

SD-CARD's structure is below;

This is simplified.

Please refer <http://www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx> about details.

<i>FAT system</i>	<i>Sector Number</i>
<i>MBR</i>	<i>0x0</i>
<i>BPB</i>	<i>firstSectorNumbers</i>
<i>FAT</i>	<i>FirstSectorNumbers + reservedSector</i>
<i>RDE</i>	<i>FirstSectorNumbers + reservedSector + (numberOfFATS X sectorsPerFAT)</i>
<i>User Area</i>	<i>FirstSectorNumbers + reservedSector + (numberOfFATS X sectorsPerFAT) + (rootEntries X 0x20) / bytePerSector</i>

BPB's address = firstSectorNumbers X 0x200

Other area's address of SD-CARRD = Sector Number X bytePerSectors.

RDE's size for SDHC is zero, because RDE for SDHC is not system area.

RDE for SDHC is within User Area and it has cluster-chain same as files.

First RDE for SDHC generally exist next to FAT area.

sd_viewer's buffer is end of Hub RAM.

<i>Hub RAM</i>	<i>address</i>
<i>Forth kernal</i>	<i>0x0</i>
<i>dictionary</i>	
	<i>here</i>
<i>sd_buf</i>	<i>dictend - 0x200</i>
	<i>dictend</i>

Sample is 2G(SDSC:kingston)

MBR(Master Boot Record) Sector:0x0 Address:0x0

firstSectorNumbers

1C0	0C	00	06	38	F8	B8	89	00	00	00	77	9F	3A	00	00	00	...	8	...	w	:	...
1D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
1E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
1F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA			U	

BTB(BIOS Parameter Block) Sector:0x89 Address:0x11200

<i>numberOfFATS</i>										<i>bytePerSector</i>										<i>sectorPerCluster</i>										<i>reservedSectors</i>									
<i>rootEntries</i>										<i>sectorsPerFAT</i>										<i>reservedSectors</i>																			
0		1		2		3		4		5		6		7		8		9		A		B		C		D		E		F									
000	EB	00	90	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00	02	40	01	00	@	...										
010	02	00	02	00	00	F8	EB	00	3F	00	40	00	89	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	@	...				
020	77	9F	3A	00	80	00	29	2D	9D	76	62	4E	4F	20	4E	41	w	:	...)	-	.	v	b	N	O	N	A	ME	FAT16							
030	4D	45	20	20	20	20	46	41	54	31	36	20	20	20	00	00																							
040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																							
										<i>bigTotalSectors</i>																													

Refer WORD'collect_infi' and WORD'info'.

FAT(File Allocation Table) Sector:0x8A Address:0x11400

<i>Cluster0x0</i>		<i>Cluster0x1</i>		<i>Cluster0x2</i>		<i>Cluster0x3</i>		<i>Cluster0x4</i>		<i>Cluster0x5</i>		<i>Cluster0x6</i>		<i>Cluster0x7</i>																			
0		1		2		3		4		5		6		7		8		9		A		B		C		D		E		F			
000	F8	FF	FF	FF	03	00	04	00	05	00	06	00	07	00	08	00	...																
010	09	00	0A	00	0B	00	0C	00	0D	00	0E	00	0F	00	10	00																
020	11	00	12	00	13	00	14	00	FF	FF	FF	FF	FF	FF	FF	FF																
030	FF	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00																

Sample FAT is FAT16.

FAT16's cluster number occupy 2byte.

Cluster number0 and Cluster number1 are reserved on sd-card system.

User's start-cluster is 2.

Each file has first cluster number inside each file's entry on RDE.

Its first cluster number point out datas inside FAT.

Each data of FAT point out next cluster number.

If its contents is 'FFFF', file data end at its cluster.

In case of FAT32(SDHC) occupy 4byte as cluster number.

But its number's valid value is 28bit.

RDE(Root Directory Entry) Sector:0x260 Address:0x4C000

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Entry of file
000	E5	45	57	46	49	4C	45	58	54	58	54	00	00	00	00	20	.EWFFILEEXTXT....
010	27	36	00	00	00	00	00	20	27	36	02	00	78	00	00	00	'6.....'6..x...
020	E5	59	5F	50	4F	53	20	20	54	58	54	00	00	00	00	00	.Y_POS TXT.....
030	41	3B	00	00	00	00	00	00	41	3B	03	00	7B	01	00	00	A;.....A;...{...

Top data'E5' for entry of file is deleted file.

Top data'00' for entry of file is end of file.

File's entry has 32byte(0x20).

There are 16 entry inside 1-sector(generally 512bytes).

FAT16 has 512(0x200) entry. $512 / 16 = 32$ sectors

FAT32 can have many file entry, because its RDE has structure of cluster-chain.

SFN(ShortFileName)

SFN(ShortFileName)										NTRes									
Name					Extension					Attribute					CreateTimeTenth				
1C0	E5	44	5F	43	41	52	44	20	50	44	46	20	00	59	5C	A4	.D_CARD	PDF	.Y\.
1D0	3E	3D	3E	3D	00	00	7C	85	11	3B	56	00	62	D4	06	00	>=>=.. ..;V.b...		
1E0	41	53	4C	49	43	45	52	20	46	20	20	20	18	52	7C	A4	ASLICER	F	.R .
1F0	3E	3D	55	3D	00	00	20	6F	EA	3C	64	00	8B	0B	00	00	>=U=..o.<d.....		

CreateDate

LastAccessDate

FirstClusterNumberHigh

WriteDate

WriteTime

FirstClusterNumberLow

CreateTime

SFN & LFN

When filename exceeds 8-characters, LFN is made.

LFN entry has 13-character. Total LFN entry is 20(decimal).

LFN's max character is 255-characters.

```

Input Address(hex) > 4c800

      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000 E5 48 49 4B 41 57 7E 31 20 20 20 10 00 0E 2A 5E .HIKAW~1    ...*^
010 51 3D 51 3D 00 00 2B 5E 51 3D 14 00 00 80 00 00 Q=Q=...+^Q=.....
020 E5 65 00 64 00 64 00 69 00 6E 00 0F 00 05 67 00 .e.d.d.i.n....g.
030 00 00 FF FF FF FF FF FF FF FF 00 00 FF FF FF FF .....
040 E5 B3 30 D4 30 FC 30 20 00 5E FF 0F 00 05 20 00 ..0.0.0 .^.....
050 43 00 68 00 69 00 6B 00 61 00 00 00 20 00 57 00 C.h.i.k.a....W.
060 E5 52 83 73 81 5B 7E 31 20 20 20 10 00 03 90 5E .R.s.[~1     ....^
070 51 3D 51 3D 00 00 91 5E 51 3D BD 51 00 80 00 00 Q=Q=...^Q=.Q....
080 41 66 00 61 00 74 00 5F 00 31 00 0F 00 4B 2E 00 Af.a.t._.1...K..
090 30 00 2E 00 66 00 00 00 FF FF 00 00 FF FF FF FF 0...f.....
0A0 46 41 54 5F 31 30 7E 31 46 20 20 20 00 46 87 73 FAT_10~1F    .F.s
0B0 6D 3D 6D 3D 00 00 59 AB 44 3D 15 00 AF 05 00 00 m=m=...Y.D=.....
0C0 41 73 00 64 00 5F 00 66 00 75 00 0F 00 31 6E 00 As.d._.f.u...ln.
0D0 63 00 5F 00 31 00 2E 00 30 00 00 00 2E 00 66 00 c._.1...0....f.
0E0 53 44 5F 46 55 4E 7E 31 46 20 20 20 00 62 8B 73 SD_FUN~1F    .b.s
0F0 6D 3D 6D 3D 00 00 36 9B 44 3D 16 00 62 30 00 00 m=m=..6.D=..b0..
100 42 2E 00 66 00 00 00 FF FF FF FF 0F 00 0E FF FF B..f.....
110 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF .....
120 01 73 00 64 00 5F 00 76 00 69 00 0F 00 0E 65 00 .s.d._.v.i....e.
130 77 00 65 00 72 00 5F 00 31 00 00 00 2E 00 30 00 w.e.r._.1....0.
140 53 44 5F 56 49 45 7E 31 46 20 20 20 00 2C 8E 73 SD_VIE~1F    ,.s
150 6D 3D 6D 3D 00 00 8F A5 56 3D 17 00 5D 27 00 00 m=m=....V=...]'..
160 53 50 45 45 44 20 20 20 46 20 20 20 18 90 A0 73 SPEED   F     ...s
170 6D 3D 6D 3D 00 00 E0 4C 51 3D 18 00 AF 02 00 00 m=m=...LQ=.....
180 41 64 00 69 00 63 00 74 00 5F 00 0F 00 39 74 00 Ad.i.c.t._...9t.
190 65 00 73 00 74 00 2E 00 66 00 00 00 00 00 FF FF e.s.t...f.....
1A0 44 49 43 54 5F 54 7E 31 46 20 20 20 00 8B 07 74 DICT_T~1F    ...t
1B0 6D 3D 6D 3D 00 00 D7 73 6D 3D 19 00 2D 01 00 00 m=m=...sm=...-...
1C0 43 6C 00 65 00 72 00 5F 00 53 00 0F 00 D8 70 00 Cl.e.r._.S....p.
1D0 65 00 63 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.c...p.d...f...
2nd LFN 1E0 02 5F 00 48 00 6F 00 73 00 74 00 0F 00 D8 5F 00 ._..H.O.s.t...._.
1F0 43 00 6F 00 6E 00 74 00 72 00 00 00 6F 00 6C 00 C.o.n.t.r...o.l.

```

Input Address(hex) > 4ca00

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
1st LFN	000	01	53	00	69	00	6D	00	70	00	6C	00	0F	00	D8	69	00	.S.i.m.p.l....i.
	010	66	00	69	00	65	00	64	00	5F	00	00	00	53	00	44	00	f.i.e.d._...S.D.
SFN	020	53	49	4D	50	4C	49	7E	31	50	44	46	20	00	6C	C7	8C	SIMPLI~1PDF .l..
	030	70	3D	70	3D	00	00	F3	88	1D	3D	02	00	9B	0B	09	00	p=p=.....=.....
3rd(last) LFN	040	43	46	00	69	00	6C	00	65	00	4E	00	0F	00	E8	61	00	CF.i.l.e.N....a.
	050	6D	00	65	00	5F	00	74	00	65	00	00	00	73	00	74	00	m.e._.t.e...s.t.
2nd LFN	060	02	66	00	69	00	6C	00	65	00	5F	00	0F	00	E8	66	00	.f.i.l.e._....f.
	070	6F	00	72	00	5F	00	4C	00	6F	00	00	00	6E	00	67	00	o.r._.L.o...n.g.
1st LFN	080	01	54	00	68	00	69	00	73	00	5F	00	0F	00	E8	69	00	.T.h.i.s._....i.
	090	73	00	5F	00	74	00	65	00	78	00	00	00	74	00	2E	00	s._.t.e.x...t...
SFN	0A0	54	48	49	53	5F	49	7E	31	46	49	4C	20	00	80	9C	7C	THIS_I~1FIL ...
	0B0	73	3D	73	3D	00	00	F6	7E	73	3D	1A	00	04	00	00	00	s=s=...~s=.....
	0C0	41	73	00	69	00	7A	00	65	00	5F	00	0F	00	3F	7A	00	As.i.z.e._....?z.
	0D0	65	00	72	00	6F	00	2E	00	74	00	00	00	78	00	74	00	e.r.o...t...x.t.
	0E0	53	49	5A	45	5F	5A	7E	31	54	58	54	20	00	43	CF	B9	SIZE_Z~1TXT .C..
	0F0	73	3D	73	3D	00	00	B8	B9	73	3D	00	00	00	00	00	00	s=s=....s=.....
	100	4E	4F	54	48	49	4E	47	20	20	20	20	10	08	5E	43	48	NOTHING ..^CH
	110	74	3D	74	3D	00	00	2C	48	74	3D	1B	00	00	00	00	00	t=t=...,Ht=.....
	120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

	5 characters										2 characters				6 characters				
040	43	46	00	69	00	6C	00	65	00	4E	00	0F	00	E8	61	00	CF.i.l.e.N....a.		
050	6D	00	65	00	5F	00	74	00	65	00	00	00	73	00	74	00	m.e._.t.e...s.t.		
060	02	66	00	69	00	6C	00	65	00	5F	00	0F	00	E8	66	00	.f.i.l.e._....f.		
070	6F	00	72	00	5F	00	4C	00	6F	00	00	00	6E	00	67	00	o.r._.L.o...n.g.		
080	01	54	00	68	00	69	00	73	00	5F	00	0F	00	E8	69	00	.T.h.i.s._....i.		
090	73	00	5F	00	74	00	65	00	78	00	00	00	74	00	2E	00	s._.t.e.x...t...		
0A0	54	48	49	53	5F	49	7E	31	46	49	4C	20	00	80	9C	7C	THIS_I~1FIL ...		
0B0	73	3D	73	3D	00	00	F6	7E	73	3D	1A	00	04	00	00	00	s=s=...~s=.....		

LFN entry's top data is LFN's number. (Max 0x20)

Last LFN entry's top data is added 0x40. LFN use 2byte code.

Refer WORD" longname" and "file_detail".

sd_viewer don't display 2byte code.

WORD" longname" display about current sector and previous sector.

If LFN exceed 208-characters and SFN exist on top of sector, sd-viewer cause stack error.

I think no one make such a long file name.