

# EXT4: Bit by Bit

Hal Pomeranz

Deer Run Associates

# What's New in EXT4?

- 48-bit address space
- Uses extents instead of indirect block chains
- 64-bit nanosecond resolution timestamps
- File creation time timestamp

# Backwards Compatibility

- Backwards compatibility was a design goal
- Inodes expanded to 256 bytes:
  - Much of the first 128 bytes unchanged from EXT[23]...
  - ... except that block pointers replaced by extents
  - Extended timestamps, etc in upper 128 bytes

# Let's Make a File!

```
# echo Time for knowledge >testfile  
# touch -a -t 211101231917.42 testfile  
# touch -m -t 204005160308.19 testfile
```

No fractional seconds!

	stat	istat	debugfs
Access	2111-01-23 19:17:42.0	1974-12-17 12:49:26	1974-12-17 12:49:26.0
Modify	2040-05-16 03:08:19.0	2040-05-16 03:08:19	2040-05-16 03:08:19.0
Change	2011-03-12 07:36:13...	2011-03-12 07:36:13	2011-03-12 07:36:13...
Create	N/A	N/A	2011-03-12 07:36:04...

# Timestamps In The Inode

The screenshot shows a GHex window titled "testfile.inode - GHex" with a menu bar (File, Edit, View, Windows, Help). The main area displays hex data with annotations:

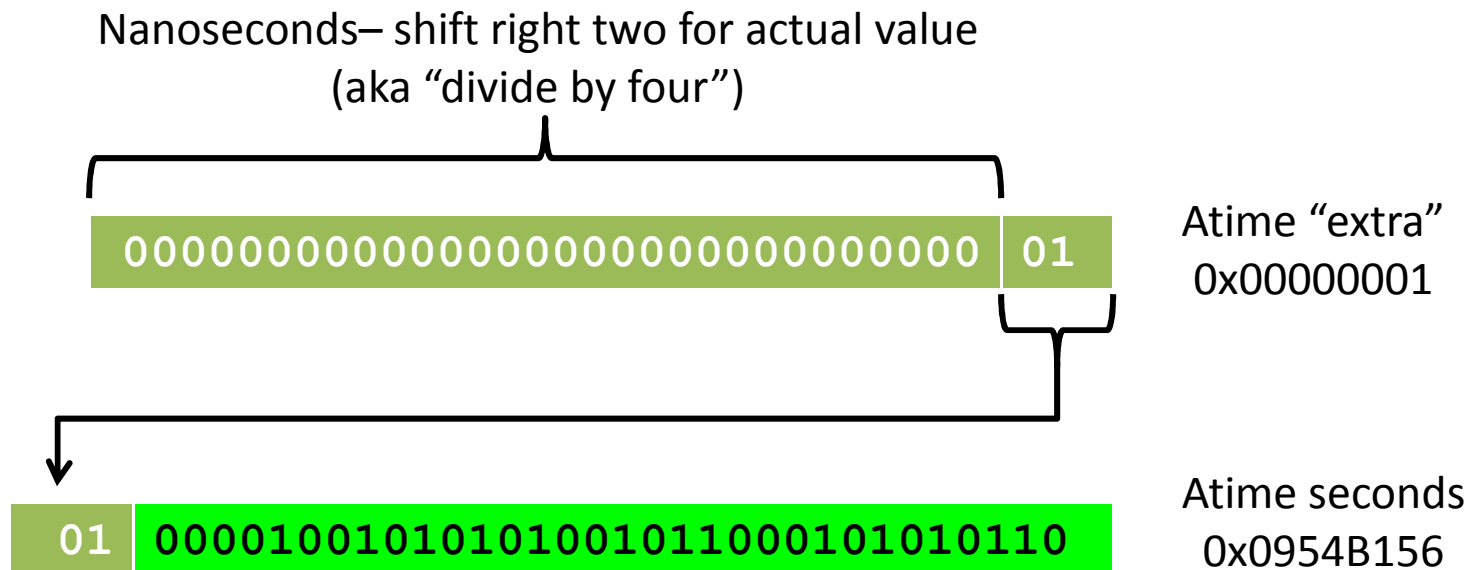
- Mtime** (Modification Time): 13 59 5E 84 (yellow highlight)
- Atime** (Access Time): 56 B1 54 09 (green highlight)
- Ctime** (Change Time): ED 92 7B 4D (magenta highlight)
- Seconds**: Two boxes labeled "Seconds" with arrows pointing to the Mtime and Atime fields.
- Creation Time (Btime)**: E4 92 7B 4D (blue highlight)
- Extra**: Two boxes labeled "Extra" with arrows pointing to the fields 18 BE FF CF and 01 00 00 00.

Other hex data visible includes: 00000000 A4 81 00 00 13 00 00 00 00 56 B1 54 09 ED 92 7B 4D, 00000010 13 59 5E 84 00 00 00 00 00 00 00 00 00 00 00 00, 00000020 00 00 08 00 01 00 00 00 01 00 04 00 00 00 00, 00000030 00 00 00 00 00 00 00 00 00 00 36 87 00 01, 00000040 00 00 Mtime Atime Ctime 0 00, 00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00, 00000060 00 00 00 00 BD 9F CF 00 00 00 00 00 00 00 00, 00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00, 00000080 1C 00 00 00 18 BE FF CF 00 00 00 00 01 00 00 00, 00000090 E4 92 7B 4D 6C F0 8A 14 00 00 00 00 00 00 00 00, 000000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00, 000000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00, 000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00, 000000D0 00 00 Seconds Creation Time (Btime) 00 00 00 00, 000000E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00, 000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00.

Offset: FF

# “Extra” – Not Just Nanoseconds!

- Only need 30 bits for nanosecond resolution
- Low-order two bits used to extend seconds field



# Extent Header (Bytes 40-51)

The screenshot shows the GHex application window titled "testfile.inode - GHex". The main display area shows a hex dump of the file's content. The Extent Header is located at offset 0x00000040. The following table represents the data shown in the hex dump:

Offset	Hex
00000000	A4 81 00 00 13 00 00 00 00 56 B1 54 09 ED 92 7B 4D
00000010	13 59 5E 84 00 00 00 00 00 00 01 00 08 00 00 00
00000020	00 00 08 00 01 00 00 00 0A F3 01 00 04 00 00 00
00000030	00 00 00 00 00 00 00 00 00 00 00 00 36 87 90 01
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080	1C 00 00 00 18 BE FF CF 00 00 00 00 01 00 00 00
00000090	E4 92 7B 4D 6C F0 8A 14 00 00 00 00 00 00 00 00
000000A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Annotations in the image point to the following fields in the Extent Header:

- Generation ID:** Points to the first four bytes (00 00 00 00) at offset 0x00000030.
- Magic Number:** Points to the next four bytes (0A F3 01 00) at offset 0x00000034.
- Number of Extents:** Points to the next two bytes (04 00) at offset 0x00000038.
- Max Poss Extents:** Points to the next two bytes (00 00) at offset 0x0000003C.
- Depth of Tree:** Points to the next two bytes (00 00) at offset 0x00000040.

The status bar at the bottom left shows "Offset: FF".

# Extent Structure

The screenshot shows a GHex window titled "testfile.inode - GHex" displaying a hex dump of an extent structure. The hex dump is organized into rows of 16 bytes each, with the first two bytes of each row representing the logical block offset. The following table represents the data shown in the hex dump:

Offset	Hex	ASCII
00000000	A4 81 00 00 13 00 00 00 56 B1 54 09 ED 92 7B 4D	.....V.T...{M
00000010	13 59 5E 84 00 00 00 00 00 00 01 00 08 00 00 00	.Y^.....
00000020	00 00 08 00 01 00 00 00 0A F3 01 00 04 00 00 00	.....6...
00000030	00 00 00 00 00 00 00 00 01 00 00 00 36 87 90 01	.....
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000080	1C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000090	E4 92 7B 4D 6C F0 8A 14 00 00 00 00 00 00 00 00	..{ML.....
000000A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

Annotations in the image point to specific fields in the hex dump:

- Logical Block Offset:** Points to the first two bytes of the third row (00 00).
- Length in Blocks:** Points to the next four bytes (00 00 00 00).
- Phys Start Addr (upper 16 bits):** Points to the next two bytes (01 00).
- Phys Start Addr (lower 32 bits):** Points to the final six bytes (00 00 36 87 90 01).

A calculation box at the bottom indicates the resulting start address:

Start Address = 0x0000 01908736 = 26249014

The status bar at the bottom left shows "Offset: FF".



# Limitations

- Only 15 bits for extent length (high bit reserved)
  - *Max extent size is 128MB* (assuming 4K blocks)
- Only 4 extents per inode

*What about large files (> 0.5GB)?*

*What about heavily fragmented files?*

# Extent Trees

The image shows a hex editor window titled 'ino-721' displaying a hex dump of an Extent Index structure. The hex dump is organized into columns of 16 bytes each, with corresponding ASCII characters on the right. Annotations include:

- One extent**: Points to the first byte of the first extent (0A F3 01 00).
- "Depth of Tree" is now one**: Points to the first byte of the first extent (0A).
- Extent Index struct**: A green box pointing to the entire first extent (0A F3 01 00 04 00 01 00).
- Logical Block Offset**: Points to the first byte of the first extent (0A).
- Phys Block Addr (lower 32 bits)**: Points to the second and third bytes of the first extent (F3 01).
- Phys Block Addr (upper 16 bits)**: Points to the fourth and fifth bytes of the first extent (00 04).
- (unused)**: Points to the sixth and seventh bytes of the first extent (00 01).
- Block Address = 0x0000 00020012 = 131090**: A box containing the calculated block address, with arrows pointing to the second and third bytes of the first extent (F3 01).

The hex dump shows the following data for the first extent (offset 00000020):

00000020	00	00	08	00	01	00	00	00	0A	F3	01	00	04	00	01	00
----------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

The ASCII column shows the corresponding characters: ..e.....N..M...M

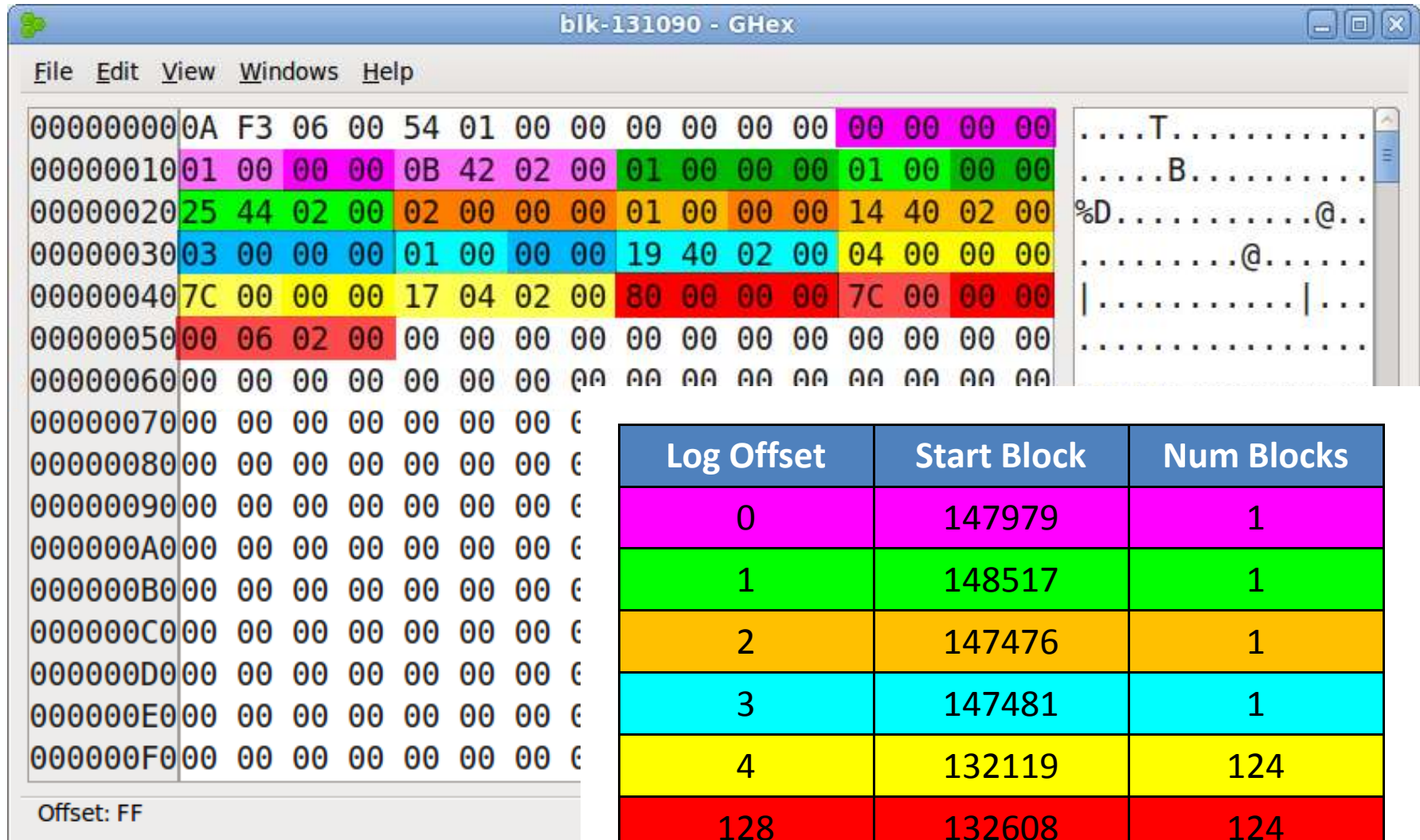
# Block 131090 (Bytes 0-255)

The image shows a hex editor window titled "blk-131090" with a menu bar (File, Edit, View, Windows, Help). The main area displays a hex dump of 256 bytes. The first six bytes (00000000 to 00000005) are highlighted in cyan and annotated with callouts:

- Magic Number for Extent Header:** Points to the first byte (0A).
- Num Extents (6):** Points to the second byte (F3).
- Max Extents (340!):** Points to the third byte (06).
- Depth of Tree (now zero):** Points to the fourth byte (00).

The rest of the hex dump shows various byte sequences, including 54 01 00 00, 00 00 00 00, 01 00 00 00, 01 00 00 00, 14 40 02 00, 03 00 00 00, 01 00 00 00, 19 40 02 00, 04 00 00 00, 7C 00 00 00, 17 04 02 00, 80 00 00 00, 7C 00 00 00, and 00 06 02 00. The right pane shows a hex-to-ASCII conversion with characters like 'T', 'B', '%D', and '@'. The status bar at the bottom indicates "Offset: FF".

# Block 131090 - Extents



The screenshot shows a GHex window titled 'blk-131090 - GHex'. The main pane displays hex data from offset 00000000 to 000000F0. Several rows are highlighted with different colors: 00000000 (magenta), 00000010 (green), 00000020 (orange), 00000030 (cyan), 00000040 (red), and 00000050 (red). The right pane shows the corresponding ASCII characters: '....T.....', '....B.....', '%D.....@.', '.....@.....', '|.....|...', and '.....'. The status bar at the bottom left shows 'Offset: FF'.

Log Offset	Start Block	Num Blocks
0	147979	1
1	148517	1
2	147476	1
3	147481	1
4	132119	124
128	132608	124

# Testing Those Numbers

```
# blkcat /dev/mapper/RD-var 147979 >ext1-blks
# blkcat /dev/mapper/RD-var 148517 >ext2-blks
# blkcat /dev/mapper/RD-var 147476 >ext3-blks
# blkcat /dev/mapper/RD-var 147481 >ext4-blks
# blkcat /dev/mapper/RD-var 132119 124 >ext5-blks
# blkcat /dev/mapper/RD-var 132608 124 >ext6-blks
# cat ext* | tr -d \\000 >newmess
# md5sum newmess /var/log/messages
8e8c9445d8ff3e17a22ef5a3034422a9  newmess
8e8c9445d8ff3e17a22ef5a3034422a9  /var/log/messages
```

# What About Inode Residue?

- What was all that junk in the inode?
  - Extents 2-4 were populated but not used
  - “Unused” bytes in extent index had data in them
- EXT4 developers were ~~lazy~~ efficient:
  - Data in inode not zeroed when extent tree needed
  - Inode extents 2-4 match block 131090 extents 2-4
  - “Unused” bytes in extent index from old extent #1

# What About File Deletion?

- How are timestamps impacted?
- What about extent structures?
- Extent trees in data blocks cleaned up?



# Post-Deletion Timestamps

ino-7210-postdelete - GHex

File Edit View Windows Help

Offset	Hex	ASCII
00000000	A0 81 65 00 00 00 00 00 F6 41 8E 4D 25 43 8E 4D	..e.....A.M%C.M
00000010	25 43 8E 4D 25 43 8E 4D 04 00 00 00 00 00 00 00	%C.M%C.M.....
00000020	00 00 08 00 01 00 00 00 0A F3 00 00 04 00 00 00	.....
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00	.....
00000040	01 00 00 00 01 00 00 00 02 00 00 00 00 00 00 00	..%D.....
00000050	01 00 00 00 14 40 00 00 01 00 00 00 00 00 00 00	.....@.....
00000060	19 40 02 00 2D 71 3A CA 00 00 00 00 00 00 00 00	.@..-q:.....
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000080	1C 00 00 00 80 81 08 77 FC B4 58 74 9C 5B 5E 36	.....w..Xt.[^6
00000090	B2 17 86 4D 8C C2 14 D7 00 00 00 00 00 00 00 00	..M.....
000000A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

Offset: FF



# Post-Deletion Extent Structs

The screenshot shows a hex editor window titled "ino-7210-postdel". The main display area contains a grid of hexadecimal data. The first column shows offsets from 00000000 to 000000FF. The data is organized into rows, with some rows highlighted in red and others in green. A callout box at the top right points to the first few bytes of the first row, stating "File size, Num Extents, and Depth of Tree zeroed". Another callout box at the bottom right points to the 19th row, stating "Extent Index untouched" and "Residue remains in unused extents".

File size, Num Extents, and Depth of Tree zeroed

Offset	Hex Data
00000000	A0 81 65 00 00 00 00 00 F6 41 8E 4D 25 43 8E 4D
00000010	25 43 8E 4D 25 43 8E 4D 04 00 00 00 00 00 00 00
00000020	00 00 08 00 01 00 00 00 0A F3 00 00 04 00 00 00
00000030	00 00 00 00 00 00 00 00 12 00 02 00 00 00 02 00
00000040	01 00 00 00 01 00 00 00 25 44 02 00 02 00 00 00
00000050	01 00 00 00 14 40 02 00 03 00 00 00 01 00 00 00
00000060	19 40 02 00 2D 71 3A CA 00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080	1C 00 00 00 80 81 08 77 FC B4 58 74 9C 5B 5E 36
00000090	B2 17 86 4D 8C C2 14 D7
000000A0	00 00 00 00 00 00 00 00
000000B0	00 00 00 00 00 00 00 00
000000C0	00 00 00 00 00 00 00 00
000000D0	00 00 00 00 00 00 00 00
000000E0	00 00 00 00 00 00 00 00
000000F0	00 00 00 00 00 00 00 00

- Extent Index untouched
- Residue remains in unused extents

Offset: FF

# Block 131090 Post-Deletion

Number of Extents zeroed

00000000	0A F3 00 00	54 01 00 00 00 00 00 00	00 00 00 00	.....T.....
00000010	00 00 00 00	00 00 00 00 01 00 00 00	00 00 00 00	.....
00000020	00 00 00 00	02 00 00 00 00 00 00 00	00 00 00 00	.....
00000030	03 00 00 00	00 00 00 00 00 00 00 00	04 00 00 00	.....
00000040	00 00 00 00	00 00 00 00 80 00 00 00	00 00 00 00	.....
00000050	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
00000060	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
00000070	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
00000080	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
00000090	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
000000A0	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
000000B0	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
000000C0	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
000000D0	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
000000E0	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....
000000F0	00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00	.....

Upper 8 bytes of extents zeroed but logical block offsets remain. Seriously, WTF?

Offset: FF

# Post-Deletion Summary

- Timestamps:
  - Deleted time (in [CMD]time fields)
  - Last access time\* and original creation time
- Extents
  - Data block address in extent index(es) [if any]
  - Unused extent structs in inode [if any]
  - Logical block offsets in extent structs
    - [allows extent sizes to be inferred in some cases]

# Wrapping Up

- Any final questions?
- Thanks for listening!

Hal Pomeranz    [hal@deer-run.com](mailto:hal@deer-run.com)

[hal@sans.org](mailto:hal@sans.org)

<http://www.deer-run.com/~hal/>

<http://computer-forensics.sans.org/blog/author/halpomeranz/>

<http://www.sans.org/security-training/instructors/Hal-Pomeranz>

[https://twitter.com/hal\\_pomeranz](https://twitter.com/hal_pomeranz)