

Übung zur Vorlesung
Digitale Medien

Dr. Alexander De Luca
Ludwig-Maximilians-Universität München
Wintersemester 2013/2014

Hexeditor

Zeigt einzelne Bytes einer Datei an

z.B.

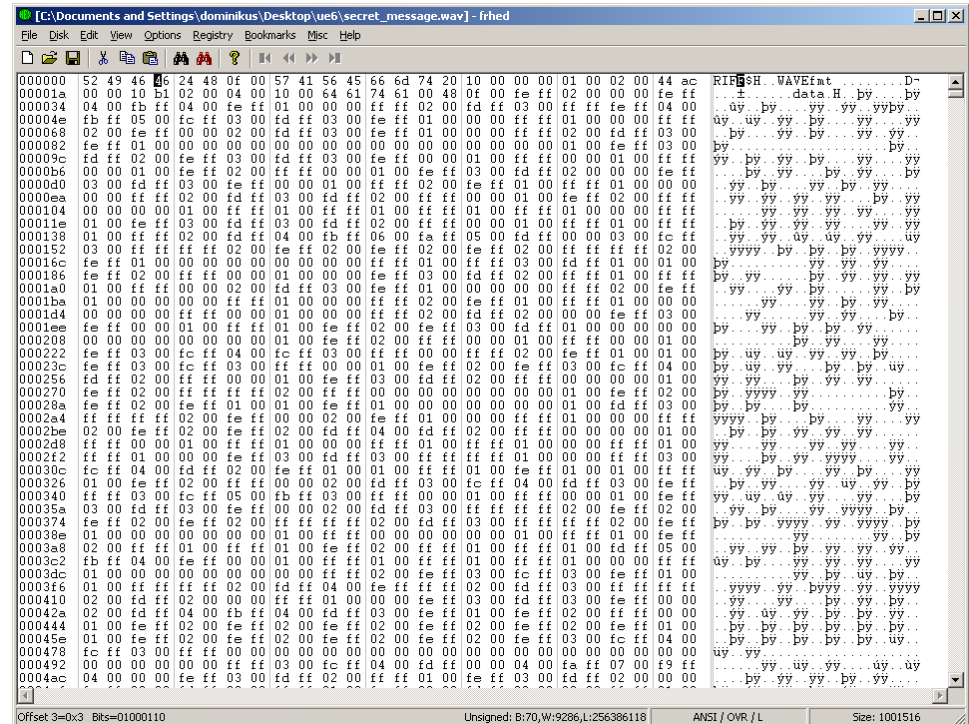
Linux: Okteta

Windows: frhed

(<http://frhed.sourceforge.net/>)

Mac OS X: HexEdit

(<http://hexedit.sourceforge.net/>)



Hexadezimaldarstellung (1)

Hexadezimal nach Dezimal

A4C₁₆ bzw. 0xA4C

$$\mathbf{C} * 16^0 + \mathbf{4} * 16^1 + \mathbf{A} * 16^2 =$$

$$12 * 1 + 4 * 16 + 10 * 256 =$$

$$12 + 64 + 2560 =$$

2636₁₀

Google

"0xXXXX to decimal"

"XXXXXX to hex"

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

[Hintergrund und Umrechnung](#)

Hexadezimaldarstellung (2)

Hexadezimal nach Dezimal

0xFF

0x7F

0x1C0

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Hexadezimaldarstellung (3)

Die Dezimalzahl **76528975** wird ins Hexadezimalsystem umgewandelt.

Gehe nach folgendem Verfahren vor:

- (1) Teile die Zahl mit Rest durch 16.
- (2) Der Divisionsrest ist die nächste Ziffer (von rechts nach links). Für Reste > 9 nimm die Buchstaben A, B, C, D, E, F
- (3) Falls der (ganzzahlige) Quotient = 0 ist, bist du fertig, andernfalls nimm den (ganzzahligen) Quotienten als neue Zahl und wiederhole ab (1).

76528975	:	16	=	4783060	Rest: 15	--> Ziffer: F
4783060	:	16	=	298941	Rest: 4	--> Ziffer: 4
298941	:	16	=	18683	Rest: 13	--> Ziffer: D
18683	:	16	=	1167	Rest: 11	--> Ziffer: B
1167	:	16	=	72	Rest: 15	--> Ziffer: F
72	:	16	=	4	Rest: 8	--> Ziffer: 8
4	:	16	=	0	Rest: 4	--> Ziffer: 4



Resultat: **48FBD4F**

Big-Endian vs. Little-Endian (1)

Byteweise Ausleserichtung:

Big-Endian: Höchster Wert zuerst

$$234 \Rightarrow 2 * 100 + 3 * 10 + 4 * 1$$

Little-Endian: Niedrigster Wert zuerst

$$234 \Rightarrow 2 * 1 + 3 * 10 + 4 * 100$$

Sprache:

24 = „twenty-four“ (Englisch – Big-Endian)

24 = „vierundzwanzig“ (Deutsch – Little-Endian)

Beispiel:

439041101 als 32-Bit Integer

Binär:

00011010 00101011 00111100 01001101

Hex:

1A 2B 3C 4D

	Big Endian			Little Endian		
Adresse	Hex	Dez	Binär	Hex	Dez	Binär
10000	1A	26	00011010	4D	77	01001101
10001	2B	43	00101011	3C	60	00111100
10002	3C	60	00111100	2B	43	00101011
10003	4D	77	01001101	1A	26	00011010

Beispiel: Wikipedia

Big-Endian vs. Little-Endian (1)

33 FA CC 00

Big-Endian: 0x33FACC00 => 872 074 240

Little-Endian: 0x00CCFA33 => 13 433 395

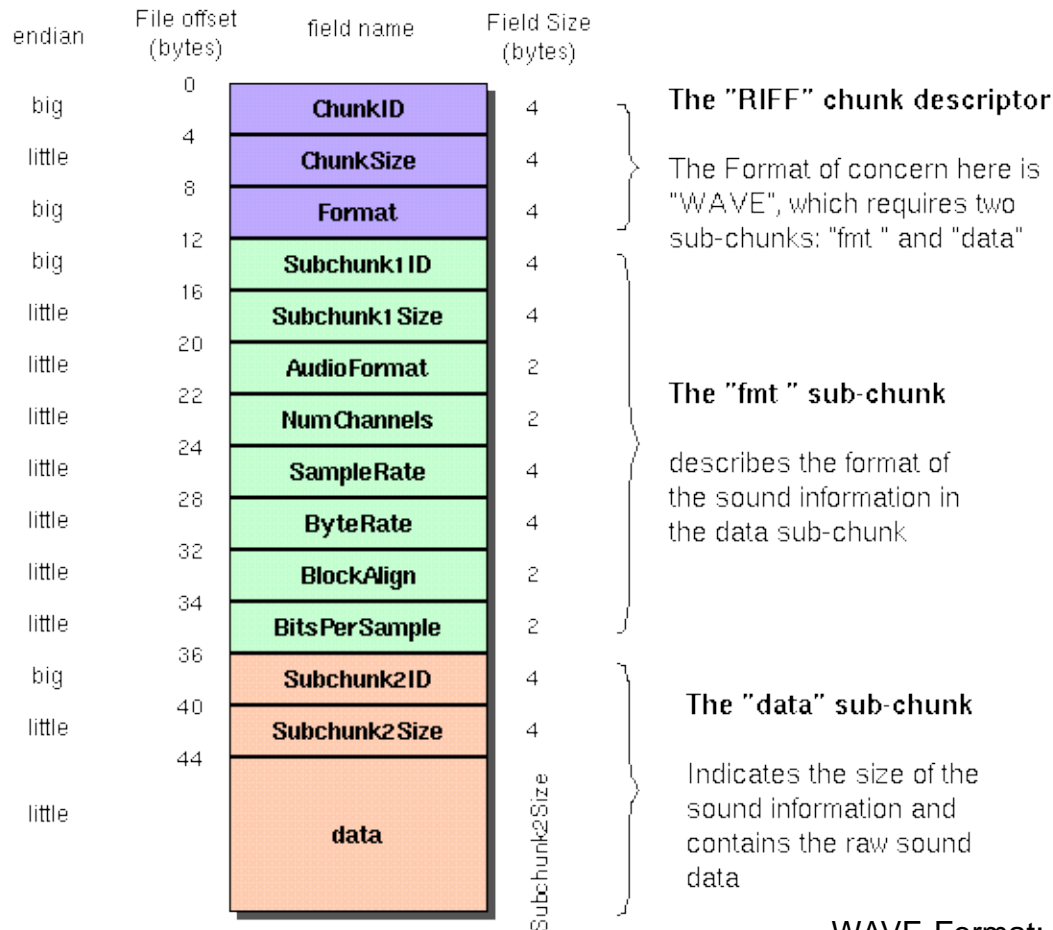
00 00 0F 11

Big-Endian: 0x00000F11 => 3 857

Little-Endian: 0x110F0000 => 286 195 712

WAVE

The Canonical WAVE file format



Beispiel:

110-220-440-880.wav

```

0000  52 49 46 46  RIFF
0004  1c 30 14 00  .0..
0008  57 41 56 45  WAVE
000c  66 6d 74 20  fmt
0010  10 00 00 00  ....
0014  01 00 01 00  ....
0018  44 ac 00 00  D-..
001c  44 ac 00 00  D-..
0020  01 00 08 00  ....
0024  64 61 74 61  data
0028  f8 2f 14 00  ø/..
002c  80 88 90 97  ....
0030  9f a7 af b6  .S~¶
0034  bd c4 cb d1  ÆÄË
0038  d7 dd e2 e7  xÿäç
    
```

Meta-Daten aus iTunes:

```

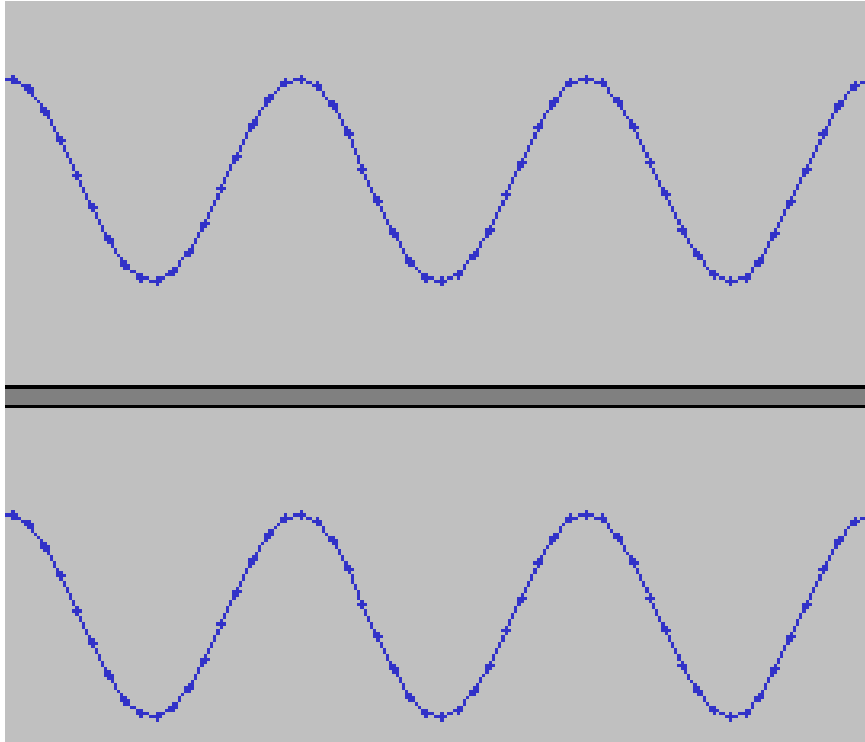
Art: WAV-Audiodatei
Größe: 44 KB
Abtastrate: 44,100 kHz
Kanäle: Mono
Datenformat: 8 Bit
    
```

WAVE-Format:

<http://www.sonicspot.com/guide/wavefiles.html>

<https://ccrma.stanford.edu/courses/422/projects/WaveFormat/> 8

Interleaving



8-Bit Stereo

Die beiden Stereospuren werden abwechselnd in der Datei abgelegt

00024	64	61	da
00026	74	61	ta
00028	80	a9	.@
0002a	03	00	..
0002c	80	7f	..
0002e	95	95	...
00030	a8	a8	...
00032	b7	b7	..
00034	be	be	%%
00036	bf	bf	??
00038	b8	b8	??
0003a	aa	aa	??
0003c	97	97	..
0003e	82	82	..
00040	6c	6c	ll
00042	58	58	XX
00044	49	49	II
00046	41	41	AA
00048	40	40	@@
0004a	46	46	FF
0004c	54	54	TT
0004e	66	66	ff
00050	7b	7b	{{
00052	91	91	''
00054	a5	a5	??
00056	b4	b4	''
00058	bd	bd	%%
0005a	bf	bf	??
0005c	b9	b9	11
0005e	ad	ad	--
00060	9b	9b	..
00062	86	86	..
00064	70	70	pp
00066	5c	5c	\\