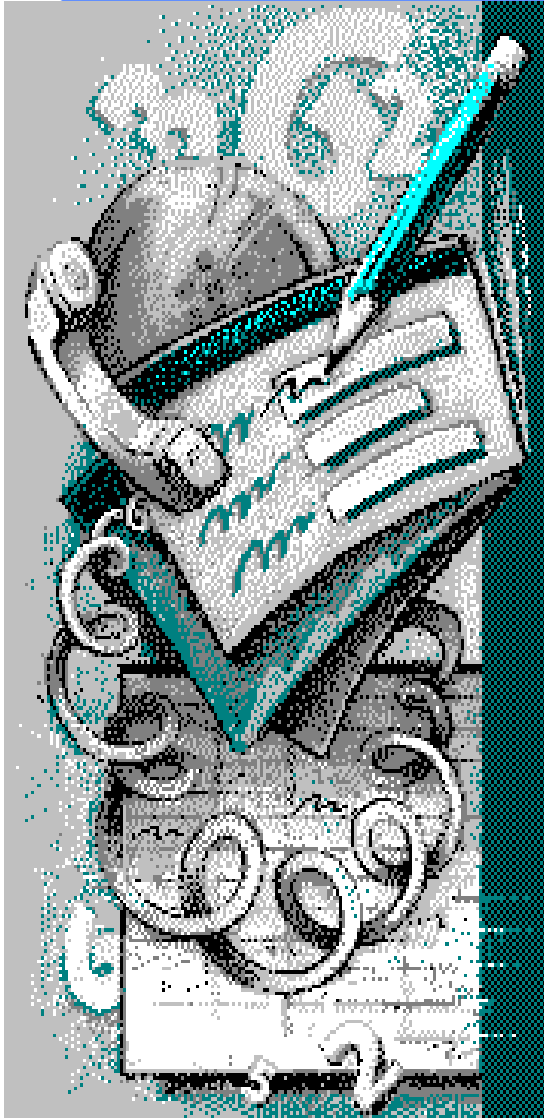


Binary Review

Binary Review for Forensic Examiner

NtWaK0



Endian-ness

- ❑ A multi-byte value is stored in memory from the lowest byte (the “little end”) to the highest byte.
- ❑ For example, the hexadecimal number 12345678 is stored as 78 56 34 12.
- ❑ This is called the little-endian format.

Endian-ness

- ❑ **Motorola and Sparc processors have the least significant byte last. A multi-byte value is stored in memory from the highest byte (the “big end”) to the lowest byte.**
- ❑ **For example, the hexadecimal number 12345678 is stored as 12 34 56 78.**
- ❑ **This is called the big-endian format.**

Integer Data Types

Format/Type	Range	Example
signed 8 bit	-128...127	FF = -1
unsigned 8 bit	0...255	FF = 255
signed 16 bit	-32,768...32,767	00 80 = -32,768
unsigned 16 bit	0...65,535	00 80 = 32,768
signed 24 bit	-8,388,608...8,388,607	00 00 80 = -8,388,608
unsigned 24 bit	0...16,777,215	00 00 80 = 8,388,608
signed 32 bit	-2,147,483,648...2,147,483,647	00 00 00 80 = -2,147,483,648
unsigned 32 bit	0...4,294,967,295	00 00 00 80 = 2,147,483,648
signed 64 bit	$-2^{63} (\approx -9 \cdot 10^{18}) \dots 2^{63} - 1 (\approx 9 \cdot 10^{18})$	00 00 00 00 00 00 00 80 = -2^{63}

Floating-Point Data Types

Type	Range	Precision [Digits]	Bytes
Float (Single)	$\pm 1.5^{-45} \dots 3.4^{38}$	7-8	4
Real	$\pm 2.9^{-39} \dots 1.7^{38}$	11-12	6
Double (Double)	$\pm 5.0^{-324} \dots 1.7^{308}$	15-16	8
Long Double (Extended)	$\pm 3.4^{-4932} \dots 1.1^{4932}$	19-20	10

Date Types

- ❑ MS-DOS Date & Time (4 bytes)
- ❑ Win32 FILETIME (8 bytes)
- ❑ OLE 2.0 Date & Time (8 bytes)
- ❑ ANSI SQL Date & Time (8 bytes)
- ❑ UNIX, C, FORTRAN (4 bytes)
- ❑ Java Date & Time (8 bytes)

MS-DOS Date & Time

Bits	Contents
0-4	Second divided by 2
5-10	Minute (0-59)
11-15	Hour (0-23 on a 24-hour clock)
16-20	Day of the month (1-31)
21-24	Month (1 = January, 2 = February, etc.)
25-31	Year offset from 1980

ANSI ASCII/IBM ASCII

Hex	Control Code	Hex	Control Code
00	Null	10	Data Link Escape
01	Start of Header	11	Device Control 1
02	Start of Text	12	Device Control 2
03	End of Text	13	Device Control 3
04	End of Transmission	14	Device Control 4
05	Enquiry	15	Negative Acknowledge
06	Acknowledge	16	Synchronous Idle
07	Bell	17	End of Transmission Block
08	Backspace	18	Cancel
09	Horizontal Tab	19	End of Medium
0A	Line Feed	1A	Substitute
0B	Vertical Tab	1B	Escape
0C	Form Feed	1C	File Separator
0D	Carriage Return	1D	Group Separator
0E	Shift Out	1E	Record Separator
0F	Shift In	1F	Unit Separator

Checksums

- ❑ A checksum is a characteristic number used for verification of data authenticity.
- ❑ Two files with equal checksums are highly likely to be equal themselves (byte by byte).
- ❑ The standard checksum is simply the sum of all bytes in a file, calculated on an 8-bit, 16-bit, 32-bit, or 64-bit accumulator.

Digests

- ❑ **Digests are strong one-way hash codes.**
- ❑ **Digests are used instead of checksums if malicious (i.e. not mere random) modifications to the original data are to be detected.**
- ❑ **It is computationally infeasible to find any data that corresponds to a given digest.**

Reference

- **Winhex Guide.**
- **Winhex is a must tool for any hard core individual.**
- **<http://www.x-ways.net/winhex/index-m.html>**