

Questo articolo si pone l'obiettivo di analizzare la struttura del cosiddetto Cestino (Recycle Bin) dei sistemi Microsoft in modo da comprendere meglio quale sia la metodologia seguita durante l'eliminazione di files e cartell.

Contrariamente a quanto comunemente si crede, quando un file viene eliminato dal sistema, in realtà i dati non vengono immediatamente perduti, ma rimangono sul disco fino a quando non saranno sovrascritti. Questo è vero sia per i Sistemi Operativi Microsoft che per quelli Unix, anche se le procedure di recupero e ripristino dei dati variano tra i due casi.

Poiché il Cestino permette di ripristinare i files in esso contenuti nelle originali locazioni sul *file system*, è evidente che debba esservi un file che conservi i *record* delle impostazioni per l'eventuale ripristino. Tale file prende il nome di **INFO2** e risiede nella directory del Cestino, la cui locazione dipende dal tipo di Windows installato.

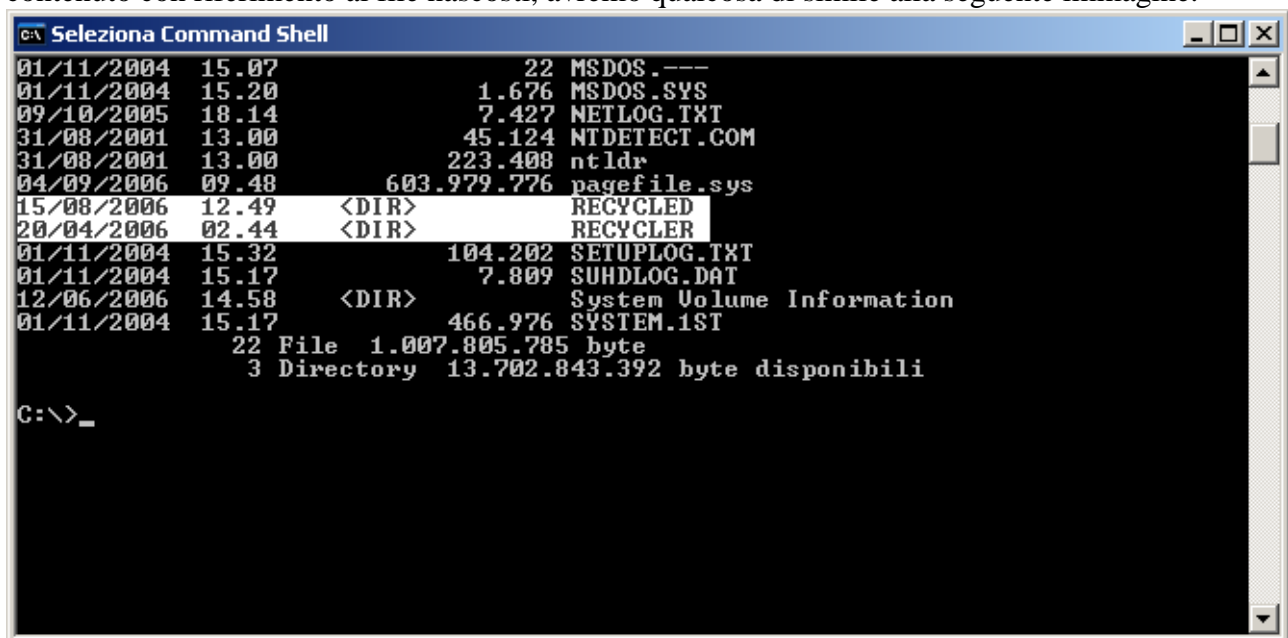
La seguente tabella descrive i possibili percorsi del Cestino nel mondo Microsoft.

Sistema Operativo	File System	Percorso del Cestino
Windows 95/98/ME	FAT	C:\Recycled\INFO2
Windows NT/2k/XP/2k3	NTFS	C:\Recycler\<USER SID>\2

Sia la directory **Recycled** che quella **Recycler** hanno impostati gli attributi *nascosto(h)* e *sistema(s)*. Nel momento in cui un file viene spostato nel Cestino, viene rinominato come **DC#.ext**, dove # è un numero intero incrementale ed univoco (il quale sarà l'indice del file **INFO2**) e .ext è l'estensione originaria del file eliminato. Ad esempio, se un file chiamato CONTO.TXT venisse eliminato, avrebbe nel Cestino un nome simile a DC1.TXT.

E' importante sottolineare come la directory del Cestino venga creata nella root di ogni volume o disco anche se trattato come unità rimovibile.

Infatti, se accediamo alla root di un volume (ad esempio C:) e digitiamo *dir /ah* per elencarne il contenuto con riferimento ai file nascosti, avremo qualcosa di simile alla seguente immagine.



```

C:\ Seleziona Command Shell
01/11/2004 15.07          22 MSDOS_---
01/11/2004 15.20        1.676 MSDOS.SYS
09/10/2005 18.14          7.427 NETLOG.TXT
31/08/2001 13.00        45.124 NTDETECT.COM
31/08/2001 13.00       223.408 ntlldr
04/09/2006 09.48     603.979.776 pagefile.sys
15/08/2006 12.49    <DIR>      RECYCLED
20/04/2006 02.44    <DIR>      RECYCLER
01/11/2004 15.32        104.202 SETUPLOG.TXT
01/11/2004 15.17          7.809 SUHDLOG.DAT
12/06/2006 14.58    <DIR>      System Volume Information
01/11/2004 15.17          466.976 SYSTEM.1ST
                22 File 1.007.805.785 byte
                3 Directory 13.702.843.392 byte disponibili

C:\>_

```

Il file **INFO2** viene automaticamente creato quando il primo file viene eliminato e spostato nel Cestino. Quando il Cestino viene svuotato del contenuto, il file **INFO2** è ripulito da tutte le informazioni di ripristino precedenti mentre il contatore univoco che funge da indice viene resettato ad 1.

Esplorando la directory **Recycler** e sempre eseguendo il comando `dir /ah`, troveremo alcune sottocartelle nominate secondo il **Sid** (Security Identifier) degli utenti che hanno usato il Cestino dopo essersi ovviamente loggati sul sistema.

Nel nostro esempio avremo:

```

C:\>cd recycler
C:\RECYCLER>dir
Il volume nell'unit  C non ha etichetta.
Numero di serie del volume: 849F-7DAA

Directory di C:\RECYCLER

File non trovato
C:\RECYCLER>dir /ah
Il volume nell'unit  C non ha etichetta.
Numero di serie del volume: 849F-7DAA

Directory di C:\RECYCLER

20/04/2006  02.44    <DIR>          .
20/04/2006  02.44    <DIR>          ..
04/09/2006  11.07    <DIR>          $-1-5-21-789336058-436374069-1202660629-1003
20/04/2006  02.50    <DIR>          $-1-5-21-789336058-436374069-1202660629-500
0 File                                0 byte
4 Directory 13.700.112.384 byte disponibili
C:\RECYCLER>

```

E' possibile ricondurre un **SID** al relativo account leggendo il valore della chiave `ProfileImagePath` localizzata nel Registro di Sistema al percorso `[HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\<SID>]`.

Esploriamo la prima directory, e lanciamo il comando `dir /a | attrib` in modo da visualizzare tutti i file contenuti in essa con i relativi attributi.

```

C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003>dir /a | attrib
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc1.ini
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc10.doc
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc11.pdf
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc12.zip
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc13.htm
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc15.htm
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc17.doc
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc18.pdf
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc19.pdf
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc2.asd
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc20.pdf
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc21.doc
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc23.reg
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc3.tmp
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc4.asd
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc5.txt
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc6.txt
A C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc7.doc
A H C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc8.doc
A H C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\Dc9.doc
SH C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\desktop.ini
A H C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003\INFO2
C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003>

```

Nella schermata sopra riportata, vediamo una serie di file nel formato gi  analizzato `dc#.ext` e per ultimo il file **INFO2** che presenta l'attributo `-h` per indicare che   nascosto.

Ora, vogliamo analizzare la struttura interna del file binario **INFO2**. Per fare ci  sarebbe possibile utilizzare il comando DOS `debug <nomefile>` seguito dal switch `-d`, ma per semplificarci la vita useremo un qualunque editor esadecimale.

Per maggiore sicurezza lavoreremo su una copia del file.

Procediamo come segue:

lanciamo dal prompt dei comandi `attrib -H INFO2` per togliere l'attributo nascosto al file e poi eseguiamo `copy INFO2 c:\temp`. Infine, reimpostiamo l'attributo nascosto con `attrib +H INFO2`. In questo modo avremo una copia del file nel percorso `c:\temp`.

Prima di addentrarci nell'analisi, dobbiamo prepararci a comprendere come i byte in formato esadecimale formino strutture dati più complesse. Apriamo quindi una parentesi teorica che ci servirà nel seguito dell'articolo.

A seconda dell'architettura del processore possiamo avere due tipi di convenzioni da seguire nel ricostruire le strutture dati:

1. Rappresentazione **Big Endian** in cui il byte più significativo (**Most Significant Byte**) di una struttura dati si trova in indirizzi di memoria più basso (i byte della struttura si leggono da sinistra verso destra).

Questa rappresentazione è tipica delle architetture SPARC, Motorola 68k e PowerPC.

2. Rappresentazione **Little Endian** in cui il byte meno significativo (**Less Significant Byte**) di una struttura dati si trova in indirizzi di memoria più bassi (i byte della struttura si leggono da destra verso sinistra).

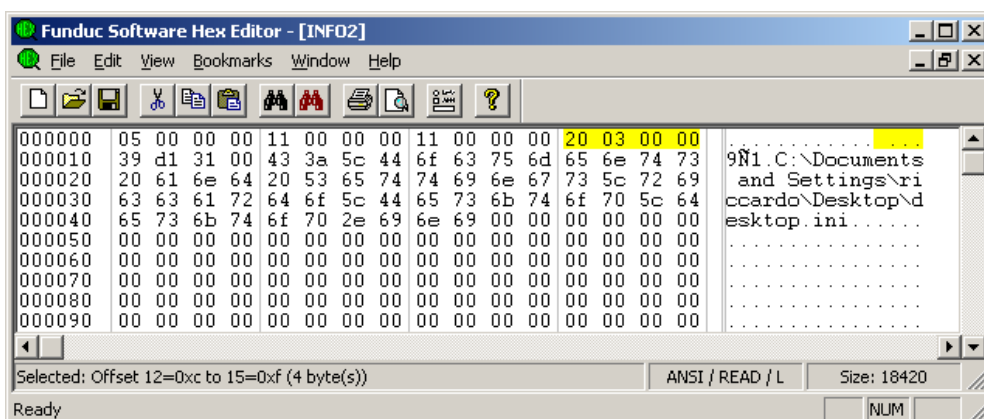
Questa rappresentazione è tipica delle architetture Intel80x86 e cloni.

Per fare un esempio concreto di quanto le due rappresentazioni siano differenti e di quali possano essere i problemi di compatibilità tra differenti architetture, ipotizziamo di avere due postazioni su una rete basata su protocolli TCP/IP; una macchina sarà dotata di processore Intel80x86 e l'altra di processore SPARC. Si supponga che le due macchine vogliano comunicare tra di loro e quindi debbano scambiarsi pacchetti secondo i protocolli stabiliti indicando l'indirizzo IP del mittente e del destinatario. Vediamo cosa accade ad un ipotetico indirizzo IP trasmesso tra i due nodi della rete con architetture del processore differenti e supporto nativo a diverse rappresentazioni dei dati. Avremo come indirizzo IP per la macchina Intel80x86 il 192.0.1.2 che diverrà in esadecimale `0x020100c0`, cioè col byte più significativo (`0xc0`) in indirizzi più alti (a destra), giacché tale macchina adotta la rappresentazione *little endian*. La macchina SPARC interpreterà la sequenza di byte inviati `0x020100c0` secondo la propria rappresentazione nativa, cioè il *big endian* (ovvero `0x02` sarà il byte più significativo), e quindi riterrà che l'indirizzo IP del trasmettente sia il 2.1.0.192 e non il 192.0.1.2

Come vedete, il risultato è completamente errato, ed è ovvio che nella realtà si provvederà ad utilizzare funzioni di conversione tra una rappresentazione e l'altra.

Ora che conosciamo meglio gli aspetti legati alla rappresentazione dei dati, adotteremo nel seguito dell'analisi una rappresentazione di tipo *little endian*.

Procediamo aprendo la copia con il nostro editor esadecimale e analizziamo la struttura del file che ci rivelerà come **INFO2** contenga le informazioni per il ripristino dei file eliminati.



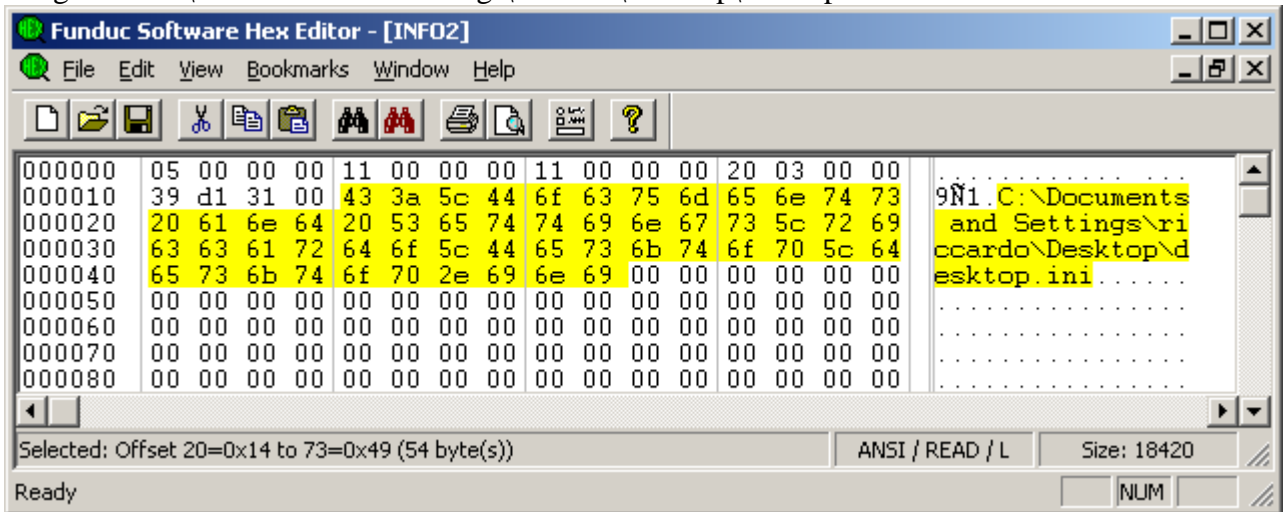
All'offset assoluti 0x0C troviamo la struttura che indica la dimensione dei singoli *records* del file, infatti il valore 0x20030000 che per la rappresentazione *little endian* deve essere interpretato 0x00000320, equivale in notazione decimale ad 800 (800 bytes).

Subito dopo l'indicazione della dimensione (offset assoluto 0x10), inizia il primo record vero e proprio che si estende lungo 800 byte fino all'offset assoluto 0x32F.

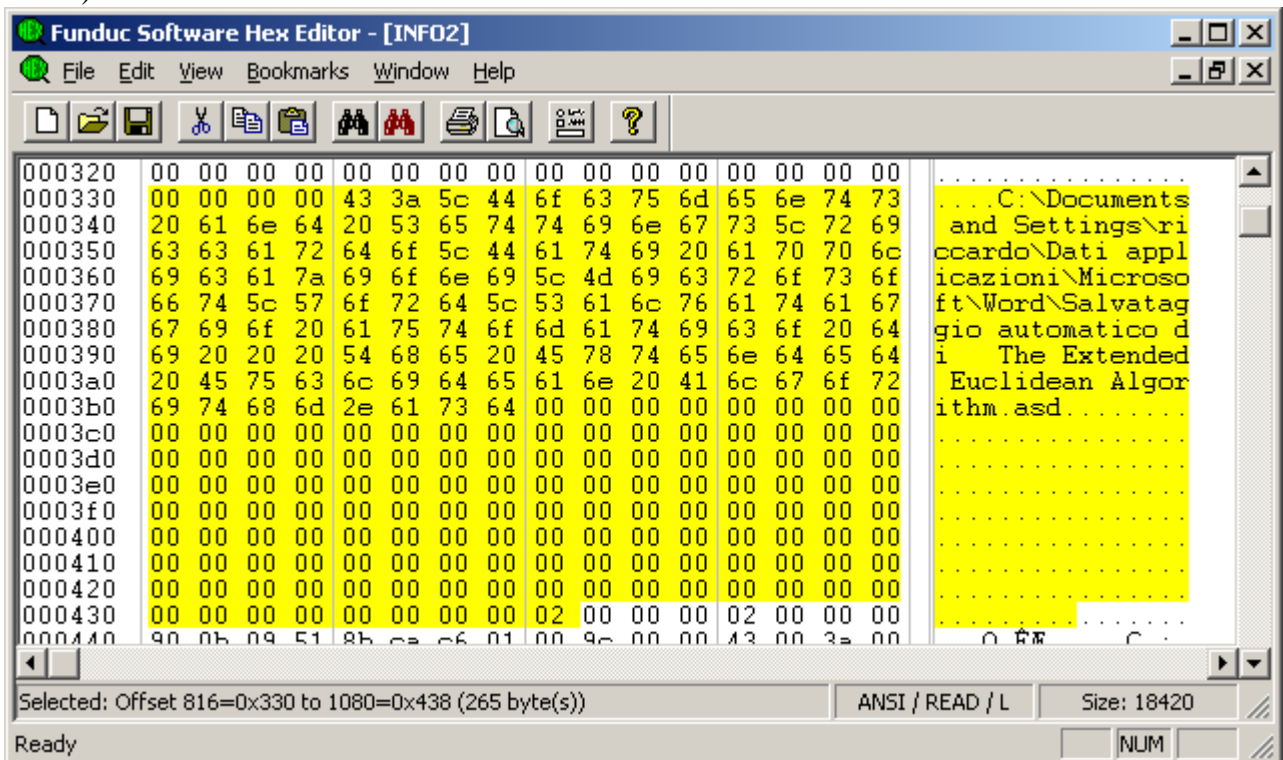
The screenshot shows the Funduc Software Hex Editor window. The main area displays a hex dump of a file. The hex values are shown in columns, and the corresponding ASCII characters are shown in the rightmost column. The hex dump is highlighted in yellow. The ASCII column shows the file path: 'C:\Documents and Settings\riccardo\Desktop\desktop.ini'. The status bar at the bottom indicates 'Selected: Offset 16=0x10 to 815=0x32f (800 byte(s))' and 'ANSI / READ / L'.

Offset	Hex	ASCII
000000	05 00 00 00 11 00 00 00 11 00 00 00 20 03 00 00
000010	39 d1 31 00 43 3a 5c 44 6f 63 75 6d 65 6e 74 73	9N1.C:\Documents
000020	20 61 6e 64 20 53 65 74 74 69 6e 67 73 5c 72 69	and Settings\ri
000030	63 63 61 72 64 6f 5c 44 65 73 6b 74 6f 70 5c 64	ccardo\Desktop\
000040	65 73 6b 74 6f 70 2e 69 6e 69 00 00 00 00 00 00	esktop.ini.....
000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000c0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000d0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000e0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000110	00 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00
000120	30 3e 17 c7 96 c2 c6 01 00 02 00 00 43 00 3a 00	0>.Ç.Æ...C:
000130	5c 00 44 00 6f 00 63 00 75 00 6d 00 65 00 6e 00	\.D.o.c.u.m.e.n
000140	74 00 73 00 20 00 61 00 6e 00 64 00 20 00 53 00	t.s.a.n.d.S
000150	65 00 74 00 74 00 69 00 6e 00 67 00 73 00 5c 00	e.t.t.i.n.g.s.\
000160	72 00 69 00 63 00 63 00 61 00 72 00 64 00 6f 00	r.i.c.c.a.r.d.o.
000170	5c 00 44 00 65 00 73 00 6b 00 74 00 6f 00 70 00	\.D.e.s.k.t.o.p
000180	5c 00 64 00 65 00 73 00 6b 00 74 00 6f 00 70 00	\.d.e.s.k.t.o.p
000190	2e 00 69 00 6e 00 69 00 00 00 00 00 00 00 00 00	.i.n.i.....
0001a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001c0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001d0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001e0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000210	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000230	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000240	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000250	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000260	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000270	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000290	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002c0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002d0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002e0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000310	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000320	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000330	00 00 00 00 43 3a 5c 44 6f 63 75 6d 65 6e 74 73	...C:\Documents
000340	20 61 6e 64 20 53 65 74 74 69 6e 67 73 5c 72 69	and Settings\ri

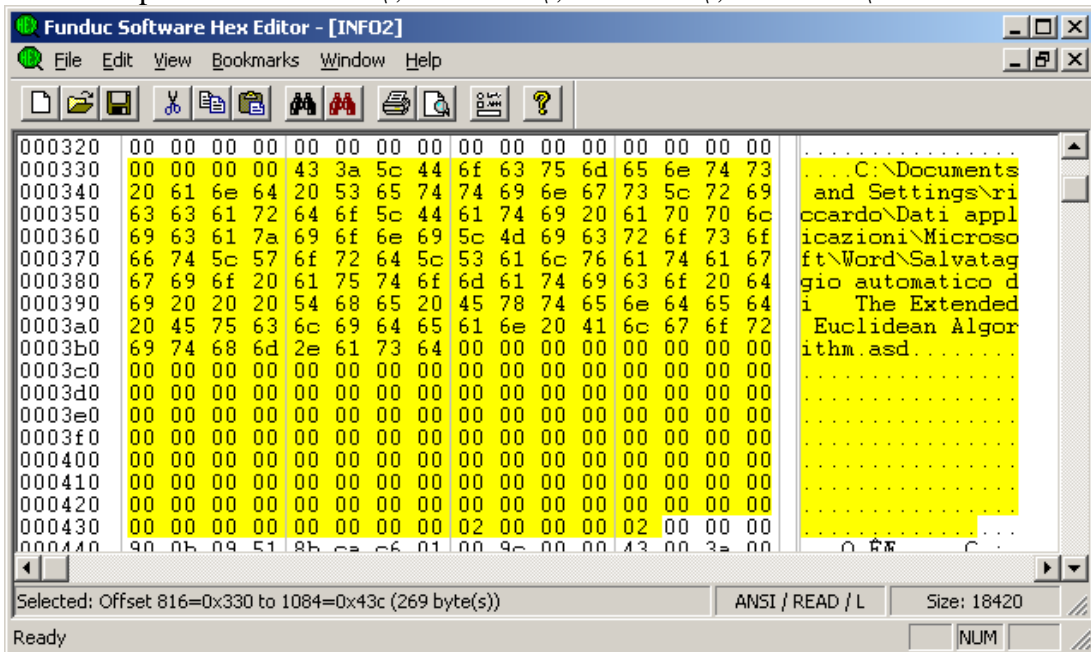
Come si vede dall'area selezionata, il nome del file eliminato è presente due volte nel *record*. C'è una versione ASCII prossima all'inizio del record ed una versione UNICODE vicina alla fine dello stesso. Prendendo in esame la versione ASCII (offset assoluto 0x14) si vede che il contenuto è il seguente: C:\Documents and Settings\riccardo\Desktop\desktop.ini



Il secondo tipo di informazione che vogliamo indagare è il numero identificativo univoco dei *record* presenti nel file **INFO2**, che si trova sempre all'offset 0x108 relativo all'inizio del *record*. Per esempio il primo record che comincia all'offset assoluto 0x10 presenta all'offset assoluto 0x118 (offset relativo 0x108) il valore 0x01 (in quanto si tratta del primo record), mentre il secondo, che inizia all'offset assoluto 0x330, presenta il valore 0x02 all'offset assoluto 0x438 (offset relativo 0x108).



Un'altra informazione importante è il numero indicante il volume o la partizione in cui risiedeva il file prima di venire eliminato. Tale dato si trova all'offset 0x10C relativo all'inizio di un record. Ad esempio, per il secondo record che comincia all'offset assoluto 0x330, il valore che ci interessa si trova all'offset assoluto 0x43C (ovvero 0x330+0x10C) che nel nostro caso vale 0x02, ovvero C:\. Infatti, lo schema prevede 0x00 = A:\, 0x01 = B:\, 0x02 = C:\, 0x03 = D:\ e così via.



Un'informazione importante, anche e soprattutto per la *forensics analysis* (ovvero l'utilizzo delle informazioni recuperate ai fini di un'indagine legale), è quella ottenuta dalla data in cui il file è stato cancellato. Infatti, il *timestamp* dell'eliminazione si trova all'offset 0x110 relativo all'inizio del *record* e si estende per una lunghezza di 8 byte. Nel nostro caso, i dati temporali relativi al secondo *record* partono all'offset assoluto 0x440 (0x330+0x110) e arrivano all'offset assoluto 0x447. Il loro valore, secondo la rappresentazione *little endian*, è uguale a **0x 01 C6 CA 8B 51 09 0B 90**.

Purtroppo, Windows salva i *timestamp* secondo il formato FILETIME calcolato in numero di ticks, con incrementi di 100ns, a partire dalle 00.00 del 1 Gennaio 1601, mentre il resto del mondo usa il formato temporale di Unix calcolato come il numero dei secondi trascorsi dalle 00.00 del 1 Gennaio 1970.

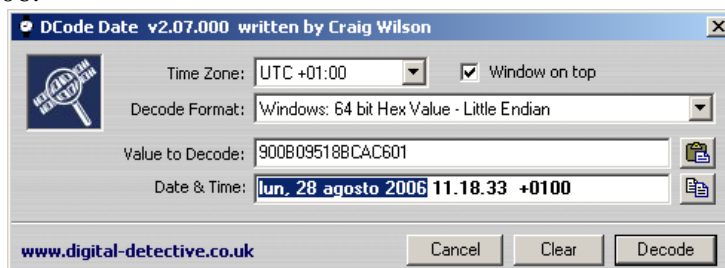
Per convertire il formato Microsoft nel formato più comune è necessario applicare la seguente formula:

$$(\text{Tempo Unix}) = A * (\text{Tempo Windows}) + B$$

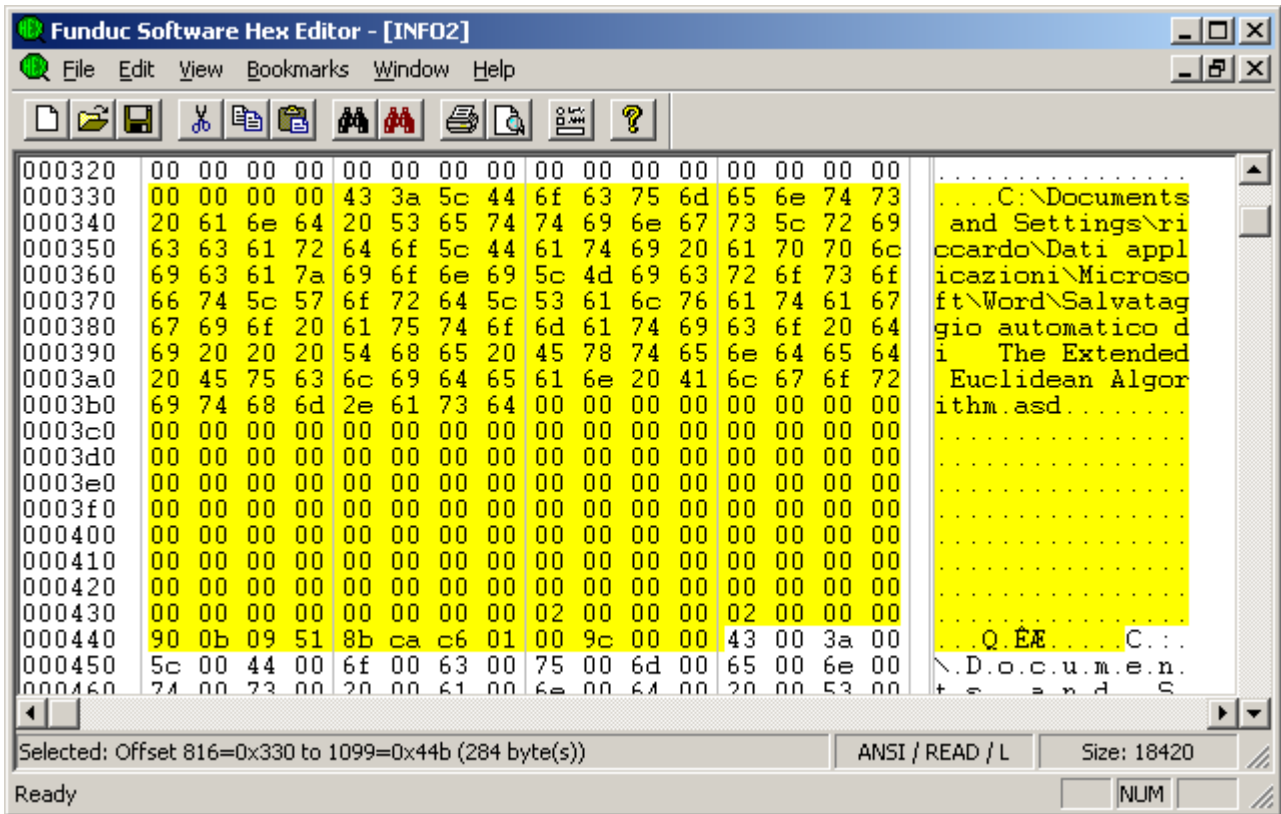
dove A vale 10^{-7} (cioè converte i nanosecondi in secondi) e B è uguale al numero di secondi tra il 1 Gennaio 1601 ed il 1 Gennaio 1970, ovvero 11644473600 secondi

Per evitare di compiere noiosi calcoli, è disponibile **Dcode**, un prezioso tool gratuitamente scaricabile da www.digital-detective.co.uk che consente di convertire in formato umano, una serie di formati tra cui quello che ci interessa (ovvero Windows:64 bit Hex Value – Little Endian).

Utilizzando questo semplice software verifichiamo che il file in questione è stato eliminato Lunedì 28 Agosto 2006.



Inoltre identifichiamo la struttura indicante la dimensione del file eliminato che si trova all'offset 0x118 relativo all'inizio del *record* e si estende per 4 bytes. Bisogna sottolineare che la dimensione riportata è da intendersi come fisica e non logica ovvero si riferisce ai settori occupati sul disco rigido e quindi è sempre uguale ad un multiplo della dimensione dei settori (512 byte). Nel caso del secondo *record*, i dati che cominciano all'offset assoluto 0x448 (0x330+0x118) e terminano all'offset assoluto 0x44B (0x448+0x04), valgono secondo la solita rappresentazione *little endian*, 0x 00 00 9C 00, ovvero 39.936 bytes (cioè 78 settori da 512 bytes ciascuno).

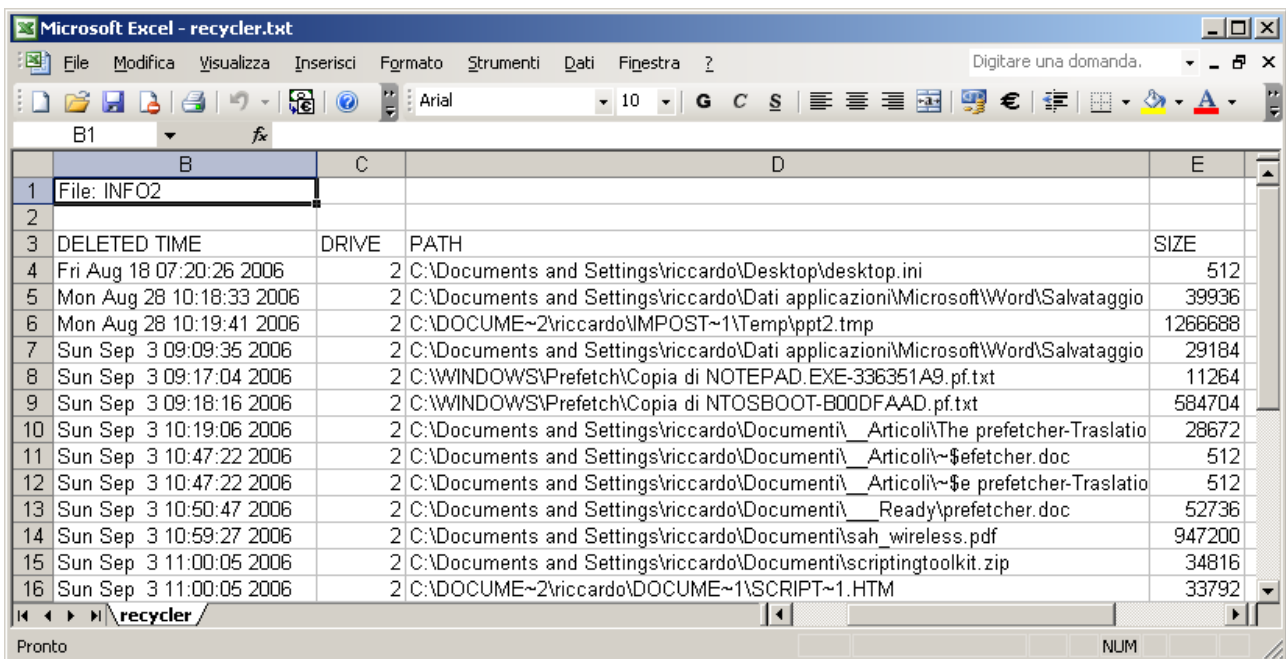


Infine, dopo aver condotto un'analisi di tipo manuale è forse utile sapere che esiste un ottimo tool chiamato **Rifiuti** distribuito gratuitamente dalla Foundstone (McAfee) che permette in pochi secondi di recuperare tutte le informazioni di cui abbiamo parlato. Scarichiamo il programma dal sito <http://www.foundstone.com/resources/proddesc/rifiuti.htm> e poniamo per ipotesi il contenuto dell'archivio compresso nel percorso locale C:\Temp\, si procede esplorando la directory del Cestino fino a trovare il file INFO2, così come già visto nella prima parte dell'articolo.

Ipotizzando di voler analizzare il Cestino del volume C: , avremo la seguente situazione al prompt dei comandi:

```
C:\RECYCLER\S-1-5-21-789336058-436374069-1202660629-1003> C:\Temp\rifiuti INFO2 >
recycler.txt
```

In questo modo, l'output del programma viene salvato su un semplice file di testo (recycler.txt) che per default riporta le informazioni in modalità delimitata da tabulazione, rendendo così possibile l'importazione in un programma tipo MS Excel che ci renda più facile l'analisi.



DELETED TIME	DRIVE	PATH	SIZE
Fri Aug 18 07:20:26 2006	2	C:\Documents and Settings\riccardo\Desktop\desktop.ini	512
Mon Aug 28 10:18:33 2006	2	C:\Documents and Settings\riccardo\Dati applicazioni\Microsoft\Word\Salvataggio	39936
Mon Aug 28 10:19:41 2006	2	C:\DOCUME~2\riccardo\IMPOST~1\Temp\ppt2.tmp	1266688
Sun Sep 3 09:09:35 2006	2	C:\Documents and Settings\riccardo\Dati applicazioni\Microsoft\Word\Salvataggio	29184
Sun Sep 3 09:17:04 2006	2	C:\WINDOWS\Prefetch\Copia di NOTEPAD.EXE-336351A9.pf.txt	11264
Sun Sep 3 09:18:16 2006	2	C:\WINDOWS\Prefetch\Copia di NTOSBOOT-B00DFAAD.pf.txt	584704
Sun Sep 3 10:19:06 2006	2	C:\Documents and Settings\riccardo\Documenti\Articoli\The prefetcher-Traslato	28672
Sun Sep 3 10:47:22 2006	2	C:\Documents and Settings\riccardo\Documenti\Articoli\~\$efetcher.doc	512
Sun Sep 3 10:47:22 2006	2	C:\Documents and Settings\riccardo\Documenti\Articoli\~\$e prefetcher-Traslato	512
Sun Sep 3 10:50:47 2006	2	C:\Documents and Settings\riccardo\Documenti\Ready\prefetcher.doc	52736
Sun Sep 3 10:59:27 2006	2	C:\Documents and Settings\riccardo\Documenti\sah_wireless.pdf	947200
Sun Sep 3 11:00:05 2006	2	C:\Documents and Settings\riccardo\Documenti\scriptingtoolkit.zip	34816
Sun Sep 3 11:00:05 2006	2	C:\DOCUME~2\riccardo\DOCUME~1\SCRIPT~1.HTM	33792

Conclusioni

Questo articolo vuole essere uno stimolo a ripensare le cose che vediamo ed utilizziamo migliaia di volte al giorno, come il Cestino di Windows, sotto una lente di ingrandimento che ci permette di coglierne gli aspetti solitamente ignorati. L'analisi capillare di qualsiasi fenomeno ci circonda è d'obbligo per chiunque voglia davvero comprendere il funzionamento intrinseco delle cose e non solo la loro apparenza.

Buon Divertimento a tutti!