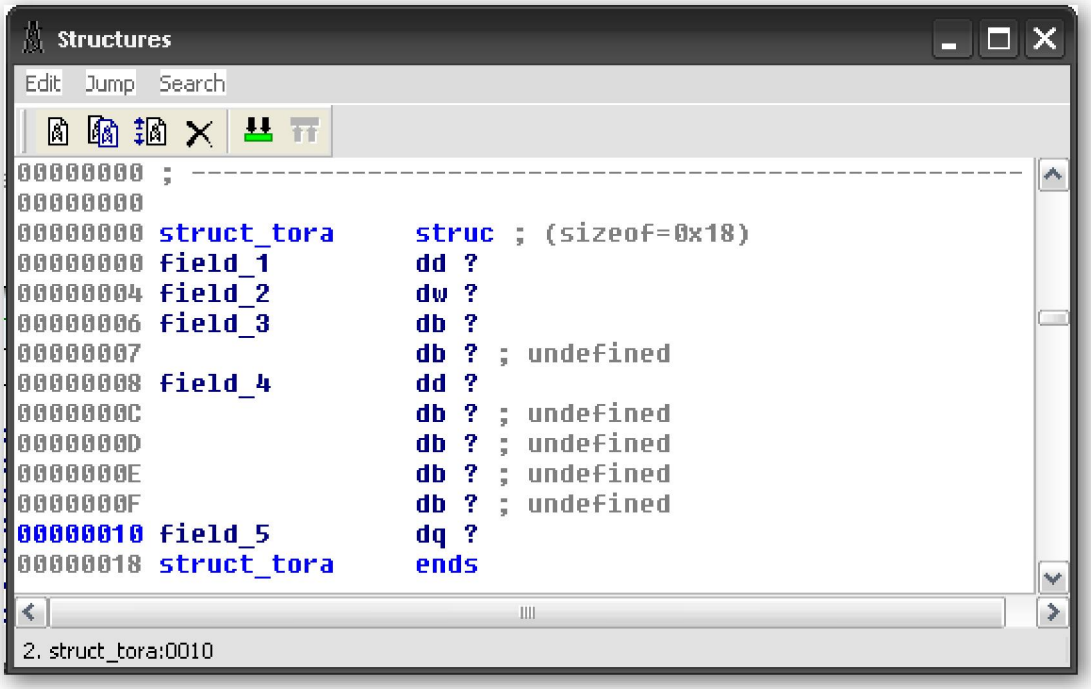
7.3.—Utilizar estructuras plantilla

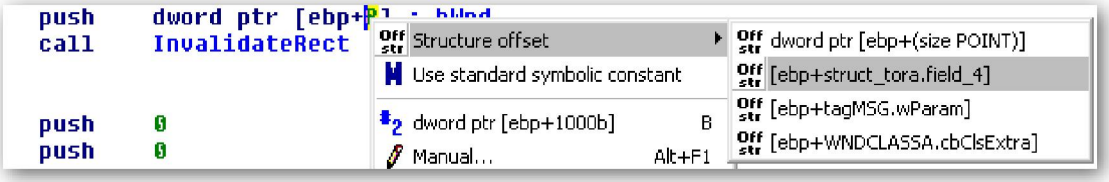
Tenemos dos formas de utilizar las definiciones de estructura en los desensamblados. Primera, podemos reformatear las referencias de memoria para hacerlas más legibles convirtiendo los offset numéricos de la estructura **[ebx + 8]** en referencias simbólicas como **[ebx + struct_tora.field4]**. Esta última forma nos proporciona más información acerca de lo que se está referenciado. Ya que IDA utiliza una notación jerárquica, se sabe con exactitud el tipo de estructura y exactamente a que campo en la estructura se está accediendo. La técnica de aplicar plantillas de estructura es utilizada frecuentemente cuando tenemos una estructura la cual es referenciada a través de un puntero. La segunda forma de utilizar plantillas de estructura es proporcionar tipos de dato adicionales que puedan ser aplicados a las variables de pila y globales.



A fin de comprender como pueden aplicarse las instrucciones de operandos en las definiciones de estructura, es útil mirar cada definición similarmente como un conjunto de constantes enumeradas. Por ejemplo la definición de struct_tora, figura arriba, podría expresarse en pseudocódigo C de la siguiente forma:

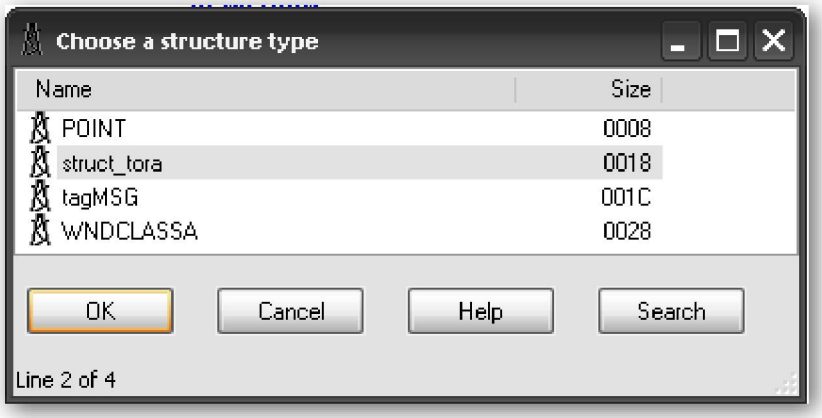
```
enum
        {
        struct_tora.field1 = 0,
        struct_tora.field2 = 4,
        struct_tora.field3 = 6,
        struct_tora.field4 = 8,
        struct_tora.field5 = 16
        };
```

Con esta definición, IDA te permite reformatear cualquier valor constante, utilizado en un operando, en una representación simbólica equivalente. En la figura abajo vemos el proceso de dicha operación. La referencia de memoria **[ebp +8]** puede representar un acceso a **field4** de **struct_tora**.



La opción **Structure offset**, la cual se mostrará haciendo click derecho en 8, en este caso, nos ofrece tres alternativas para formatear el operando de la instrucción. Las alternativas son mostradas como estructuras habilitadas las cuales contienen un campo cuyo valor es 8.

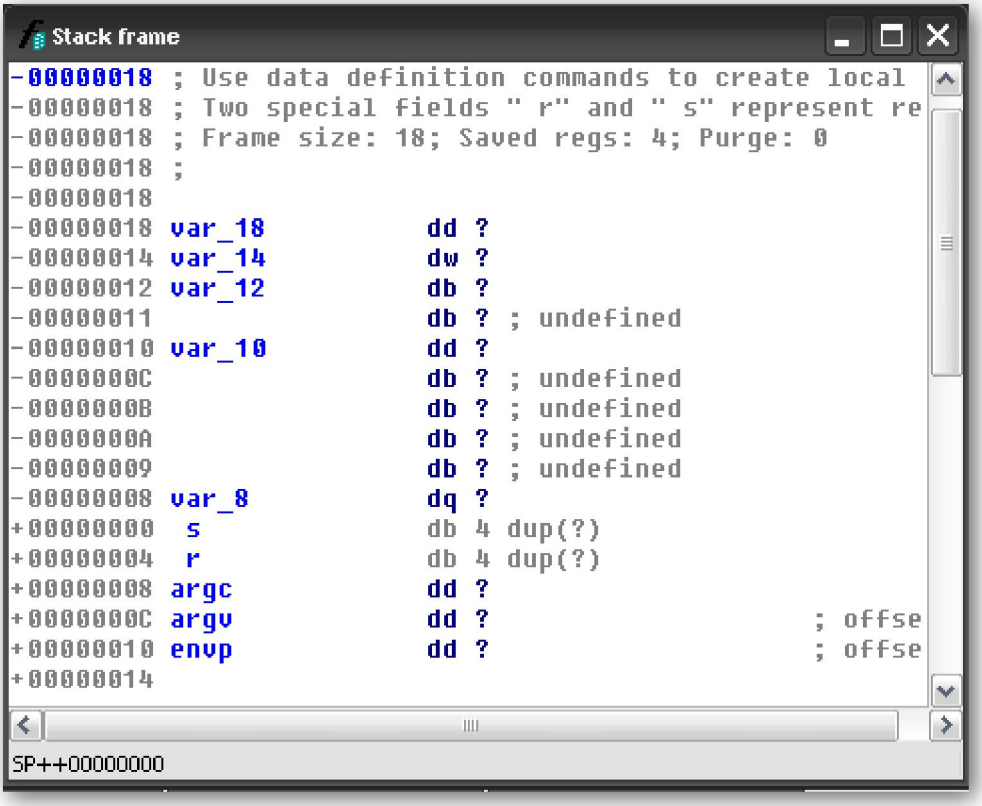
Como alternativa a formatear individualmente referencias de memoria, las variables de pila y globales pueden formatearse como estructuras enteras. Para formatear una variable de pila como una estructura, abrimos la vista detallada de stack frame y hacemos doble click en la variable que va a ser formateada como estructura, una vez hecho esto realizamos la acción **Edit > Struct Var** o el atajo **ALT-Q** con lo que nos mostrará una lista de estructuras conocidas similar a la figura abajo.



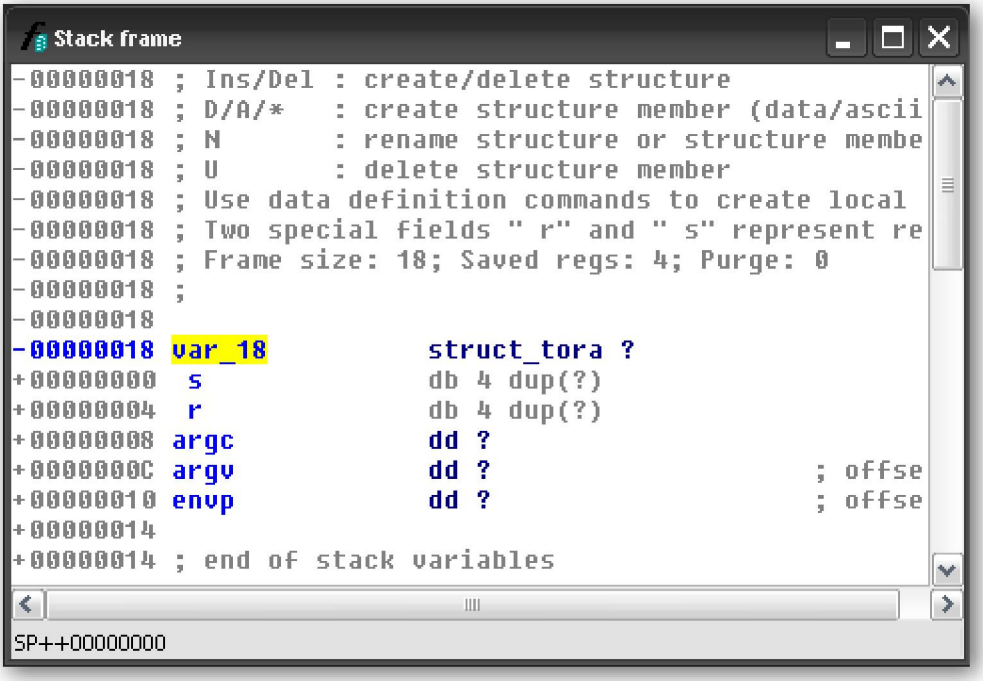
Seleccionar una de las estructuras proporcionadas, combina el número de bytes correspondientes que existen en la pila dentro de la estructura correspondiente y reforma todas las referencias de memoria como referencias a la estructura. El siguiente código es un extracto del ejemplo anterior, **stack_struct_demo**.

```
.text:00401006                mov     [ebp+var_18], 0Ah
.text:0040100D                mov     [ebp+var_14], 14h
.text:00401013                mov     [ebp+var_12], 1Eh
.text:00401017                mov     [ebp+var_10], 28h
.text:0040101E                fld     ds:dbl_40B128
.text:00401024                fstp    [ebp+var_8]
```

Recordando, llegamos a la conclusión de que **var_18** es en realidad el primer campo de una estructura de 24 byte. El **stack frame detallado** de nuestra particular interpretación se muestra en, figura abajo.



Si seleccionamos **var_18** y la formateamos como **struct_tora** (**Edit > Struct Var**) encogemos los 24 bytes, tamaño de **struct_tora**, en una sola variable **var_18** y en la cual se inicia la estructura, el resultado del formateado de la vista de pila lo vemos en, figura abajo. En este caso, si aplicamos la estructura plantilla a **var_18** nos generará un mensaje de aviso, indicándonos que algunas variables serán destruidas en el proceso de conversión de **var_18** a una estructura. Basándonos en nuestro primer análisis, esto era de esperar, con lo cual leeremos la advertencia y completaremos la operación.



Una vez reformateada, IDA comprenderá que cualquier referencia de memoria en el bloque de 24 bytes distribuidos para **var_18** deberá referirse a un campo de la estructura. Cuando IDA encuentra dicha referencia, hace lo posible para resolver la referencia a uno de los campos definidos en la estructura de la variable. En nuestro caso, el desensamblado es automáticamente reformateado incorporando el esquema de la estructura, con lo cual lo veríamos de la siguiente forma:

```
.text:00401000 ; =============== S U B R O U T I N E =======================================
.text:00401000
.text:00401000 ; Attributes: bp-based frame
.text:00401000
.text:00401000 ; int __cdecl main(int argc, const char **argv, const char *envp)
.text:00401000 _main           proc near               ; CODE XREF: ___tmainCRTStartup+15A↓p
.text:00401000
.text:00401000 var_18          = struct_tora ptr -18h
.text:00401000 argc            = dword ptr  8
.text:00401000 argv            = dword ptr  0Ch
.text:00401000 envp            = dword ptr  10h
.text:00401000
.text:00401000                 push    ebp
.text:00401001                 mov     ebp, esp
.text:00401003                 sub     esp, 18h
.text:00401006                 mov     [ebp+var_18.field_1], 0Ah
.text:0040100D                 mov     [ebp+var_18.field_2], 14h
.text:00401013                 mov     [ebp+var_18.field_3], 1Eh
.text:00401017                 mov     [ebp+var_18.field_4], 28h
.text:0040101E                 fld     ds:dbl_40B128
.text:00401024                 fstp    [ebp+var_18.field_5]
.text:00401027                 xor     eax, eax
.text:00401029                 mov     esp, ebp
.text:0040102B                 pop     ebp
.text:0040102C                 retn
.text:0040102C _main           endp
```

La ventaja de utilizar la notación de la estructura en el desensamblado, es el mejoramiento general de lectura de éste. La utilización de los nombres de los campos en la vista reformateada proporciona una mejor comprensión de cómo son manipulados los datos en el código fuente original.

El procedimiento para formatear variables globales como estructuras es casi idéntico al de las variables de pila. Para realizarlo, seleccionas la variable o dirección que marca el inicio de la estructura y realizas la acción **Edit > Struct Var** o el atajo (**ALT-Q**), eligiendo el tipo de estructura apropiado.