Una Breve Descripción del Programa FSIT

I. Aparicio, J. V. Álvarez, J. J. Álvarez

30 de Junio de 2017

1. El programa

El lenguaje utilizado para realizar este proyecto es **Python 3**, que suele tener mucha aceptación en Informática Forense ("Digital Forensic") gracias a la cantidad de librerías existentes y su gran portabilidad: puede correr en diferentes sistemas operativos. Para no restringir el ámbito del programa se decidió usar la librería *PyQt4* en el diseño de la interfaz gráfica. Otras de las razones que nos llevó a elegir este paquete, es que sus componentes son de un alto nivel y también gozan de una gran aceptación en la comunidad.

Como esto es sólo el inicio y aspiramos a ampliar la herramienta a otros sistemas de archivos el código ha sido escrito basándose en la programación orientada a objetos. Y para que el programa sea más fácil de reutilizar se ha seguido el patrón **MVC**. Todo este esfuerzo ha permitido mejorar la usabilidad de la aplicación.

1.1. Descripción para el Usuario

La ventana principal del programa está dividida en dos marcos ("frames"): en la parte de la izquierda se ve la tabla del registro y cada celda contiene un byte en hexadecimal; el lado de la derecha visualiza la información de la celda al activarla con un "double click"(ver figura 1).

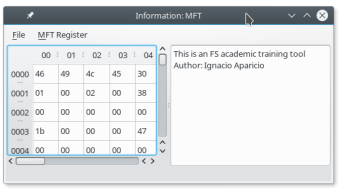

Figura 1: Ventana principal de la aplicación

Cuando la aplicación lee un registro de la tabla **MFT** analiza inmediatamente su estructura básica, es decir, determina su cabecera y todos los atributos presentes. Con esta información inicial genera el menú "MFT Register"(ver figura 2).

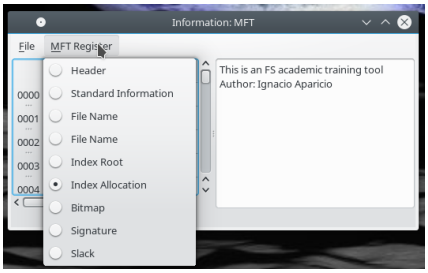

Figura 2: Menu de la estructura del fichero

Al seleccionar una de las opciones (cabecera del registro o alguno de sus atributos) se marca el correspondiente segmento en la tabla, de tal manera que cada campo aparecen con un color distinto. Cada campo corresponde a datos o valores que se almacenan en el segmento. En la figura 3 se puede ver un ejemplo concreto, que representa un atributo.

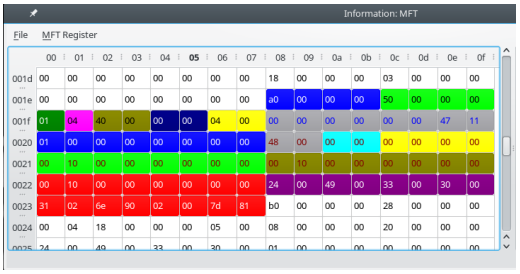

Figura 3: Atributo: Index Allocation

Por ejemplo el campo con el color verde oscuro (celda 1f:00) corresponde al "flag"que nos indica si el atributo es residente o no. Su información es visualizada en el otro marco al realizar un "double click" sobre esa misma celda (ver figura 4)

1.2. Descripción del Programa

La aplicación, como ya se ha mencionado, se ha programado orientado a objetos en el lenguaje de **Python v3**. Sin embargo toda la información que

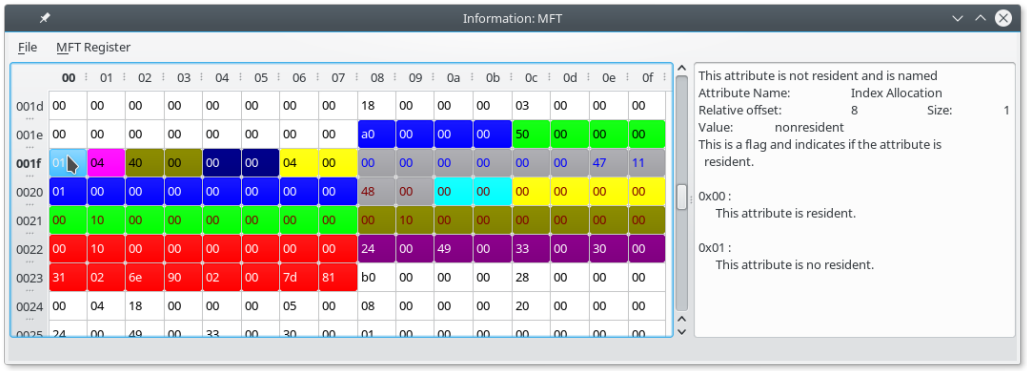

Figura 4: Seleccionado el campo: Resident Flag

se ofrece al usuario se guarda en ficheros en formato *xml*. La separación de datos y código ofrece varias ventajas: el mantenimiento es mucho más fácil y su internacionalización es sencillo de llevar a cabo.

La estructura de clases se ha creado pensando en la ampliación futura de la aplicación, de manera que con un esfuerzo razonable se puedan añadir otros sistemas de archivos como el **extN** del sistema operativo **Linux**. Así que se ha optado por la siguiente estructura (ver figura 5):

- El registro de **MFT** corresponde al fichero que se visualiza en la tabla de la aplicación.
- Se divide en segmentos: el primero es la cabecera, seguido de los atributos, que contiene el registro, la marca final de los datos y finalmente el "slack"del registro.
- Los segmentos están compuestos por campos, que representan los diferentes valores (p.e. identificadores, "flags", valores numéricos o de cadenas, etc.). El significado y la información asociada a estos campos están guardados en los ficheros *xml*.

Un típico ejemplo de los ficheros *xml* se puede ver en el código 1.

```
1 <mftheader:flags offset="22" size="2" absolute="True"
    essential="True" function="getFlagInfo" bgcolor="
    darkGray" fgcolor="white">
2 <mftheader:description type="hex" used="0x0001"
    directory="0x0002" unknown="0x0003" Unknown="0x0004"
    >
3 <mftheader:used type="hex" flag="0x0001">
```

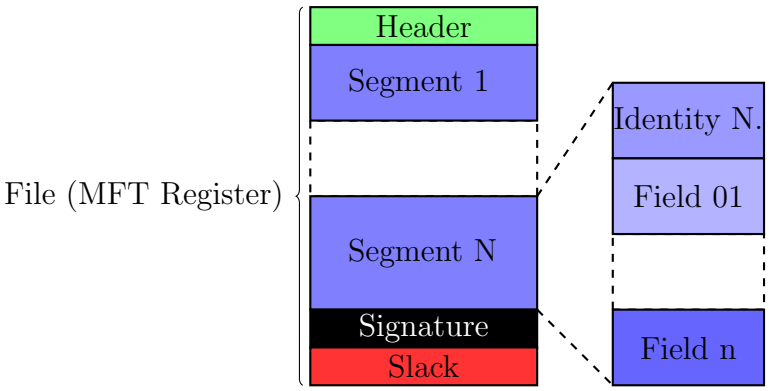

Figura 5: Estructura del fichero (ejemplo: **MFT** Register)

```
4  The record is in use.
5  </mftheader:used>
6  <mftheader:directory type="hex" flag="0x0002">
7  The record is a directory.
8  </mftheader:directory>
9  <mftheader:unknown type="hex" flag="0x0003">
10 Use is unknown.
11 </mftheader:unknown>
12 <mftheader:Unknown type="hex" flag="0x0004">
13 Use is unknown.
14 </mftheader:Unknown>
15 </mftheader:description>
16 <mftheader:info>
17 The flag give information about the use of the record
18 or it is a directory.
19 </mftheader:info>
20 </mftheader:flags>
```

Listing 1: Campo: flags

El ejemplo "Listing 1" representa un campo que como indica la primera etiqueta *flags* (ver línea 1) es un "flag". Sus atributos determinan los valores del "offset" del campo y si es absoluto o no, su tamaño, la función que el programa debe aplicarle, su color de fondo y el del texto, etc.

En este caso la función *getFlagInfo* provoca que el programa lea los atributos de la etiqueta *description* (ver línea 2) y sus valores. El nombre de los atributos se usan como etiquetas para guardar la información correspondiente.

Por último, en la línea 16 se encuentra la etiqueta *info*, que guarda la descripción del campo, donde se vuelve a comprobar de que se trata de un "flag".

Este ejemplo posee la estructura más compleja que se puede ver en los ficheros *xml* y es la razón de haber sido elegido.

1.3. Paquetes Requeridos

Para poder utilizar esta herramienta se requiere tener instalados los siguientes paquetes de **Python**, sin olvidar que deben ser compatibles con la versión 3:

- PyQt4,
- argparse, (se elimará en el futuro)
- sys, stat,
- xml.etree.ElementTree.

Referencias

[1] Carrier, Brian. *File System Forensic Analysis*. Pearson Education, Inc., 2005. ISBN 0-32-126817-2.

[2] Russon, Richard & Fledel, Yuval. NTFS Documentation, 2011. URL ftp://ftp.jaist.ac.jp/pub/sourceforge/n/nt/ntfsofuefi/NTFSReferenceDocuments/ntfsdoc.pdf.

[3] Wilkinson, Michael. NTFS Reference Sheet. URL http://www.writeblocked.org/resources/NTFS_CHEAT_SHEETS.pdf.

[4] Yasinsac, A. and Erbacher, R. F. and Marks, D. G. and Pollitt, M. M. and Sommer, P. M. Computer forensics education. 1:15–23, 2003. doi: 10.1109/MSECP.2003.1219052. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1219052.