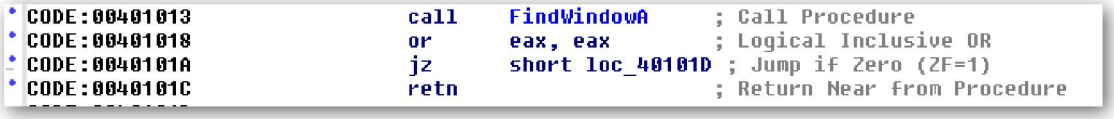
12.2.-- Aumentar los comentarios predefinidos con loadint

En el escrito 6 estudiamos el concepto de **autocomments**, recordemos que cuando lo habilitábamos nos producía en IDA poder ver los comentarios describiendo cada instrucción de lenguaje ensamblador. Un ejemplo de dichos comentarios:



El código fuente de estos comentarios predefinidos está en el archivo **Archivos de programa\IDA\ida.int**, éste contiene los comentarios clasificados primero por tipo de CPU y segundo por tipo de instrucción. Cuando están habilitados los auto comentarios, IDA busca los comentarios asociados a cada instrucción en ida.int y los muestra en el margen derecho del desensamblado.

La utilidad **loadint**, proporciona la habilidad de modificar los comentarios existentes o añadir comentarios nuevos al archivo **ida.int**. Al igual que las anteriores utilidades, loadint se documenta en un archivo **readme.txt** incluido en la distribución de loadint. Loadint también contiene los comentarios predefinidos para todos los módulos de procesador en forma de multitud de archivos **.cmt**. Modificar los comentarios existentes es un trabajo fácil, simplemente hay que localizar el archivo de comentarios asociado al procesador que nos interese, por ejemplo **pc.cmt** para **x86**, cambiando cualquier comentario que deseemos, una vez realizado ejecutamos **loadint.exe** y volvemos a crear el archivo de comentarios **ida.int** y para finalizar copiamos el archivo resultante **ida.int** en el directorio principal de IDA para que sea cargado la próxima vez que ejecutemos IDA. Veamos la creación de una base de datos:

```
E:\Archivos de programa\IDA\Loadint.v5.20>loadint.exe comment.cmt ida.int
Comment base loader. Version 2.04. Copyright (c) 1991-2007 by Ilfak Guilfanov
Output database is not found. Creating...

15958 cases, 15498 strings, total length: 512811
```

Una cosa a observar, la cual se explica en la documentación de loadint. Es que **loadint.exe** debe poder localizar el archivo **ida.hlp**, el cual está incluido en la distribución de IDA. Si no es así, nos mostrará un mensaje de error, ver siguiente, y tendremos que copiar ida.hlp en el directorio de loadint.exe.

```
E:\Archivos de programa\IDA\Loadint.v5.20> loadint.exe comment.cmt ida.int
Comment base loader. Version 2.04. Copyright (c) 1991-2007 by Ilfak Guilfanov
Can't initialize help system.
File name: 'ida.hlp', Reason: can't find file (take it from IDA distributive).
```

Otra alternativa es utilizar la opción **–n** con loadint especificando la ubicación del directorio de IDA, como se muestra a continuación:

```
E:\Archivos de programa\IDA\Loadint.v5.20>loadint.exe -n Archivos de programa\ID
A comment.cmt ida int
Comment base loader. Version 2.04. Copyright (c) 1991-2007 by Ilfak Guilfanov
Usage: loadint [-sw] <infile> <outfile>
 example: loadint comment.cmt ida.int
          -C show preprocessor output
          -n=PATH path to ida.hlp
```

El archivo **comment.cmt** sirve como archivo maestro de entrada para el proceso **loadint**. La sintaxis de dicho archivo está descrita en la documentación de loadint. El comment.cmt crea mapeados de los tipos de procesador asociados a los archivos de comentarios. Un archivo de comentarios específico a un procesador nos proporciona un mapeado de instrucciones específico y su comentario asociado. El proceso está guiado por distintos conjuntos de constantes que definen el tipo de procesador, ubicados en **comment.cmt**, y todas las posibles instrucciones de cada procesador, ubicadas en **allins.hpp**.

Si queremos añadir los comentarios predefinidos de un nuevo tipo de procesador el proceso es algo más complicado que cambiar solamente los comentarios existentes y está muy relacionado con la creación de módulos de procesador, lo cual estudiaremos más adelante. Sin profundizar mucho en los módulos de procesador, realizar los comentarios de un nuevo tipo de procesador, requiere primero crear un conjunto de constantes enumeradas (compartidas con el módulo de procesador) dentro de **allins.hpp** y se definen de forma que haya una constante para cada instrucción de nuestro interés. En segundo lugar, tendremos que crear un archivo comment con el mapeado de cada instrucción constante enumerada y asociada al texto de comentario. Tercero, deberemos definir una nueva constante para nuestro tipo de procesador (también compartida con el modulo de procesador) y crear una entrada en **comment.cmt** este mapeado asocia el tipo de procesador con el archivo de comentarios. Una vez completados estos pasos, deberás ejecutar **loadint.exe** para crear una nueva base de datos que incorpore el nuevo tipo de procesador y sus comentarios asociados.

Performance Bigundill@