

<http://de.wikipedia.org/wiki/Partitionstabelle>

zeigt in einer sehr anschaulichen Grafik Einträge im MasterBootRecord einer Festplatte.

http://de.wikipedia.org/wiki/GUID_Partition_Table

erklärt das neuere GPT-Schema.

Außerdem las ich weitere Beiträge in de.wikipedia.org und <http://www.bsdforen.de>, wo ich auch Fragen stellte.

Den Erklärungen folgend habe ich mir versuchsweise einen USB-Stick mit GPT eingerichtet und will meine Resultate nun mit den Artikeln vergleichen und die Ergebnisse ein wenig grafisch aufgepeppt wiedergeben.

Ausgangslage ist ein FreeBSD partitionierter USB-Stick mit einer EXT2FS-Partition. Diese Information ist hier unwichtig, sie dient nur dem Vergleich:

DOS Partition Table

Offset Sector: 0

Units are in 512-byte sectors

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Primary Table (#0)
01:	-----	0000000000	0000000062	0000000063	Unallocated
02:	00:00	0000000063	0015952544	0015952482	FreeBSD (0xA5)
03:	-----	0015952545	0015954943	0000002399	Unallocated

Mit Anfang und Ende bezeichne ich die Lage der Sektoren gemäß ihrer Nummerierung, Offset 0 ist dann der Anfang.

In *kursiver* Schrift und nach einem # füge ich Kommentare und Erläuterungen in den Ausgaben ein. Bestimmte Stellen hebe ich hervor.

Es gibt Zahlen (vornehmlich in der der Byte-Weisen Darstellung des Inhaltes der Sektoren), die in Hexadezimaler Schreibweise angegeben sind. Dies erwähne ich nicht gesondert und erkläre es nicht. Hier gehe ich einfach davon aus, dass jeder weiß, worum es geht, der sich für diese Darstellung interessiert.

Nun richte ich auf diesem Stick eine FreeBSD-ufs Partition mit GPT ein. Dann betrachte ich immer nur den primären GPT-Eintrag, den sekundären GPT Eintrag ignoriere ich.

Die Darstellung des neu eingerichteten USB-Sticks:

GUID Partition Table (EFI)

Offset Sector: 0

Units are in 512-byte sectors

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Safety Table
01:	-----	0000000000	0000000033	0000000034	Unallocated
02:	Meta	0000000001	0000000001	0000000001	GPT Header
03:	Meta	0000000002	0000000033	0000000032	Partition Table
04:	00	0000000034	0015954910	0015954877	GPTUFS
05:	-----	0015954911	0015954943	0000000033	Unallocated

Eine andere Darstellung der gleichen Verhältnisse:

```
Geom name: da0
state: OK
fwheads: 255
fwsectors: 63
last: 15954910                                #15954910 x 512 = 8168913920
first: 34                                       #34 x 512 = 17408
entries: 128
scheme: GPT
Providers:
1. Name: da0p1
   Mediasize: 8168897024 (7.6G)                #8168897024 = 15954877 x 512
   Sectorsize: 512                             #15954910 - 15954877 = 33 x 512
   Mode: r0w0e0
   rawuuid: 91788346-5a40-11e0-8959-0019d1f83c93
   rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
   label: GPTUFS
   length: 8168897024
   offset: 17408
   type: freebsd-ufs
   index: 1
   end: 15954910
   start: 34
Consumers:
1. Name: da0
   Mediasize: 8168931328 (7.6G)                #8168931328 = 15954944 x 512
   Sectorsize: 512                             #15954944 - 34 = 15954910
   Mode: r0w0e0
```

Ein Blick auf diese Zahlen zeigt:
Das Medium ist insgesamt **8168931328B** groß,
was bei einer Sektorengroße von **512B**
(8168931328 / 512 =) **15954944** Sektoren entspricht.

Ein Sektor ist gleich ein Block, oder hier nun ein LBA.
Die ersten 34 LBA enthalten im ersten Block den MBR (= LBA0).
Dieser wird im sekundären GPT-Eintrag am Ende der Platte nicht gespeichert.

Der primäre GPT-Eintrag belegt die ersten **34**,
der sekundäre die letzten **33** Sektoren.¹
Der sekundäre ist eine Sicherungskopie des primären Eintrages ohne Block 0.

In der Reihenfolge ist also (1 + 33 =) 34 + 15954877 + 33 = 15954944
Oder: (512 + 16896 =) 17408 + 8168897024 + 16896 = 8168931328

Dies kann sehr gut in der folgenden Ausgabe dargestellt werden. Hier wird
beginnend mit Block 0 gezählt.

Der erste LBA (=512B) ist also Block 0.
Es gibt insgesamt 15954944 Blöcke, der letzte ist Block 15954943.

¹Es wird manchmal auch so dargestellt, dass der Primäre Eintrag 33 Sektoren groß ist und von LBA1 bis LBA33 reicht. Der erste LBA, Block 0, wird dann nicht zum GPT gezählt.

```

LBA 0
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
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 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 |.....|
00000100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 |.....|
000001c0 01 00 ee ff ff ff 01 00 00 00 ff 73 f3 00 00 00 |.....s...|
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa |.....U.|

```

```

00          1B      80=bootfähig, 00=nicht bootfähig
01 01 00    3B      CHS-Eintrag des ersten Sektors
ee          1B      Partitionstyp: ee = this legacy MBR is followed by an EFI header
ff ff ff    3B      CHS-Eintrag des letzten Sektors
01 00 00 00 4B      Startsektor (relativ zum Anfang der Festplatte)
ff 73 f3 00 4B      Anzahl der Sektoren in der Partition

```

ff 73 f3 : $16741363 \times 512 = 8571577856$?
 Es werden hier mehr Sektoren gezählt, als es gibt?
 Die Zahlen müssen umgedreht, also f3 73 ff gelesen werden.
 Das sind dann 15954943 Sektoren. Startsektor ist 1. Also 15954944 Insgesamt.

Diese Technik ist durchaus üblich und wird uns noch mehrmals begegnen. Die Hintergründe dazu kenne ich nicht.

LBA 1	Data 1								Data 2								Comment
00000200	45	46	49	20	50	41	52	54	00	00	01	00	5c	00	00	00	EFI PART....\...
00000210	ee	ae	79	2c	00	00	00	00	01	00	00	00	00	00	00	00	...y,.....
00000220	ff	73	f3	00	00	00	00	00	22	00	00	00	00	00	00	00	.s.....".....
00000230	de	73	f3	00	00	00	00	00	0f	00	65	de	3f	5a	e0	11	.s.....e.?Z..
00000240	89	59	00	19	d1	f8	3c	93	02	00	00	00	00	00	00	00	.Y....<.....
00000250	80	00	00	00	80	00	00	00	05	07	2b	08	00	00	00	00+.....
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000310	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003b0	00	00	00	00	00	00	00	00	00	00	00	00					

Die GUID müsste der zuvor ausgegebenen rawuuid entsprechen. Diese ist aber nicht komplett gleich: rawuuid: 91788346-5a40-11e0-8959-0019d1f83c93. Wie hier von den Bytes zu der Ausgabe gelangt wird, weiß ich nicht.

Wieso Anzahl der Partitionen und Größe des Eintrages hier auf 80 stehen, weiß ich nicht. Die Länge des Eintrages sollte immer 128 Bytes (= 80 Hex) sein und es kann maximal 128 Partitionen geben ($32 \text{ LBA} \times 512 / 128 = 128$).

```

LBA 2
00000400 b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000410 46 83 78 91 40 5a e0 11 89 59 00 19 d1 f8 3c 93 |F.x.@Z...Y....<.|
00000420 22 00 00 00 00 00 00 00 de 73 f3 00 00 00 00 00 |".....s.....|
00000430 00 00 00 00 00 00 00 00 47 00 50 00 54 00 55 00 |.....G.P.T.U.|
00000440 46 00 53 00 00 00 00 00 00 00 00 00 00 00 00 |F.S.....|
00000450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000004a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000004b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000004c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000004d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000004e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000004f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000500 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000510 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000520 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000540 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000550 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000580 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000590 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000005a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000005b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000005c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000005d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000005e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000005f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000600 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|

```

```

b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b Partitionstyp (ID)
46 83 78 91 40 5a e0 11 89 59 00 19 d1 f8 3c 93 GUID der Partition
22 00 00 00 00 00 00 00 Beginn der Partition
de 73 f3 00 00 00 00 00 Ende der Partition
00 00 00 00 00 00 00 00 Attribute
47 00 50 00 54 00 55 00 46 00 53 00 ... 72 Bytes G.P.T.U.F.S... Partitionsname

```

Die Partitionstyp - ID entspricht der zuvor ausgegebenen rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b, die Stellen stimmen nicht überein, es wird teilweise wieder rückwärts gelesen. 516E7CB6-6ECF-11D6-8FF8-00022D09712B ist die ID für eine Unix File System (UFS) Partition.

Die Partition beginnt Bei Block **34** und das ist **22** Hexadezimal. Der Offset in Byte ist dann also **17408** (34 x 512) oder **4400** in Hexadezimal.

Die Partition endet am letzten benutzbaren Block, es ist ja die einzige vorhandene Partition. Wie üblich die Bytes von recht nach links ordnen.

Was für Attribute?

Als Namen hatte ich GPTUFS vergeben. Offenbar wird nach jedem Zeichen eine Lücke gelassen. Bei 72 Byte könnten dann 36 Zeichen für den Namen benutzt werden?

Nur Zeilen, mit Inhalt, erstes Beispiel (von eben):

-> hexdump -C gpt_35b

00000000	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 0
*				
000001b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 01	
000001c0	01 00 ee ff ff ff 01 00	00 00 ff 73 f3 00 00 00s...	
000001d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 1
*				
000001f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 55 aaU...	
00000200	45 46 49 20 50 41 52 54	00 00 01 00 5c 00 00 00	EFI PART....\...	
00000210	ee ae 79 2c 00 00 00 00	01 00 00 00 00 00 00 00	..y,.....	L B A 2
00000220	ff 73 f3 00 00 00 00 00	22 00 00 00 00 00 00 00	.s.....".....	
00000230	de 73 f3 00 00 00 00 00	0f 00 65 de 3f 5a e0 11	.s.....e.?Z...	
00000240	89 59 00 19 d1 f8 3c 93	02 00 00 00 00 00 00 00	.Y....<.....	
00000250	80 00 00 00 80 00 00 00	05 07 2b 08 00 00 00 00+.....	L B A 2
00000260	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
*				
00000400	b6 7c 6e 51 cf 6e d6 11	8f f8 00 02 2d 09 71 2b	. nQ.n.....-.q+	
00000410	46 83 78 91 40 5a e0 11	89 59 00 19 d1 f8 3c 93	F.x.@Z...Y....<.	L B A 2
00000420	22 00 00 00 00 00 00 00	de 73 f3 00 00 00 00 00	".....s.....	
00000430	00 00 00 00 00 00 00 00	47 00 50 00 54 00 55 00G.P.T.U...	
00000440	46 00 53 00 00 00 00 00	00 00 00 00 00 00 00 00	F.S.....	
00000450	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 2
*				
00004600				

Nach Änderung und Anlegen der zweiten Partition (siehe auch folgende Seite):

hexdump -C gpt2_35b

00000000	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 0
*				
000001b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 01	
000001c0	01 00 ee ff ff ff 01 00	00 00 ff 73 f3 00 00 00s...	
000001d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 1
*				
000001f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 55 aaU...	
00000200	45 46 49 20 50 41 52 54	00 00 01 00 5c 00 00 00	EFI PART....\...	
00000210	b4 c6 58 90 00 00 00 00	01 00 00 00 00 00 00 00	..X.....	L B A 2
00000220	ff 73 f3 00 00 00 00 00	22 00 00 00 00 00 00 00	.s.....".....	
00000230	de 73 f3 00 00 00 00 00	0f 00 65 de 3f 5a e0 11	.s.....e.?Z...	
00000240	89 59 00 19 d1 f8 3c 93	02 00 00 00 00 00 00 00	.Y....<.....	
00000250	80 00 00 00 80 00 00 00	21 aa 36 f8 00 00 00 00!.6.....	L B A 2
00000260	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
*				
00000400	b6 7c 6e 51 cf 6e d6 11	8f f8 00 02 2d 09 71 2b	. nQ.n.....-.q+	
00000410	46 83 78 91 40 5a e0 11	89 59 00 19 d1 f8 3c 93	F.x.@Z...Y....<.	L B A 2
00000420	22 00 00 00 00 00 00 00	d0 0b 9a 00 00 00 00 00	".....	
00000430	00 00 00 00 00 00 00 00	47 00 55 00 54 00 45 00G.U.T.E...	
00000440	4d 00 49 00 4e 00 45 00	00 00 00 00 00 00 00 00	M.I.N.E.....	
00000450	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 2
*				
00000480	ba 7c 6e 51 cf 6e d6 11	8f f8 00 02 2d 09 71 2b	. nQ.n.....-.q+	
00000490	d7 b6 3c 2b 8d 5c e0 11	94 61 00 19 d1 f8 3c 93	..<+. \...a....<.	
000004a0	d1 0b 9a 00 00 00 00 00	de 73 f3 00 00 00 00 00s.....	L B A 2
000004b0	00 00 00 00 00 00 00 00	54 00 45 00 53 00 54 00T.E.S.T...	
000004c0	32 00 50 00 41 00 52 00	54 00 5f 00 54 00 79 00	2.P.A.R.T...T.y...	
000004d0	70 00 65 00 2d 00 5a 00	46 00 53 00 00 00 00 00	p.e.-.Z.F.S.....	
000004e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	L B A 2
*				
00004600				

Wesentliche Unterschiede und interessante Stellen sind farblich markiert.

Dazu die passenden Ausgaben der anderen Tools, wie wir sie schon gesehen haben.
Verwendete Namen (Label) sind vollkommen willkürlich, ebenso die Größen und Typen.

Alle Einträge sind einfach zu verstehen und ich erspare mir weitere Kommentare.
Der Sinn dieser Beschreibung ist mit der Darstellung auf der vorhergehenden Seite schon erfüllt. Die nächsten Ausgaben sind einfach zu kopieren und könnten für zukünftige Fragestellungen vielleicht interessant werden. Deshalb stelle ich sie ein.

```
gpart list da0
Geom name: da0
state: OK
fwheads: 255
fwsectors: 63
last: 15954910
first: 34
entries: 128
scheme: GPT
Providers:
1. Name: da0p1
  Mediasize: 5168913920 (4.8G)
  Sectorsize: 512
  Mode: r0w0e0
  rawuuid: 91788346-5a40-11e0-8959-0019d1f83c93
  rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
  label: GUTEMINE
  length: 5168913920
  offset: 17408
  type: freebsd-ufs
  index: 1
  end: 10095568
  start: 34
2. Name: da0p2
  Mediasize: 2999983104 (2.8G)
  Sectorsize: 512
  Mode: r0w0e0
  rawuuid: 2b3cb6d7-5c8d-11e0-9461-0019d1f83c93
  rawtype: 516e7cba-6ecf-11d6-8ff8-00022d09712b
  label: TEST2PART_Type-ZFS
  length: 2999983104
  offset: 5168931328
  type: freebsd-zfs
  index: 2
  end: 15954910
  start: 10095569
Consumers:
1. Name: da0
  Mediasize: 8168931328 (7.6G)
  Sectorsize: 512
  Mode: r0w0e0
```

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Safety Table
01:	-----	0000000000	0000000033	0000000034	Unallocated
02:	Meta	0000000001	0000000001	0000000001	GPT Header
03:	Meta	0000000002	0000000033	0000000032	Partition Table
04:	00	0000000034	0010095568	0010095535	GUTEMINE
05:	01	0010095569	0015954910	0005859342	TEST2PART_Type-ZFS
06:	-----	0015954911	0015954943	0000000033	Unallocated

Die zweite Partition wurde angelegt, nachdem die bestehende verkleinert worden war. Die enthaltenen Daten waren dabei nicht beschädigt worden. Der Label (Name) war vorab schon von GPTUFS nach GUTEMINE geändert und dabei die Partition von UFS2 nach EXT2 formatiert worden. Dies spielt für die Betrachtung an dieser Stelle keine Rolle, es erklärt den unterschiedlichen Namen bei der späteren Ausgabe des Codes.

Beide Partitionen wurden mit einem anderen Typ angelegt, als das spätere Dateisystem tatsächlich wurde. Die Angabe des Typs scheint also nicht wesentlich zu sein. Eine FreeBSD-UFS und FreeBSD-ZFS Partition haben auch den gleichen rawtype Eintrag: 516e7cba-6ecf-11d6-8ff8-00022d09712b und sind daran nicht zu unterscheiden. Eine Partition wurde von mir in EXT2 und die andere in FAT32 formatiert. Das bereitete keinerlei Probleme.

Verwirrend sind die Label.

Neben den Namen für die Partitionen im GPT-Code, können viele Dateisysteme auch Label vergeben. Dann gibt es vielleicht zwei Namen, für die gleiche Partition. Außerdem benutzt mein Betriebssystem noch die GPT-ID (rawuuid) der Partition und bindet letztlich darüber ein.

So wurde mein Stick von der benutzten Automatik nun gleich als sechs verschiedene Geräte erkannt. Jeweils einmal den Namen des GPT-Labels, dann den Namen des Filesystem-Labels und schließlich die GPT-ID für jede Partition. Letztere wurde dann von der Automatik benutzt, um die Dateisysteme einzubinden:

```
/dev/gptid/2b3cb6d7-5c8d-11e0-9461-0019d1f83c93 on /media/STILLDOS (msdosfs,
local, nosuid)
/dev/gptid/91788346-5a40-11e0-8959-0019d1f83c93 on /media/GUTEMINE (ext2fs, local,
nosuid)
```

Die Gerätedatei wird also anhand der rawuuid angelegt, der Mountpoint benutzt den Label des Dateisystems, der in dem EXT2FS identisch zum GPT-Label ist, beim FAT32 (MSDOSFS) jedoch den Namen STILLDOS beim Formatieren erhielt.

Für die Betrachtung des GPT-Schemas ist die letzte Information unwichtig. Sie beschreibt den Umgang eines Betriebssystems und einer installierten Automatik zur Geräteerkennung mit diesem Schema.

Im folgenden Beispiel gehe ich nun „beyond the limits“, lege also mehr als die vier Partitionen an, die im älteren MBR-Schema direkt das Maximum waren (damit meine ich die so genannten Primären Partitionen. Durch eine bestimmte Technik wurde es ermöglicht, sogenannte Erweiterte Partitionen hinzuzufügen und damit auch die Grenzen des alten MBR zu sprengen).

Mit dem GPT lege ich nun acht Partitionen an, die jeweils 750k groß sein sollen und dann eine neunte, die den Rest meines Stick einnehmen kann. Dazu die bereits bekannten Ausgaben, Kommentare kann ich mir dabei sparen:

Providers:

1. Name: da0p1
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: 5ac7243a-5d30-11e0-bc93-0019d1f83c93
rawtype: c12a7328-f81f-11d2-ba4b-00a0c93ec93b
label: eins
length: 768000
offset: 17408
type: efi
index: 1
end: 1533
start: 34
2. Name: da0p2
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: 8d3ddc85-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: zwei
length: 768000
offset: 785408
type: freebsd-ufs
index: 2
end: 3033
start: 1534
3. Name: da0p3
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: 94306f03-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: drei
length: 768000
offset: 1553408
type: freebsd-ufs
index: 3
end: 4533
start: 3034
4. Name: da0p4
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: 9c494578-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: vier
length: 768000
offset: 2321408
type: freebsd-ufs
index: 4
end: 6033
start: 4534
5. Name: da0p5
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: a5ddab1c-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: fuenf
length: 768000
offset: 3089408
type: freebsd-ufs
index: 5

```
end: 7533
start: 6034
6. Name: da0p6
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: ad95fec5-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: sechs
length: 768000
offset: 3857408
type: freebsd-ufs
index: 6
end: 9033
start: 7534
7. Name: da0p7
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: b4332e14-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: sieben
length: 768000
offset: 4625408
type: freebsd-ufs
index: 7
end: 10533
start: 9034
8. Name: da0p8
Mediasize: 768000 (750K)
Sectorsize: 512
Mode: r0w0e0
rawuuid: bc324a30-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: acht
length: 768000
offset: 5393408
type: freebsd-ufs
index: 8
end: 12033
start: 10534
9. Name: da0p9
Mediasize: 8162753024 (7.6G)
Sectorsize: 512
Mode: r0w0e0
rawuuid: c5a5ddeb-5d30-11e0-bc93-0019d1f83c93
rawtype: 516e7cb6-6ecf-11d6-8ff8-00022d09712b
label: neun
length: 8162753024
offset: 6161408
type: freebsd-ufs
index: 9
end: 15954910
start: 12034
Consumers:
1. Name: da0
Mediasize: 8168931328 (7.6G)
Sectorsize: 512
Mode: r0w0e0
```

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Safety Table
01:	-----	0000000000	0000000033	0000000034	Unallocated
02:	Meta	0000000001	0000000001	0000000001	GPT Header
03:	Meta	0000000002	0000000033	0000000032	Partition Table
04:	00	0000000034	0000001533	0000001500	eins
05:	01	0000001534	0000003033	0000001500	zwei
06:	02	0000003034	0000004533	0000001500	drei
07:	03	0000004534	0000006033	0000001500	vier
08:	04	0000006034	0000007533	0000001500	fuenf
09:	05	0000007534	0000009033	0000001500	sechs
10:	06	0000009034	0000010533	0000001500	sieben
11:	07	0000010534	0000012033	0000001500	acht
12:	08	0000012034	0015954910	0015942877	neun
13:	-----	0015954911	0015954943	0000000033	Unallocated

-> hexdump -C gpt9_35b

```

00000000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000001b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 01 |.....|
000001c0  01 00 ee ff ff ff 01 00 00 00 ff 73 f3 00 00 00 |.....s....|
000001d0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000001f0  00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa 00 |.....U..|
00000200  45 46 49 20 50 41 52 54 00 00 01 00 5c 00 00 00 |EFI PART....\...|
00000210  08 df dd ec 00 00 00 00 01 00 00 00 00 00 00 00 |.....|
00000220  ff 73 f3 00 00 00 00 00 22 00 00 00 00 00 00 00 |.s.....".....|
00000230  de 73 f3 00 00 00 00 00 ac 3d 7f 23 30 5d e0 11 |.s.....=.#0]..|
00000240  bc 93 00 19 d1 f8 3c 93 02 00 00 00 00 00 00 00 |.....<.....|
00000250  80 00 00 00 80 00 00 00 80 58 c0 9c 00 00 00 00 |.....X.....|
00000260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000400  28 73 2a c1 1f f8 d2 11 ba 4b 00 a0 c9 3e c9 3b |(|s*.....K...>. ;|
00000410  3a 24 c7 5a 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |:$.Z0].....<..|
00000420  22 00 00 00 00 00 00 00 fd 05 00 00 00 00 00 00 |".....|
00000430  00 00 00 00 00 00 00 00 65 00 69 00 6e 00 73 00 |.....e.i.n.s..|
00000440  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000480  b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000490  85 dc 3d 8d 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |..=.0].....<..|
000004a0  fe 05 00 00 00 00 00 00 d9 0b 00 00 00 00 00 00 |.....|
000004b0  00 00 00 00 00 00 00 00 7a 00 77 00 65 00 69 00 |.....z.w.e.i..|
000004c0  00 00 00 00 00 00 00 00 0a 00 00 00 00 00 00 00 |.....|
*
00000500  b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000510  03 6f 30 94 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |.o0.0].....<..|
00000520  da 0b 00 00 00 00 00 00 b5 11 00 00 00 00 00 00 |.....|
00000530  00 00 00 00 00 00 00 00 64 00 72 00 65 00 69 00 |.....d.r.e.i..|
00000540  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000580  b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000590  78 45 49 9c 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |xEI.0].....<..|
000005a0  b6 11 00 00 00 00 00 00 91 17 00 00 00 00 00 00 |.....|
000005b0  00 00 00 00 00 00 00 00 76 00 69 00 65 00 72 00 |.....v.i.e.r..|
000005c0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000600  b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000610  1c ab dd a5 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |....0].....<..|
00000620  92 17 00 00 00 00 00 00 6d 1d 00 00 00 00 00 00 |.....m.....|
00000630  00 00 00 00 00 00 00 00 66 00 75 00 65 00 6e 00 |.....f.u.e.n..|
00000640  66 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |f.....|
00000650  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*

```

```

00000680 b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000690 c5 fe 95 ad 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |...0].....<.|
000006a0 6e 1d 00 00 00 00 00 00 49 23 00 00 00 00 00 00 |n.....I#.....|
000006b0 00 00 00 00 00 00 00 00 73 00 65 00 63 00 68 00 |.....s.e.c.h.|
000006c0 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |s.....|
000006d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000700 b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000710 14 2e 33 b4 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |...3.0].....<.|
00000720 4a 23 00 00 00 00 00 00 25 29 00 00 00 00 00 00 |J#.....%).....|
00000730 00 00 00 00 00 00 00 00 73 00 69 00 65 00 62 00 |.....s.i.e.b.|
00000740 65 00 6e 00 00 00 00 00 00 00 00 00 00 00 00 00 |e.n.....|
00000750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000780 b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000790 30 4a 32 bc 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |0J2.0].....<.|
000007a0 26 29 00 00 00 00 00 00 01 2f 00 00 00 00 00 00 |&)...../.....|
000007b0 00 00 00 00 00 00 00 00 61 00 63 00 68 00 74 00 |.....a.c.h.t.|
000007c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000800 b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000810 eb dd a5 c5 30 5d e0 11 bc 93 00 19 d1 f8 3c 93 |...0].....<.|
00000820 02 2f 00 00 00 00 00 00 de 73 f3 00 00 00 00 00 00 |./.....s.....|
00000830 00 00 00 00 00 00 00 00 6e 00 65 00 75 00 6e 00 |.....n.e.u.n.|
00000840 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00004600

```

Es werden also einfach fortlaufend die Bereiche gefüllt, die für die Partitionen reserviert sind. Dabei ändert sich nichts (wesentliches) in LBA0 und LBA1.

Nun interessierte mich noch, wie es aussieht, wenn ich mein System (FreeBSD) von einem GPT-Schema booten will. Für die i386 und amd64 Architektur unterstützt mein System kein EFI. Es wird mit einem BIOS gebootet, das auf diesen Rechnern (noch) Standard ist.

BIOS-Systeme booten aber mit MBR.

Bei meinem System wird daher ein spezieller Bootcode in den LBA0 geschrieben, in dem die Anweisung steht, von der bootfähigen Partition 1 zu booten. In diese Partition 1 wird dann ein weiterer Bootcode geschrieben, der GPT versteht und also als Mittler zwischen MBR und GPT-Partitionierung dient.

So kann mein System mit GPT-Schema und BIOS booten. Es gelten dabei aber die Einschränkungen, dass die Partition mit dem Kernel kleiner als 2TB ist, unter 2TB liegt. Bei Festplatten größer 2TB wird also außer der Partition 1 mit dem speziellen GPT-Bootcode noch eine Partition 2 kleiner 2TB nötig, die den Kernel enthält.

Das sieht dann etwa so aus:

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Safety Table
01:	-----	0000000000	0000000033	0000000034	Unallocated
02:	Meta	0000000001	0000000001	0000000001	GPT Header
03:	Meta	0000000002	0000000033	0000000032	Partition Table
04:	00	0000000034	0000000161	0000000128	gptboot
05:	01	0000000162	0015954910	0015954749	freebsd-ufs
06:	-----	0015954911	0015954943	0000000033	Unallocated

```

.
.
.
*... BOOTCODE im LBA0 (MBR)
00000140 6f 74 20 6c 6f 61 64 65 72 00 4d 69 73 73 69 6e |ot loader.Missin|
00000150 67 20 62 6f 6f 74 20 6c 6f 61 64 65 72 00 01 00 |g boot loader...|
00000160 00 00 00 00 00 00 9d 6b bd 83 41 7f dc 11 be 0b |.....k..A.....|
00000170 00 15 60 b8 4f 0f 90 90 90 90 90 90 90 90 90 |...`.O.....|
00000180 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 |.....|
*
000001b0 90 90 90 90 90 90 90 90 00 00 00 00 00 00 80 01 |.....|
000001c0 01 00 ee ff ff ff 01 00 00 00 ff 73 f3 00 00 00 |.....s.....|
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa |.....U..|
00000200 45 46 49 20 50 41 52 54 00 00 01 00 5c 00 00 00 |EFI PART....\...|
00000210 3d 72 21 d9 00 00 00 00 01 00 00 00 00 00 00 00 |=r!.....|
00000220 ff 73 f3 00 00 00 00 00 22 00 00 00 00 00 00 00 |.s....."......|
00000230 de 73 f3 00 00 00 00 00 10 91 fa f4 19 5e e0 11 |.s.....^..|
00000240 bf 11 00 19 d1 f8 3c 93 02 00 00 00 00 00 00 00 |.....<.....|
00000250 80 00 00 00 80 00 00 00 75 00 b6 fc 00 00 00 00 |.....u.....|
00000260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000400 9d 6b bd 83 41 7f dc 11 be 0b 00 15 60 b8 4f 0f |.k..A.....`.O.|
00000410 c4 13 57 33 1a 5e e0 11 bf 11 00 19 d1 f8 3c 93 |..W3.^.....<.|
00000420 22 00 00 00 00 00 00 00 a1 00 00 00 00 00 00 00 |".....|
00000430 00 00 00 00 00 00 00 00 67 00 70 00 74 00 62 00 |.....g.p.t.b.|
00000440 6f 00 6f 00 74 00 00 00 00 00 00 00 00 00 00 |o.o.t.....|
00000450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000480 b6 7c 6e 51 cf 6e d6 11 8f f8 00 02 2d 09 71 2b |.|nQ.n.....-.q+|
00000490 ed 72 69 4c 1a 5e e0 11 bf 11 00 19 d1 f8 3c 93 |.riL.^.....<.|
000004a0 a2 00 00 00 00 00 00 de 73 f3 00 00 00 00 00 00 |.....s.....|
000004b0 00 00 00 00 00 00 00 00 66 00 72 00 65 00 65 00 |.....f.r.e.e.|
000004c0 62 00 73 00 64 00 2d 00 75 00 66 00 73 00 00 00 |b.s.d.-.u.f.s...|
000004d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00004400 31 c9 8e c1 8e d9 8e d1 bc 00 7c bb 63 7c 8b 77 |1.....|.c|.w|
00004410 0a 01 de 89 f0 c1 e8 04 83 e6 0f 8e d8 83 c6 f0 |.....|
00004420 b8 00 0a bf f0 ff 8e c0 fd 89 f9 41 f3 a4 8e d9 |.....A....|
00004430 8e c1 8b 4f 0a 89 de bf 00 90 01 ce 01 cf 4e 4f |...O.....NO|
00004440 f3 a4 fc fa 49 74 14 e4 64 a8 02 75 f7 b0 d1 e6 |....It..d..u....|
00004450 64 e4 64 a8 02 75 fa b0 df e6 60 fb 88 16 00 09 |d.d..u....`......|
*... BOOTCODE in Partition 1.
.
.
.

```

Wie eine GPT-Tabelle eines Microsoft-Systems aussieht, kann ich nicht sagen und habe auch kein solches System, um damit zu spielen. Wenn solche Systeme von einem BIOS-PC booten wollen, muss die benutzte Technik ganz ähnlich sein. Bios-Systeme booten mit MBR.

Die folgende Information habe ich dazu gelesen, kann sie nicht selbst bestätigen: Windows kann GPT-Tabellen nur mit dem EFI-Verfahren booten. Daher unterstützt Windows den Boot von Festplatten von >2TB nur in Kombination mit EFI. EFI findet sich anstatt eines BIOS meist auf teurer Hardware. Mhmm. Vielleicht möchte Microsoft keine Technik benutzen, die es ermöglicht, auf billiger (nicht EFI) Hardware Windows auf Festplatten >2TB zu installieren? Vielleicht möchte man auch die Verbreitung des (U)EFI anstatt des betagten BIOS fördern? Das bleibt Spekulation.

FreeBSD, Linux und andere können (mit gewissen Einschränkungen) GPT mit und ohne EFI nutzen.