures

Fixing up the IDA with data definitions

C Structures

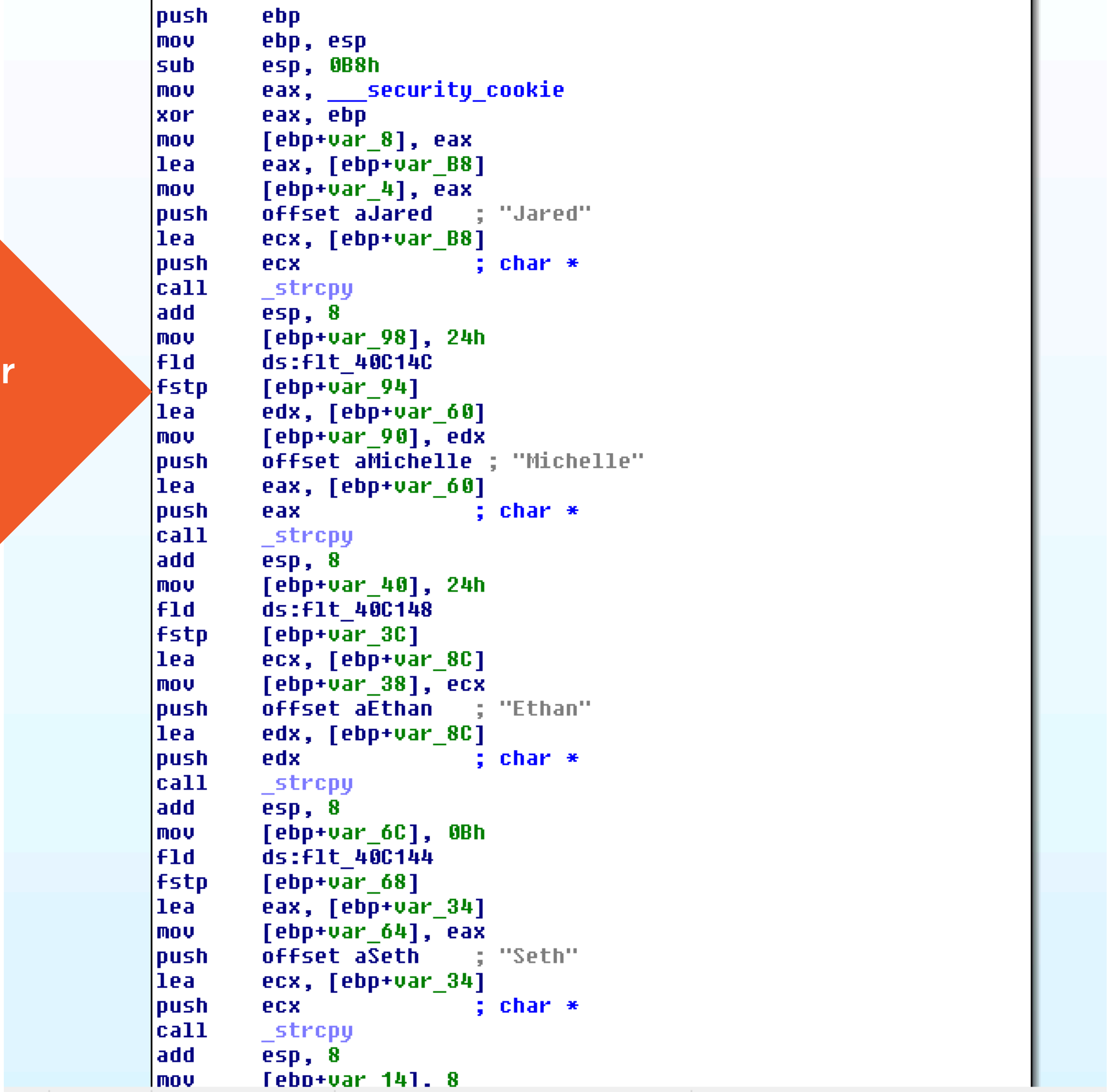
Can be tough to reverse

- Accesses to structures may resemble separate stack variables
- For common structures, IDA may auto recognize via types that it has built in
  - Called signatures and types by IDA (IDA/ids & IDA/til)

Once reconstructed via human analysis, a structure can be defined and overlaid on the numeric offsets

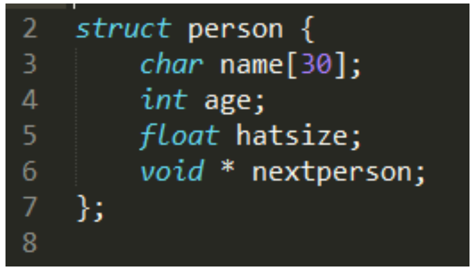
- Header files can be imported as well to teach IDA about structures we know exist in the .idb

Lot of local variables. What is var_98 or var_94?

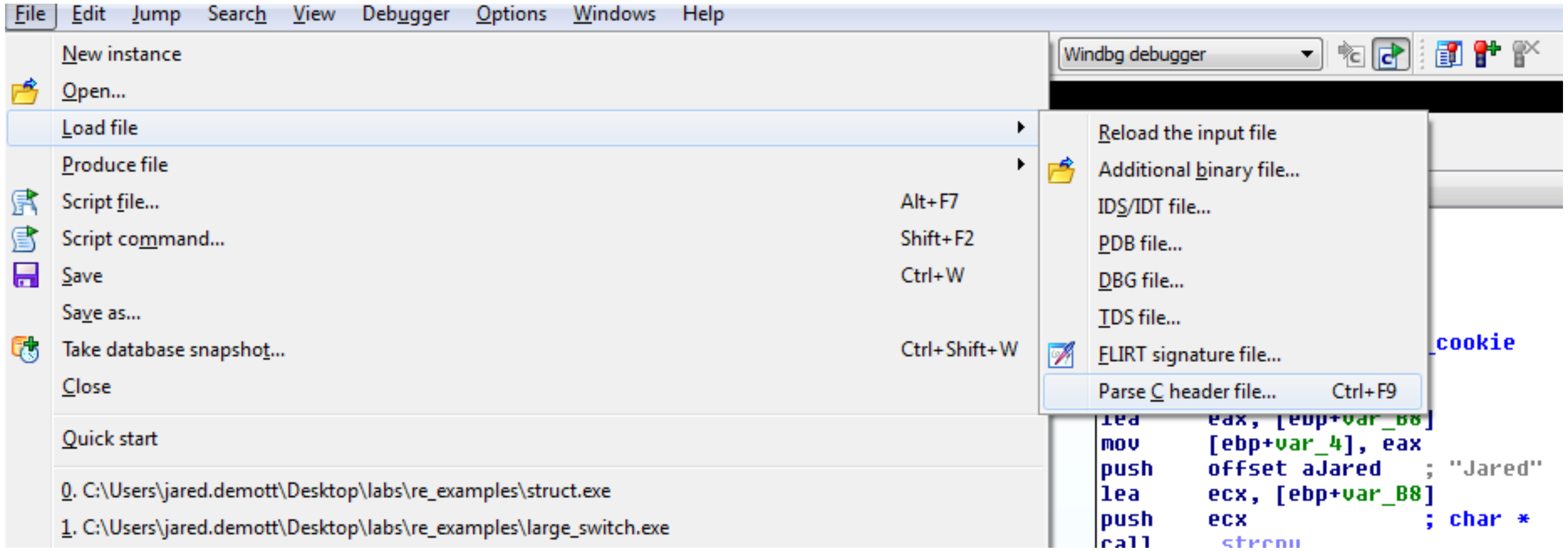


```
push    ebp
mov     ebp, esp
sub     esp, 0B8h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_8], eax
lea     eax, [ebp+var_B8]
mov     [ebp+var_4], eax
push    offset aJared   ; "Jared"
lea     ecx, [ebp+var_B8]
push    ecx             ; char *
call    _strcpy
add     esp, 8
mov     [ebp+var_98], 24h
fld     ds:flt_40C14C
fstp    [ebp+var_94]
lea     edx, [ebp+var_60]
mov     [ebp+var_90], edx
push    offset aMichelle ; "Michelle"
lea     eax, [ebp+var_60]
push    eax             ; char *
call    _strcpy
add     esp, 8
mov     [ebp+var_40], 24h
fld     ds:flt_40C148
fstp    [ebp+var_3C]
lea     ecx, [ebp+var_8C]
mov     [ebp+var_38], ecx
push    offset aEthan   ; "Ethan"
lea     edx, [ebp+var_8C]
push    edx             ; char *
call    _strcpy
add     esp, 8
mov     [ebp+var_6C], 0Bh
fld     ds:flt_40C144
fstp    [ebp+var_68]
lea     eax, [ebp+var_34]
mov     [ebp+var_64], eax
push    offset aSeth    ; "Seth"
lea     ecx, [ebp+var_34]
push    ecx             ; char *
call    _strcpy
add     esp, 8
mov     [ebp+var_14], 8
```

Create or Import a structure



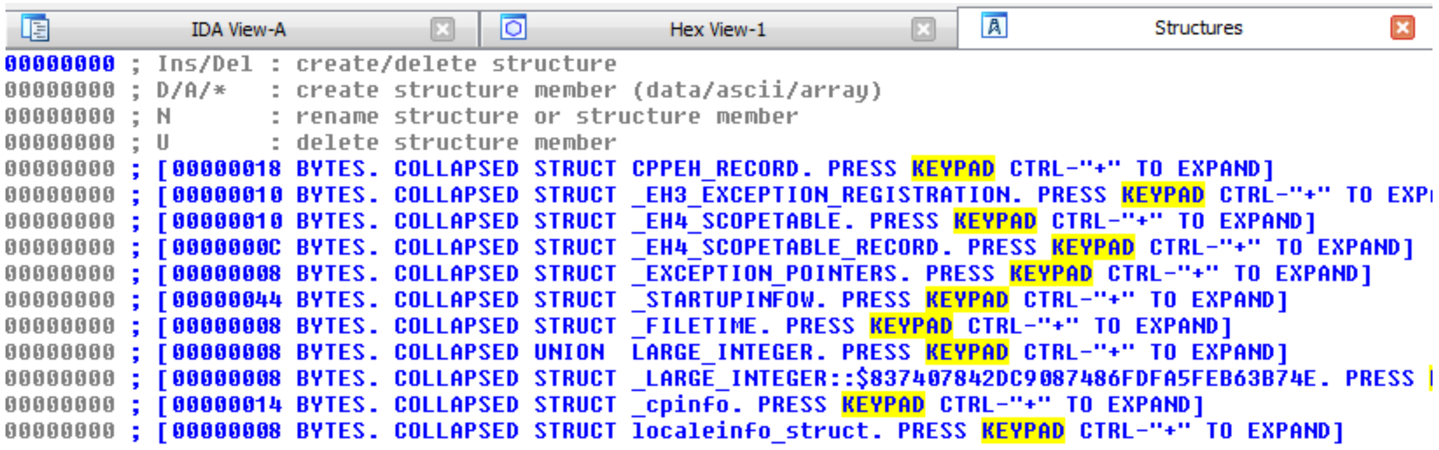
```
struct person {
    char name[30];
    int age;
    float hatsize;
    void * nextperson;
};
```



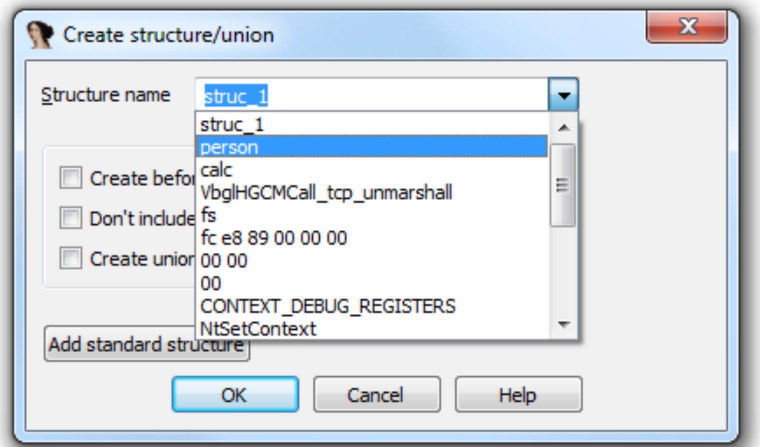
File Edit Jump Search View Debugger Options Windows Help
New instance
Open...
Load file
Produce file
Script file... Alt+F7
Script command... Shift+F2
Save Ctrl+W
Save as...
Take database snapshot... Ctrl+Shift+W
Close
Quick start
0. C:\Users\jared.demott\Desktop\labs\re_examples\struct.exe
1. C:\Users\jared.demott\Desktop\labs\re_examples\large_switch.exe
Windbg debugger
Reload the input file
Additional binary file...
IDS/IDT file...
PDB file...
DBG file...
TDS file...
FLIRT signature file...
Parse C header file... Ctrl+F9

```
                                         cookie
lea     eax, [ebp+var_B8]
mov     [ebp+var_4], eax
push    offset aJared      ; "Jared"
lea     ecx, [ebp+var_B8]
push    ecx                ; char *
call    strcpy
```

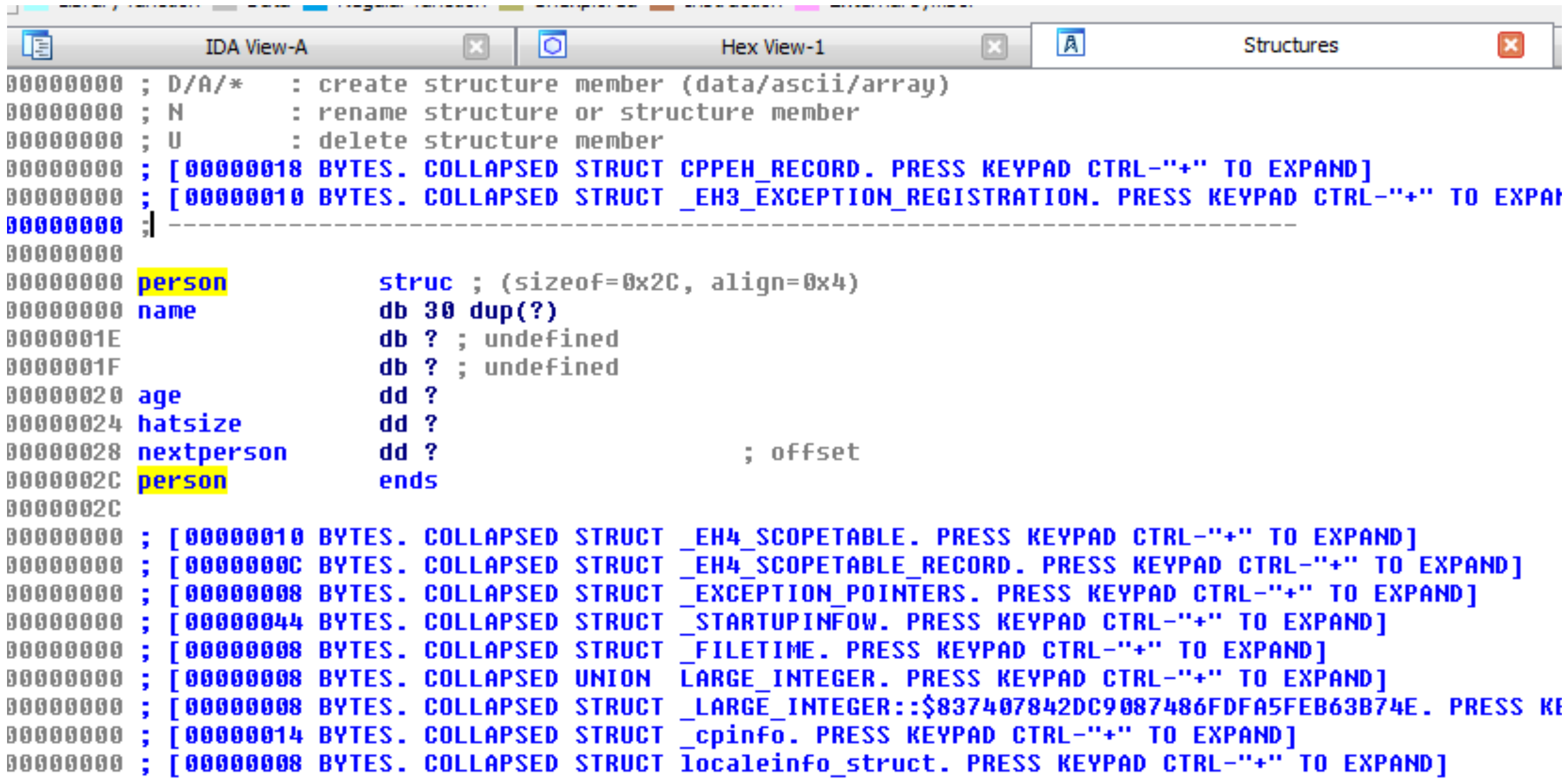
Add Struct



```
00000000 ; Ins/Del : create/delete structure
00000000 ; D/A/*   : create structure member (data/ascii/array)
00000000 ; N       : rename structure or structure member
00000000 ; U       : delete structure member
00000000 ; [00000018 BYTES. COLLAPSED STRUCT CPPEH_RECORD. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000010 BYTES. COLLAPSED STRUCT _EH3_EXCEPTION_REGISTRATION. PRESS KEYPAD CTRL-"+" TO EXP
00000000 ; [00000010 BYTES. COLLAPSED STRUCT _EH4_SCOPETABLE. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [0000000C BYTES. COLLAPSED STRUCT _EH4_SCOPETABLE_RECORD. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT _EXCEPTION_POINTERS. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000044 BYTES. COLLAPSED STRUCT _STARTUPINFOW. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT _FILETIME. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED UNION  LARGE_INTEGER. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT _LARGE_INTEGER::$837407842DC9087486FDFA5FEB63B74E. PRESS
00000000 ; [00000014 BYTES. COLLAPSED STRUCT _cpinfo. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT localeinfo_struct. PRESS KEYPAD CTRL-"+" TO EXPAND]
```



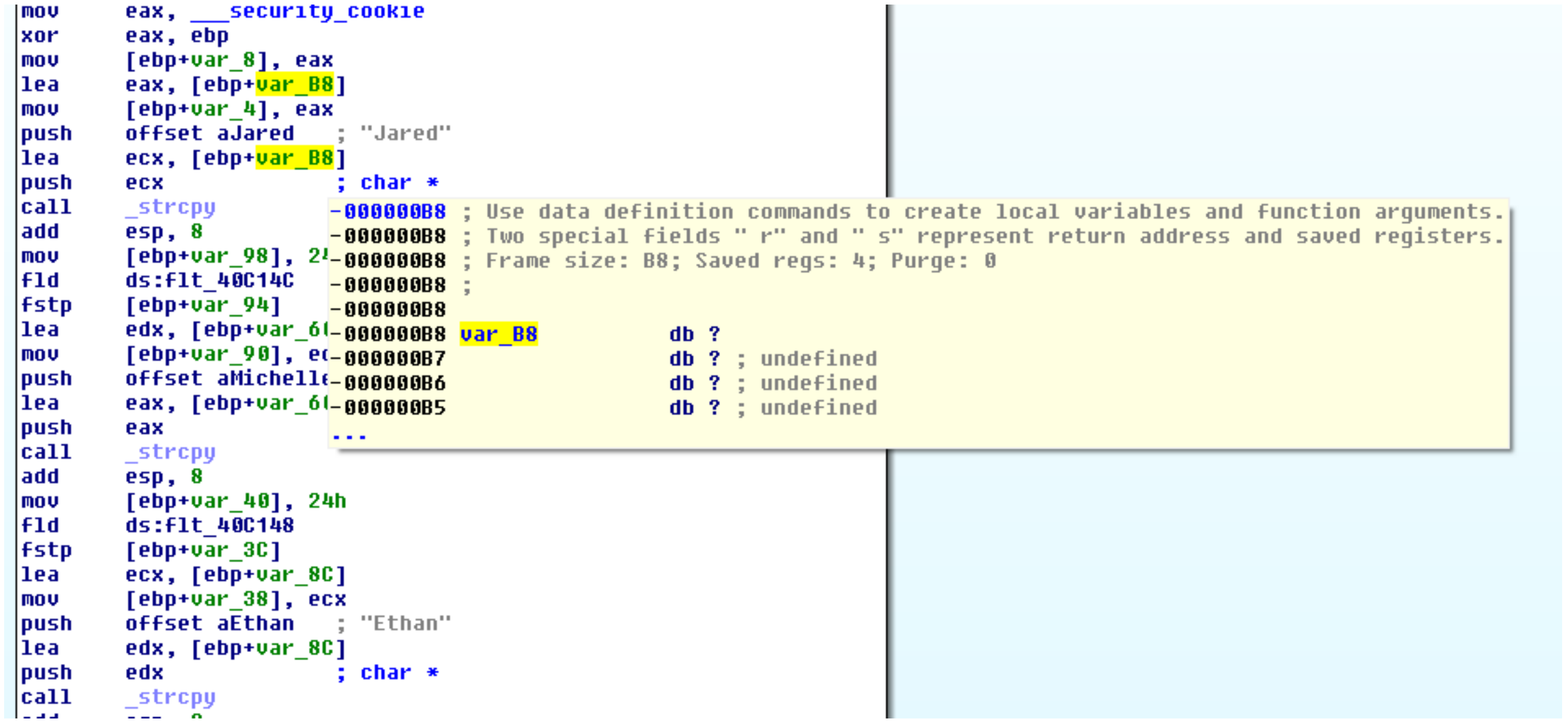
Added Struct



IDA View-A
Hex View-1
Structures

```
00000000 ; D/A/*    : create structure member (data/ascii/array)
00000000 ; N        : rename structure or structure member
00000000 ; U        : delete structure member
00000000 ; [00000018 BYTES. COLLAPSED STRUCT CPPEH_RECORD. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000010 BYTES. COLLAPSED STRUCT _EH3_EXCEPTION_REGISTRATION. PRESS KEYPAD CTRL-"+" TO EXPAI
00000000 ; ---------------------------------------------------------------------------
00000000
00000000 person          struc ; (sizeof=0x2C, align=0x4)
00000000 name            db 30 dup(?)
0000001E                 db ? ; undefined
0000001F                 db ? ; undefined
00000020 age             dd ?
00000024 hatsize         dd ?
00000028 nextperson      dd ?                    ; offset
0000002C person          ends
0000002C
00000000 ; [00000010 BYTES. COLLAPSED STRUCT _EH4_SCOPETABLE. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [0000000C BYTES. COLLAPSED STRUCT _EH4_SCOPETABLE_RECORD. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT _EXCEPTION_POINTERS. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000044 BYTES. COLLAPSED STRUCT _STARTUPINFOW. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT _FILETIME. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED UNION  LARGE_INTEGER. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT _LARGE_INTEGER::$837407842DC9087486FDFA5FEB63B74E. PRESS KE
00000000 ; [00000014 BYTES. COLLAPSED STRUCT _cpinfo. PRESS KEYPAD CTRL-"+" TO EXPAND]
00000000 ; [00000008 BYTES. COLLAPSED STRUCT localeinfo_struct. PRESS KEYPAD CTRL-"+" TO EXPAND]
```

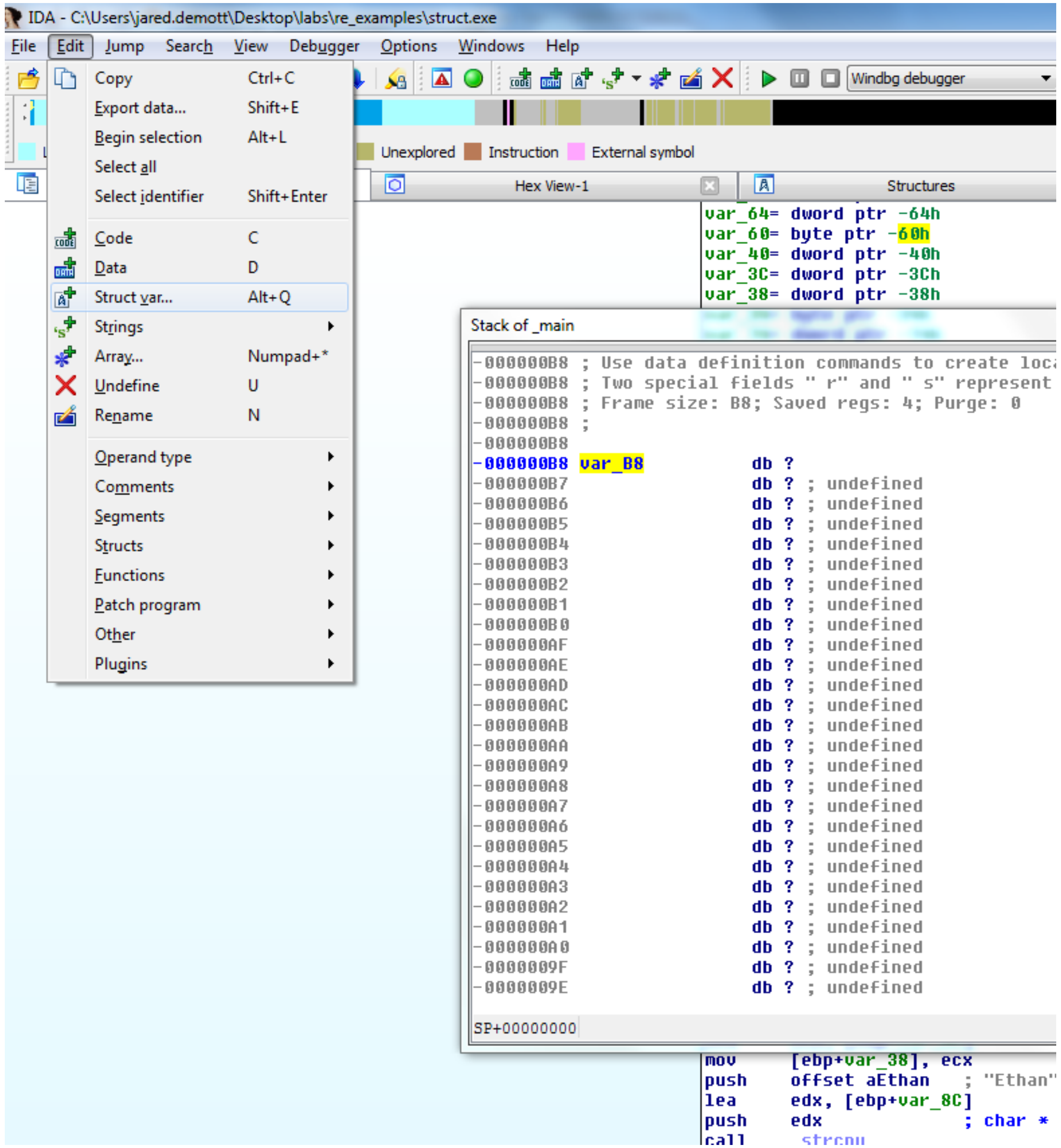
Find Stack Variable



```
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_8], eax
lea     eax, [ebp+var_B8]
mov     [ebp+var_4], eax
push    offset aJared   ; "Jared"
lea     ecx, [ebp+var_B8]
push    ecx             ; char *
call    _strcpy
add     esp, 8
mov     [ebp+var_98], 24
fld     ds:flt_40C14C
fstp    [ebp+var_94]
lea     edx, [ebp+var_60
mov     [ebp+var_90], ec
push    offset aMichelle
lea     eax, [ebp+var_60
push    eax
call    _strcpy
add     esp, 8
mov     [ebp+var_40], 24h
fld     ds:flt_40C148
fstp    [ebp+var_3C]
lea     ecx, [ebp+var_8C]
mov     [ebp+var_38], ecx
push    offset aEthan   ; "Ethan"
lea     edx, [ebp+var_8C]
push    edx             ; char *
call    _strcpy
```

```
-000000B8 ; Use data definition commands to create local variables and function arguments.
-000000B8 ; Two special fields " r" and " s" represent return address and saved registers.
-000000B8 ; Frame size: B8; Saved regs: 4; Purge: 0
-000000B8 ;
-000000B8
-000000B8 var_B8          db ?
-000000B7                 db ? ; undefined
-000000B6                 db ? ; undefined
-000000B5                 db ? ; undefined
...
```

Apply Struct Var

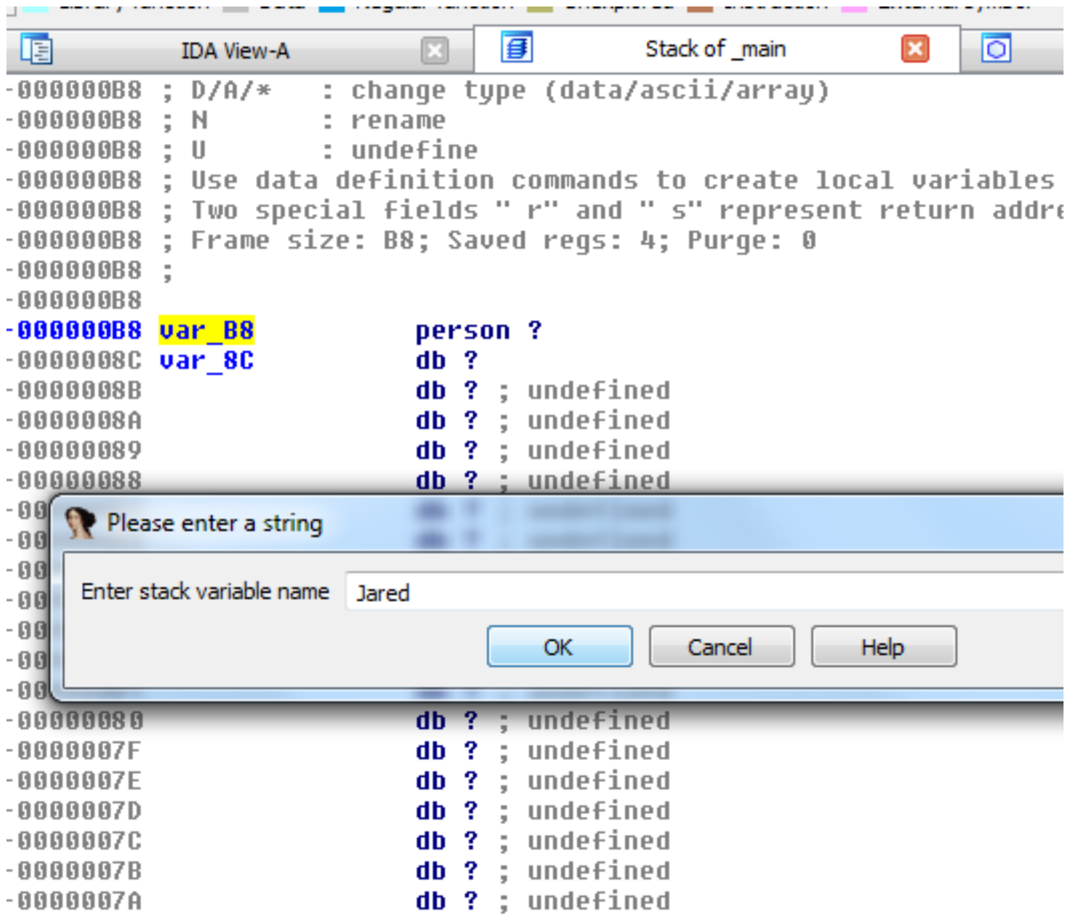


```
var_64= dword ptr -64h
var_60= byte ptr -60h
var_40= dword ptr -40h
var_3C= dword ptr -3Ch
var_38= dword ptr -38h
```

```
-000000B8 ; Use data definition commands to create loca
-000000B8 ; Two special fields " r" and " s" represent
-000000B8 ; Frame size: B8; Saved regs: 4; Purge: 0
-000000B8 ;
-000000B8
-000000B8 var_B8          db ?
-000000B7                 db ? ; undefined
-000000B6                 db ? ; undefined
-000000B5                 db ? ; undefined
-000000B4                 db ? ; undefined
-000000B3                 db ? ; undefined
-000000B2                 db ? ; undefined
-000000B1                 db ? ; undefined
-000000B0                 db ? ; undefined
-000000AF                 db ? ; undefined
-000000AE                 db ? ; undefined
-000000AD                 db ? ; undefined
-000000AC                 db ? ; undefined
-000000AB                 db ? ; undefined
-000000AA                 db ? ; undefined
-000000A9                 db ? ; undefined
-000000A8                 db ? ; undefined
-000000A7                 db ? ; undefined
-000000A6                 db ? ; undefined
-000000A5                 db ? ; undefined
-000000A4                 db ? ; undefined
-000000A3                 db ? ; undefined
-000000A2                 db ? ; undefined
-000000A1                 db ? ; undefined
-000000A0                 db ? ; undefined
-0000009F                 db ? ; undefined
-0000009E                 db ? ; undefined
SP+00000000
```

```
mov     [ebp+var_38], ecx
push    offset aEthan   ; "Ethan"
lea     edx, [ebp+var_8C]
push    edx             ; char *
call
```

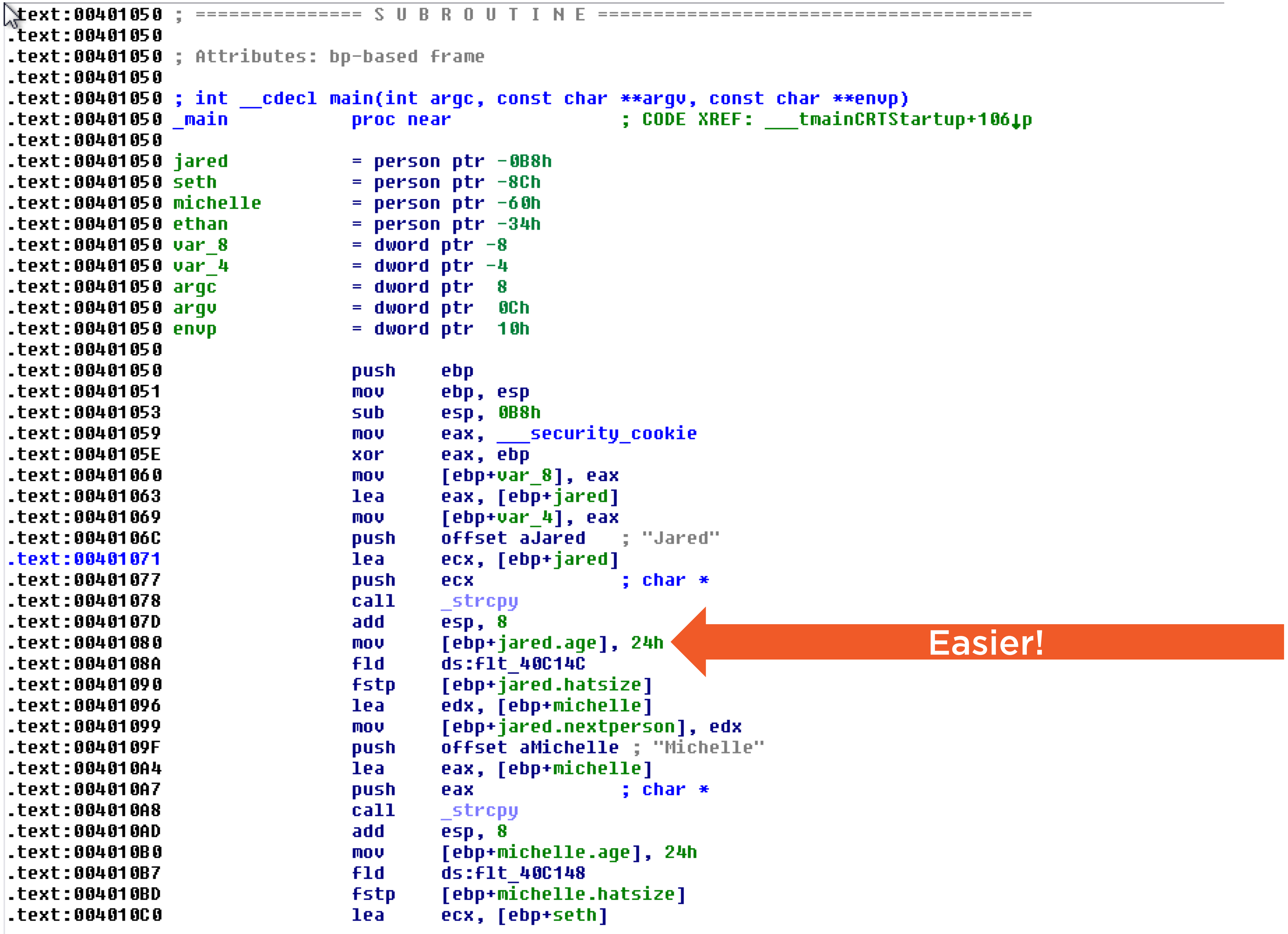
Rename Stack Var



```
-000000B8 ; D/A/*    : change type (data/ascii/array)
-000000B8 ; N        : rename
-000000B8 ; U        : undefine
-000000B8 ; Use data definition commands to create local variables
-000000B8 ; Two special fields " r" and " s" represent return addre
-000000B8 ; Frame size: B8; Saved regs: 4; Purge: 0
-000000B8 ;
-000000B8
-000000B8 var_B8          person ?
-0000008C var_8C          db ?
-0000008B                 db ? ; undefined
-0000008A                 db ? ; undefined
-00000089                 db ? ; undefined
-00000088                 db ? ; undefined
-00000080                 db ? ; undefined
-0000007F                 db ? ; undefined
-0000007E                 db ? ; undefined
-0000007D                 db ? ; undefined
-0000007C                 db ? ; undefined
-0000007B                 db ? ; undefined
-0000007A                 db ? ; undefined
```

Please enter a string

Enter stack variable name: Jared



```
.text:00401050 ; =============== S U B R O U T I N E =======================================
.text:00401050
.text:00401050 ; Attributes: bp-based frame
.text:00401050
.text:00401050 ; int __cdecl main(int argc, const char **argv, const char **envp)
.text:00401050 _main           proc near               ; CODE XREF: ___tmainCRTStartup+106↓p
.text:00401050
.text:00401050 jared           = person ptr -0B8h
.text:00401050 seth            = person ptr -8Ch
.text:00401050 michelle        = person ptr -60h
.text:00401050 ethan           = person ptr -34h
.text:00401050 var_8           = dword ptr -8
.text:00401050 var_4           = dword ptr -4
.text:00401050 argc            = dword ptr  8
.text:00401050 argv            = dword ptr  0Ch
.text:00401050 envp            = dword ptr  10h
.text:00401050
.text:00401050                 push    ebp
.text:00401051                 mov     ebp, esp
.text:00401053                 sub     esp, 0B8h
.text:00401059                 mov     eax, ___security_cookie
.text:0040105E                 xor     eax, ebp
.text:00401060                 mov     [ebp+var_8], eax
.text:00401063                 lea     eax, [ebp+jared]
.text:00401069                 mov     [ebp+var_4], eax
.text:0040106C                 push    offset aJared   ; "Jared"
.text:00401071                 lea     ecx, [ebp+jared]
.text:00401077                 push    ecx             ; char *
.text:00401078                 call    _strcpy
.text:0040107D                 add     esp, 8
.text:00401080                 mov     [ebp+jared.age], 24h
.text:0040108A                 fld     ds:flt_40C14C
.text:00401090                 fstp    [ebp+jared.hatsize]
.text:00401096                 lea     edx, [ebp+michelle]
.text:00401099                 mov     [ebp+jared.nextperson], edx
.text:0040109F                 push    offset aMichelle ; "Michelle"
.text:004010A4                 lea     eax, [ebp+michelle]
.text:004010A7                 push    eax             ; char *
.text:004010A8                 call    _strcpy
.text:004010AD                 add     esp, 8
.text:004010B0                 mov     [ebp+michelle.age], 24h
.text:004010B7                 fld     ds:flt_40C148
.text:004010BD                 fstp    [ebp+michelle.hatsize]
.text:004010C0                 lea     ecx, [ebp+seth]
```

Easier!

Summary

x86 < -- > C

Homework

Structures

Next:

- Binary patching