

Bariř SAMANCI

Her Yönüyle FAT32



2010

Giriş

Bu kitapçık nereden çıktı? Herşey ARM serisi işlemciler ile SD kartlara veri yazıp okuma çalışmalarım ile başladı.Bu kartlara yazdığım verilerin sadece benim sistemimde değil bilgisayar, cep telefonu ve diğer sistemlerde okunabilir olmasını istiyordum.Bu yüzden standart bir dosyalama sistemi kullanmam gerekiyordu.Tercihim bir çok diskte kullanılan, standartlardan biri olan FAT32 oldu. FAT32 dosyalama sistemini öğrendiğimde işlemcinin SPI donanımını kullanarak SD kartlara veri yazıp okuyan kendi fread, fwrite gibi standart C fonksiyonlarımı yazabilecektim.Konuyu araştırmaya başladığımda Türkçe kaynak yoktu.Çalışmalarına temel olması amacıyla eriştiğim yabancı kaynakları geliştirerek uygulamalı ve görsel olarak bu dökümanı hazırladım. Daha sonra anadilimizdeki kaynak eksikliğini gidermek amacıyla yeniden düzenleyip yayınlamaya karar verdim.Bu döküman okunup anlaşıldığında FAT32 ile ilgili aklınızda soru işareti kalmayacak. İhtiyacı olanların işine yaraması dileğiyle...

Barış SAMANCI

Elektronik Mühendisi

www.barissamanci.net

baris@barissamanci.net

İçindekiler

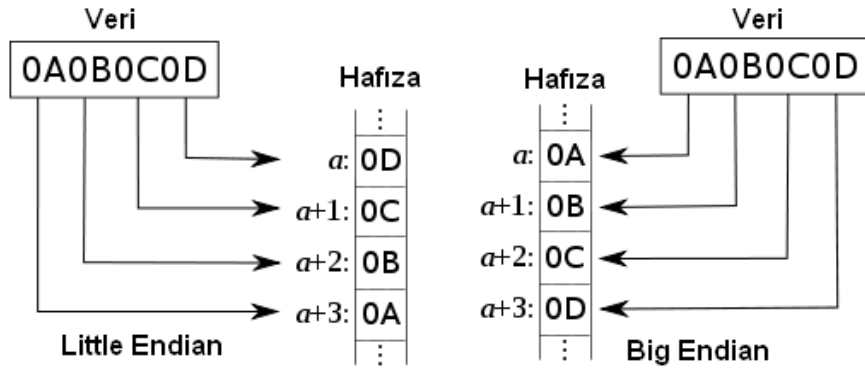
	Sayfa
1.Başlarken	4
2.Genel Bakış	5
3.Boot Sector	6
4.FS Information Sector	8
5.Küçük Dosya ve Dizinlerin Yeri	9
6.Kısa İsmle Sahip Dosyaların Dizin Girdisi	12
7.Uzun İsmle Sahip Dosyaların Dizin Girdisi	16
8.FAT ve Büyük Dosyalar	18
9.Master Boot Record	21
10.Sonuç	22

1.Başlarken

Başlamadan önce bilinmesi gereken bazı şeyler vardır.Bunlar anlatılan büyüklük birimlerinin neler olduğu ve anlatımda kullanılan programlardır.Diskleri incelemek için winhex programı kullanılmıştır.Hexadecimal sayıların hesaplanması için windowsun hesap makinasının bilimsel modu kullanılabilir.

Veriler diskte en küçük birim olan byte lar halinde saklanmaktadır.Belli sayıda byte ın ardarda dizilmesi ile sektörler, sektörlerin belli sayıda ardarda dizilmesi ile de cluster denilen kümeler oluşmaktadır.Dosyalar cluster denilen bu kümelere yayılmıştır.Küçük dosyalar 1 cluster yer kaplarken büyük dosyalar birden fazla cluster a yayılmıştır.Örnek olarak ben bu dökümanı yazarken 8Gb lık flash disk kullandım.Benim diskimde bir sektör 512 byte tan oluşmaktadır.1 cluster ise 8 sektörden oluşmaktadır.Yani benim diskimdeki 1 cluster 4Kb tır.Bu tür hesaplamalar ilerki konularda detaylı olarak incelenecektir.

Bilinmesi gereken diğer konu ise intel işlemcilerin windows ortamında verileri “**little endian**” olarak sakladığıdır.Bu şu anlama gelir.Bir byte tan büyük veriler diske yazılırken küçük adrese verinin düşük değerli byteleri, büyük adrese ise yüksek değerlikli bytelerinin yazıldığıdır.Aşağıdaki şekilde bu anlatılanlar gösterilmiştir.



Şekil 1. Little Endian , Big Endian

Şekil 1 de anlatılanlar gösterilmiştir.Winhex programını kullanırken verilerin yukarıdaki şekil de saklandığını göreceğiz.Hesaplama yapılırken buna dikkat edilmelidir. Aksi halde örnek olarak 0x200 yerine 0x2 okumuş oluruz.

2.Genel Bakış

FAT (File Allocation Table, Dosya Tahsis Tablosu) in amacı dosyaları disk üzerinde sistematik bir şekilde saklamak, yazmak veya okumak istenildiğine hızlı ve doğru bir şekilde ilgili adrese gidilip dosyanın okunmasını sağlamaktır.Bill Gates tarafından 1970 lerin sonuna doğru ilk DOS sürümlerinde kullanılmak üzere geliştirilmiştir.Daha sonra geliştirilerek FAT12, FAT16, ve bugünkü hali olan FAT32 ortaya çıkmıştır.Bu dosya tabloları küçük disklerde doğrudan disk üzerine veya büyük disklerde diskin bölünmüş partitionlarına (C, D gibi kısımlara) yerleşirler.Aşağıda dosya tahsis tablosunun partition yada disk üzerindeki genel yapısı gösterilmektedir.

İçerik	Boot Sektörü	FS Bilgi Sektörü (Sadece FAT32)	Rezerve Sektörler	Dosya Tahsis Tablosu #1	Dosya Tahsis Tablosu #2	Kök dizin (Sadece FAT12/FAT16)	Veri bölgesi (dizinler,dosyalar) (Disk sonuna kadar.)
Boyut(sektör)	(Rezerve sektör sayısı kadar)			(FAT sayısı) X (FAT başına sektör)		(kök dizin girdisi*32) / (Sektör başına byte)	(Cluster Sayısı) X (Cluster Başına Sektör sayısı)

Tablo 1. FAT genel yapısı

Diskler, diskin başında özel bir sektöre sahiptir.Bu sektörün adı Master Boot Record tur (MBR).Ve burada her partition un disk üzerinde nerede olduğu tutulmaktadır.Bu genellikle büyük kapasiteli HDD ler için geçerlidir.Bu uygulamada olduğu gibi ufak boyutlu diskler partitionlara bölünmezler.Biz anlatılarımızı flash disk için yapacağız fakat buralarda anlatılanlar partitionlar içinde geçerlidir.Bu konu 9. bölümde anlatılmıştır.

FAT dosya sistemi 4 kısımdan oluşur bunlar;

1. **Rezerve Sektörler**, en başa konumlanmışlardır.İlk rezerve sektör (sektör0) Boot Sektör diye adlandırılır.BIOS Parametre Bloğu diye bir kısım içermektedir.Bu kısım dosya sistemi bilgisi, diğer kısımların yerini gösteren adresler ve işletim sisteminin bootloder kodu gibi önemli bilgiler içermektedir.
2. **FAT bölgesi**, bu bölge genellikle iki kopya içermektedir (sisteme göre değişebilir).Kopya kısım birinci kısmın yedeğidir ve nadir kullanılır.Bu bölge bir nevi veri bölgesinin haritasıdır.Hangi clusterların dosyalar ve dizinler tarafından kullanıldığını gösterir.
3. **Kök Dizin bölgesi**, FAT12/16 sistemler için kök dizindeki dosyaların ve dizinlerin bilgilerini tutar.FAT32 sistemler için bu bilgiler veri bölgesinde tutulur.Bu yüzden FAT32 sistemlerde veri bölgesi bu adresten başlar.
4. **Veri bölgesi**, dosyaların ve dizinlerin saklandığı bölgedir.Dosyaların ve altdizinlerin boyutu boş cluster olduğu sürece istenildiği kadar arttırılabilir.Dosyalar cluster birimleri olarak saklanırlar.Bu yüzden 1Kb lık dosya, 32Kb lık clusterda saklandığında 31Kb kullanılamaz hale gelir.Diğer dosyalar için bu 31Kb lık alan kullanılamaz.FAT32 genellikle kök dizin tablosunu 2. clusterından başlatır.

3.Boot Sector

Sürücülere (partition) ayrılmış diskler(HDD)in ilk sektörü Master Boot Record (MBR) olarak, diğer disklerin (Disket) ki ise Volume Boot Record (VBR) olarak adlandırılır.Genel yapı olarak ilk 36 byte tüm FAT versiyonlarında aynıdır.Tablo 2 de bu yapı gösterilmiştir.

Byte Offset	Uzunluk (Byte)	Açıklama
0x00	3	Boot koduna jump direktifi.Bu alan 2 formda olabilir.Bunlar [0xEB, 0x??, 0x90] ve [0xE9, 0x??, 0x??] dir. 0x?? Herhangi bir değer anlamına gelmektedir.Bu alan işlemcilerin işletim sisteminin bootstrap koduna dallanabilmesi için kullanılır.
0x03	8	OEM adı.Burdaki değer diskin hangi sistemde formatlandığını gösterir.Örneğin, IBM 3.3, MSDOS5.0, MSWIN4.1, mkdosfs.
0x0B	2	Sektör başına düşen byte sayısı.Genel olarak kullanılan 512 dir. (BIOS Parameter Block buradan başlar)
0x0D	1	Cluster başına düşen sektör sayısı. 1 ile 128 arasındadır. İkinin kuvvetleri ile hesaplanır. Ancak Clusterdaki byte sayısı 32kb dan küçük olacak şekilde seçilmelidir. Bu byte sayısı (Sektör başına düşen byte sayısı) X (Cluster başına düşen sektör sayısı) bağıntısı ile hesaplanır.
0x0E	2	Rezerve sektör sayısı.Bu kadar sektörden sonra FAT bölgesi başlar.FAT32 HDD ler için bu değer genellikle 32 dir fakat disk boyutuna göre değişebilir.
0x10	1	FAT (dosya tahsis tablosu) sayısı.Bu alan genellikle 2 dir.
0x11	2	Maksimum kök dizin giriş sayısı.Bu değer FAT12/16 sistemlerde kullanılır.FAT32 için 0 olmalıdır.
0x13	2	Toplam sektör sayısı. (0 ise 0x20 den sonraki ilk 4 byte kullanılıyordur)
0x15	1	Medya tanımlayıcısı.Birçok değer alabilir. HDD ve benzeri (Flash diskler) için 0xF8 dir.
0x16	2	FAT12/16 için FAT tablosundaki sektör sayısı.
0x18	2	Track başına sektör sayısı
0x1A	2	Head sayısı
0x1C	4	Partition kullanan diskler için kullanılır.Partition dan önce FAT a kadar olan gizli sektör sayısını gösterir.
0x20	4	Bu sayı toplam sektör sayısını gösterir (65535 ten küçük ise 0x13 adresine bakılmalıdır)

Tablo 2. Boot Sector

Burdan sonrası FAT12/16 sistemler ve FAT32 sistemler için farklıdır.Genişletilmiş BIOS parametre bloğu olarak adlandırılır.Aşağıdaki tablo FAT12/16 için geçerlidir.

Byte Offset	Uzunluk (Byte)	Açıklama
0x24	1	Fiziksel sürücü numarası. HDD ler için 0x80, çıkarılabilir media için 0 dır.
0x25	1	Rezerve. WINNT de bit0 boot zamanında chkdsk isteği, bit1 ise yüzey taraması isteği oluşturur.
0x26	1	Boot imzası.0x29 olmalıdır.Bu takip eden 3 alanın var olduğunu belirtir.
0x27	4	ID (seri numarası)
0x2b	11	Birim etiketi.Boş byte lar 0x20 ile doldurulur.
0x36	8	FAT dosya sistemi tipi "FAT12 " veya "FAT16 ".Boşluklar 0x20 ile doldurulur.
0x3e	448	İşletim sistemi boot kodu
0x1FE	2	Boot sektör imzası (0x55 0xAA)

Tablo 3. FAT12/16 sistemler için genişletilmiş BIOS parametre bloğu

Aşağıdaki tablo FAT32 içindir.

Byte Offset	Uzunluk (Byte)	Açıklama
0x24	4	FAT (file allocation table) tablosu başına düşen sektör sayısı.
0x028	2	FAT bayrakları. FAT12/16 dan dönüştürme esnasında kullanılır.
0x2A	2	Versiyon (0 olarak tanımlanır)
0x2C	4	Kök dizinin başladığı cluster numarası
0x30	2	FS bilgilendirme sektörünün sektör numarası.
0x32	2	Bu boot sektörünün kopyasının sektör numarası (kopya yoksa 0 dır).
0x34	12	Rezerve
0x40	1	Fiziksel sürücü numarası. HDD ler için 0x80, çıkarılabilir media için 0 dır.
0x41	1	Rezerve. WINNT de bit0 boot zamanında chkdsk isteği bit1 ise yüzey taraması isteği oluşturur.
0x42	1	Boot imzası.0x29 olmalıdır.Bu takip eden 3 alanın var olduğunu belirtir.
0x43	4	ID (seri numarası)
0x47	11	Birim etiketi.Boş byte lar 0x20 ile doldurulur.
0x52	8	FAT dosya sistemi tipi "FAT32 ". Boşluklar 0x20 ile doldurulur.
0x5A	420	İşletim sistemi boot kodu
0x1FE	2	Boot sektör imzası (0x55 0xAA)

Tablo 4. FAT32 sistemler için genişletilmiş BIOS parametre bloğu

Yandaki şekilde diskin birim etiketi ve ID si gözükmektedir.

```
F:\>label
F: sürücüsündeki birim KINGSTON
Birim Seri Numarası FCA0-B2DD
Birim etiketi <11 karakter, boş bırakmak için ENTER>?
```

Şekil 2. Diski birim etiketi ve ID si

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	22	00	ËXMSDOS5.0...".
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	80	1F	00	00ø...?.ÿ.€...
00000020	80	80	EE	00	83	3B	00	00	00	00	00	00	02	00	00	00	€€i.f;.....
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	29	DD	B2	AO	FC	4E	4F	20	4E	41	4D	45	20	20	..)I² uNO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3É0N4ó
00000060	7B	8E	C1	8E	D9	BD	00	7C	88	4E	02	8A	56	40	B4	08	(0Á0Ù%. ^N.ŠV0 '.
00000070	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	Í.s.'ÿÿŠhf.9Æ0f.
00000080	B6	D1	80	E2	3F	F7	E2	86	CD	C0	ED	06	41	66	0F	B7	gÑEá?÷átÍÁi.Áf.·
00000090	C9	66	F7	E1	66	89	46	F8	83	7E	16	00	75	38	83	7E	Éf÷áf%Føf~..u8f~
000000A0	2A	00	77	32	66	8B	46	1C	66	83	C0	0C	BB	00	80	B9	*.w2f<F.fçÀ.».€²

Şekil 3. Boot Sector

Yukarıdaki şekilde boot sektörün başlangıcının disk üzerindeki görüntüsü gözükmektedir. Tablo 4 teki verilerden yararlanarak şekil 2 ve şekil 3 ü karşılaştırdığımızda verilerin doğru olduğunu görürüz. Birim etiketi verilmediği için "NO NAME " yazmaktadır.Boşlukların 0x20 ile doldurulduğuna dikkat edin.

4.FS Information Sector

FS bilgilendirme sektörü belirli işlemlere erişim zamanını arttırmak için geliştirilmiştir.(Boş hafızayı öğrenmek gibi).Fiziksel olarak boot sektördeki, 0x30 adresinin gösterdiği sektör numarasına konumlandırılmıştır.(Sektör numarası genellikle 1 dir.İşlemleri hızlandırmak için boot sektörden sonradır).

Byte Offset	Uzunluk (Byte)	Açıklama
0x00	4	FS bilgi sektörü imzası (0x52 0x52 0x61 0x41, "RRaA")
0x04	480	Rezerve (0 olarak okunur). Gelecekteki muhtemel eklentiler için.
0x1E4	4	FS bilgi sektörü imzası (0x72 0x72 0x41 0x61, "rrAa")
0x1E8	4	Disk üzerindeki kullanılmayan cluster sayısı. (Bilinmiyorsa -1)
0x1EC	4	En son tahsis edilen clusterın numarası
0xF0	14	Rezerve (0 olarak okunur)
0x1FE	2	FS bilgi sektörü imzası (0x55 0xAA)

Tablo 5. FS Information Sector

Tablo 4 ve şekil 3 yardımı ile FS sektörünün yerini kolayca tespit edebiliriz.Tablo 4 te FS sektörünün yerinin 0x30 ve 0x31. ofsetlerde olduğu gözükmektedir.Şekil 3 e baktığımızda bu FS sektörünün sektör numarasının 1 olduğunu görürüz.Şimdi winhexte 1. sektöre gidelim ve tablo 5 teki bilgileri bir kontrol edelim.Bu uygulamada 1 sektör 512 byte olduğundan FS sektörünün ofseti 0x200 olacaktır.Bu bilgiyi tablo 2 ve şekil 3 yardımı ile görebiliriz.

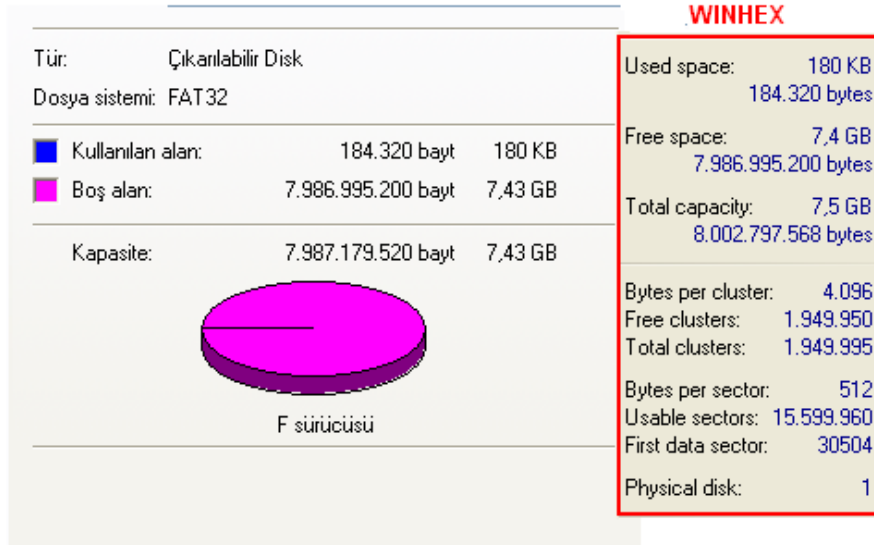
000000200	52 52 61 41 00 00 00 00 00 00 00 00 00 00 00 00	RRaA.....
000000210	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000230	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000240	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000250	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000260	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000270	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000290	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000310	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000320	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000330	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000340	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000350	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000360	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000370	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000410	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000430	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000440	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000450	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000460	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000470	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000480	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000490	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000500	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000510	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000520	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000530	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000540	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000550	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000560	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000570	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000580	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000590	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Şekil 4. FS bilgisi sektörü

Tablo 5 te anlatılanlar yukarıdaki şekil ile karşılaştırıldığında bilgilerin tutarlı oluğu gözükmektedir.FS sektörü 0x200 ofseti ile başlamaktadır.Şekil 4 te kırmızı dikdörtgen içerisinde 4 byte'lik bir veri alınmıştır.Bu alan tablo 5 ten karşılaştırıldığında bu değerlerin kullanılmayan boş cluster sayısı olduğunu öğreniriz.Bu değerler "little endian" yapısına göre okunduğunda 0x1DC0FE değerini verir.Buda ondalık sistemde 1.949.950 boş cluster var demektir.Şimdi 1 clusterın bu uygulamada 4Kb olduğunu tablo 2 nin 3. ve 4. alanları ile şekil 3 ü kullanarak hesaplayalım.Burda önemli bir bilgiyi hesaplamış olduk. 1.949.950 adet boş cluster varsa ve bir cluster 4Kb yer tutuyorsa bu iki sayıyı çarptığımızda diskimizdeki kullanılmayan boş alanı buluruz.

$$\text{Boş alan} = 1.949.950 * 4096 = 7.986.995.200 \text{ Byte}$$

Şimdi bu hesapladığımız değeri winhex ve windows ile karşılaştıralım.

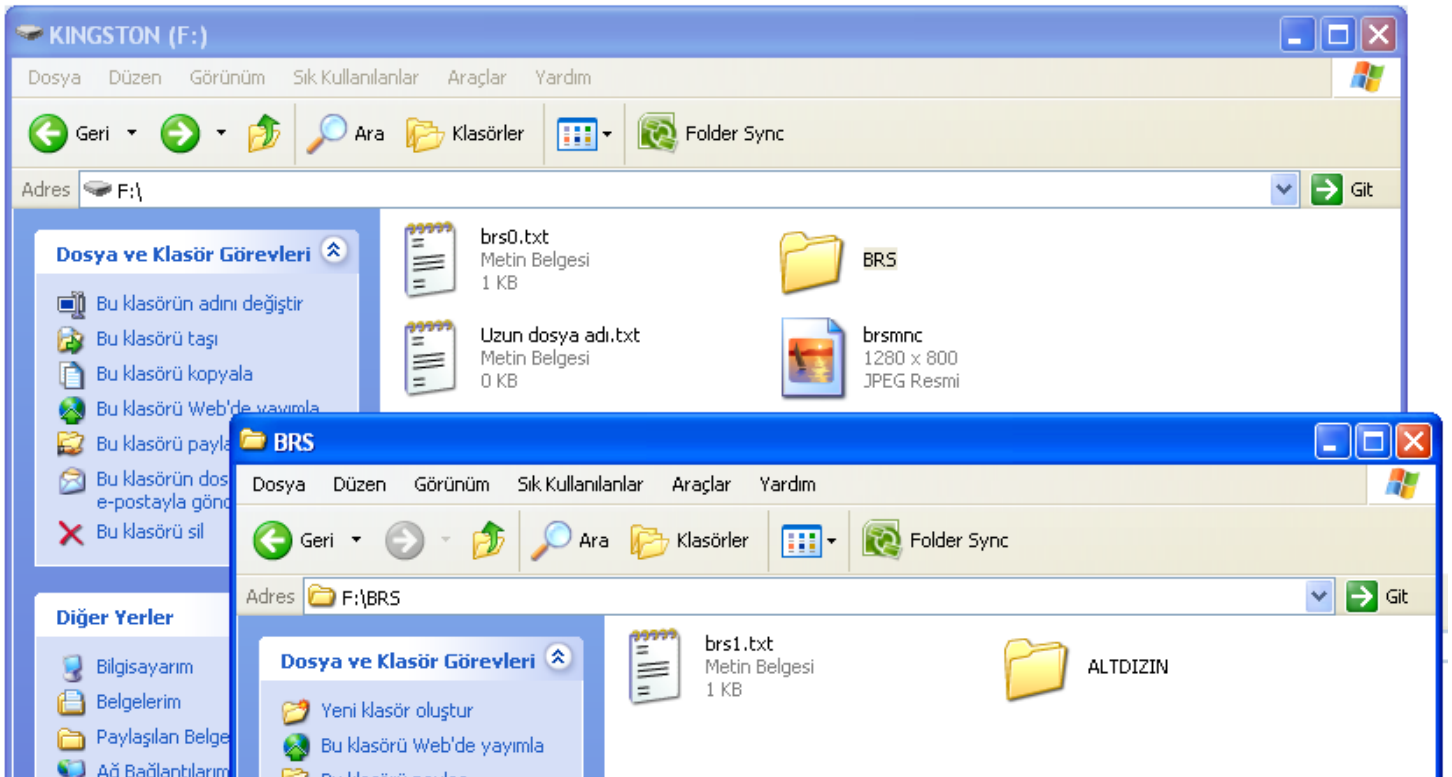


Şekil 5. Diskteki boş alanın hesaplanması

Görüldüğü gibi boş alan doğru olarak hesaplanmıştır.

5. Küçük Dosya ve Dizinlerin Yeri

Konuya başlamadan önce sürücüdeki dosyaların neler olduğuna bir göz atalım.



Şekil 6. Diskteki dosyalar

Dosyaları okumaya ilk başladığımızda elimizdeki tek veri kök dizinin nerede olduğudur. Kök dizinin cluster numarasını FAT32 için boot sektördeki 0x2C indisinden okuyabiliriz. Ayrıca kök dizinin FAT32 sistemler için veri bölgesinde olduğunu 2. kısımdaki tablo 1 den biliyoruz ve bu bölgenin sektör numarasını boot sektördeki veriler ile aşağıdaki bağıntıdan bulabiliriz. Veriler şekil 7 de işaretlidir.

$$\begin{aligned} & (\text{Rezerve sektör sayısı}) + (\text{FAT sayısı}) * (\text{FAT başına düşen sektör sayısı}) \\ & (\text{Ofset olarak}) [0x0E] + [0x10] * [0x24] \end{aligned}$$

Örnek olarak benim 8Gb lık FAT32 Kingston flash diskimin boot sektörünün winhexteki başlangıç görüntüsü aşağıdaki gibidir. Üstteki tabloların diğer bilgilerinde aşağıdaki görüntüden karşılaştırabilirsiniz.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	22	00	EXDOS5.0...
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	80	1F	00	00ø..?.ÿ.€...
00000020	80	80	EE	00	83	3B	00	00	00	00	00	00	02	00	00	00	€€i.f;.....
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	29	DD	B2	AD	FC	4E	4F	20	4E	41	4D	45	20	20	..) I* UNO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉÑPwó
00000060	7B	8E	C1	8E	D9	BD	00	7C	88	4E	02	8A	56	40	B4	08	{□Á□Ù%. ^N.ŠVØ'.
00000070	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	Í.š.'ÿÿŠřf.ŕÆØf.
00000080	B6	D1	80	E2	3F	F7	E2	86	CD	CO	ED	06	41	66	0F	B7	ŕŕŕéá?÷átÍÁí.Af..

Şekil 7. Boot Sector

$$\text{Root dizinin yer aldığı sektör} = 0x22 + 0x02 * 0x3B83 = 0x7728$$

Burada dikkat edilmesi gereken bilgilerin diskte "little endian" olarak saklandığıdır. Yani 0x24 adresindeki fat başına düşen sektör sayısı 0x833B00 değil 0x3B83 tür. Diğer dikkat edilmesi gereken ise hesaplamanın 16 lık sistemde yapıldığıdır. Şimdi üstteki hesapladığımız sektör numarasından kök dizinin byte olarak offsetini hesaplayalım. Bulduğumuz sektör indisini bir sektördeki byte sayısı ile çarparsak kök dizinin hangi byte tan başladığını hesaplayabiliriz. Sektör başına düşen byte sayısını boot sektördeki 0x0B offsetinden buluruz. Bu değer 2 bytelik bir bilgidir ve Şekil 7 de 0x200 olduğu gözükmektedir (0x002 değil!).

$$\text{Root dizinin byte olarak offseti} = 0x7728 * 0x200 = 0xEE5000$$

Şimdi bu ofseti kontrol edelim.

Cluster No.:	2	000EE5000	4B	49	4E	47	53	54	4F	4E	20	20	20	08	00	00	00	00	KINGSTON
	(Root directory)	000EE5010	00	00	00	00	00	00	82	90	5C	3C	00	00	00	00	00	00,□\<.....	
		000EE5020	85	6C	00	67	00	65	00	73	00	68	00	0E	00	83	2E	00	81	α = = i ~

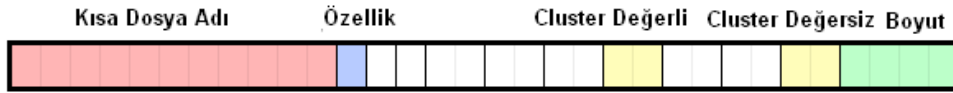
Şekil 8. Kök Dizinin disk üzerindeki yeri

Şekil 8 de hesapladığımız değer doğru olduğunu görüyoruz. Ayrıca kök dizinin cluster numarası 2 olarak verildiğini görüyoruz. Bu değeride şekil 7 deki 0x2C offsetinden kontrol edebiliriz.

Artık kök dizinin nerede olduğunu bildiğimize göre devam edebiliriz. Diziner bize içindeki dosyaların ve alt dizinlerin bir takım bilgilerini verirler. Bu bilgilerden en önemlisi içindeki dosya ve alt dizinlerin ilk cluster numaralarıdır diğerleri ise dosya boyutu, son değiştirilme zamanı, özelliği (Attribute) gibi bilgilerdir. Benim örnekte kullandığım disk için cluster boyutu 4KB tır. 4KB tan büyük olan dosyalar birden fazla clusterda zincir yapısı (bağlı liste) ile saklanırlar. Bu dosyaların diğer clusterlarına ise FAT tablosu sayesinde erişilir. Bu konu 8. bölümde anlatılmıştır. Şimdi dosyaların ve alt dizinlerin ilk cluster numaralarını bulalım. Bu sayede dosyanın yerini tespit edebiliriz.

Klasör girdileri 32 byte lık bloklar halinde tutulurlar. Bu sayede sektörlerden taşma olmaz ve sektördeki dosya/dizin bilgisi tam sayı olarak kalır. 512 byte lık sektörlerde 16 adet dosya/dizin verisi saklanmaktadır. İncelemeye başlamadan önce şunu bilelim. Saklanan dosya isimleri kısa (8.3) ve uzun olmak üzere 2 şekilde saklanırlar. Kısa isme sahip olan dosya ve dizin adlarını saklamak çok basittir. Uzun dosya isimleri ve adı içinde ASCII den farklı karakter olan dosya isimleri biraz daha kompleks olarak saklanmaktadır. Biz ilk önce kısa (Short File Name) formatında saklanan dosya isimlerini inceleyeceğiz. Uzun isme sahip olan dosyaların saklanması 7. Bölümde anlatılmıştır

Kısa dosya isimleri 8.3 yapısındadır. Yani dosya adı 8 karakter, dosya uzantısı 3 karakter dir. Bu karakterler ASCII formatındadır. Aşağıda 32 byte lık bir dosya veya dizine ait dizin girdisi gösterilmektedir. İlk etapta bizim için önemli olan renkli olarak gösterilen offsetlerdeki değerlerdir. Şuanda dosya yerinin nerede olduğunu öğrenmeye çalıştığımız için bu konu hakkında bilgilendirme yapıldıktan sonra tüm bu veriler 6. bölümde incelenecektir.



Şekil 9. 32 byte lık kısa dosya adının dizin girdisi

Burada dosya adı alanı 11 byte lıktır. Arada nokta olmadan 8.3 dosya isimlerini saklamaktadır. Eğer dosya isimleri 8 den küçük ise kalan boşluklar 0x20 ile doldurulmaktadır. Burdan dosya adını öğrendikten sonra dosyanın nerede olduğunu gösteren cluster adres değerlerinden dosyanın nerede olduğu bulunur. Aşağıdaki görüntüyü inceleyelim.

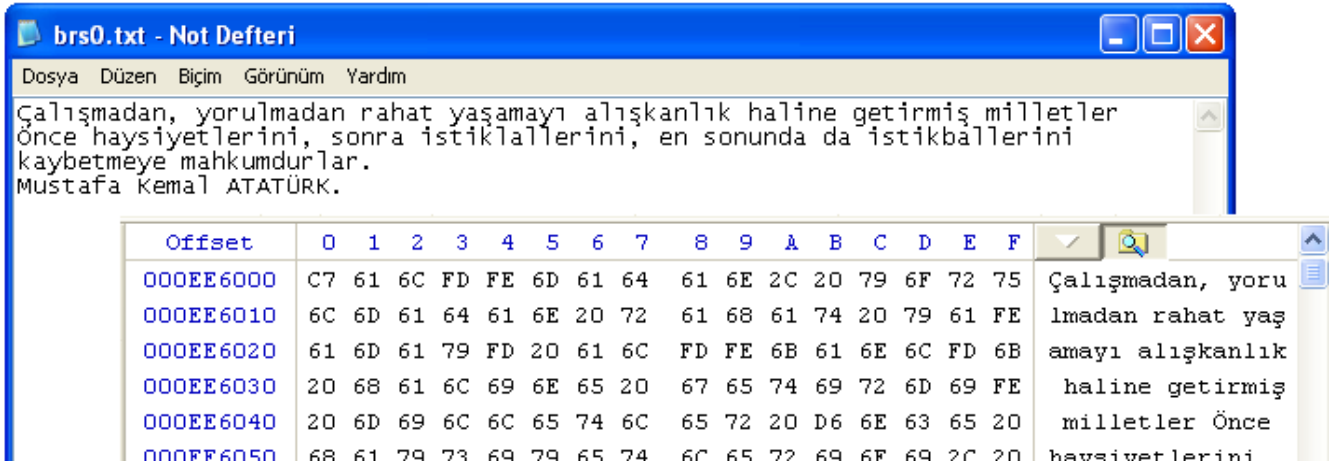
Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
(Root directory)		4,0 KB					30504
BRS		4,0 KB	28.02.2010 18:04:24	28.02.2010 18:04:26	28.02.2010		30520
brs0.txt	txt	195 bytes	28.02.2010 18:04:09	02.03.2010 17:13:42	02.03.2010	A	30512
brsmnc.jpg	jpg	163 KB	02.03.2010 00:54:59	31.12.2009 01:51:30	02.03.2010	A	15234856
Uzun dosya adi.txt	txt	0 bytes	02.03.2010 00:50:13	02.03.2010 00:50:14	02.03.2010	A	
Yeni Klasör		0 bytes	28.02.2010 18:04:24	28.02.2010 18:04:26	28.02.2010		30520

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000EE5080	42	52	53	30	20	20	20	20	54	58	54	20	18	8F	84	90
000EE5090	5C	3C	62	3C	00	00	B5	89	62	3C	03	00	C3	00	00	00
000EE50A0	E5	59	00	65	00	6E	00	69	00	20	00	0F	00	26	4B	00
000EE50B0	6C	00	61	00	73	00	F6	00	72	00	00	00	00	00	FF	FF
000EE50C0	E5	45	4E	49	4B	4C	7E	31	20	20	20	10	00	1F	8C	90
000EE50D0	5C	3C	5C	3C	00	00	8D	90	5C	3C	04	00	00	00	00	00

Şekil 10. Kök dizin ve dizin girdisinin incelenmesi

Burada örnek olarak "brs0.txt" dosyasının inceliyelim.32 bytelik veri offset alanında işaretlenmiştir. Burada ilk 11 byte dosya adıdır.(Boşlukların 0x20 ile doldurulduğuna dikkat edin).Sağ taraftaki text alanında dosya adı işaretlenmiştir.Burda şuna dikkat edebilirsiniz.Diskteki dosya adı küçük harfler ile yazılmıştır fakat büyük harfler ile saklanmaktadır.Bu yüzden Brs0.txt, BRS0.txt, brs0.txt gibi dosyalar aynı dizin içerisinde saklanamazlar.Şimdi bu dosyanın nerede saklandığını gösteren cluster değerlerini inceliyelim.Yüksek değerlikli cluster değeri bu 32 byte lık girdinin içerisinde 0x14. offsette, düşük değerlikli cluster 0x1A. offsette saklanmaktadır.Yukarıdaki örnekte gördüğümüz gibi brs0.txt dosyasının cluster değeri 0x0003 tür.Yani bu dosya 3. cluster dadır.

Daha önce kök dizinin 0x7728. sektörde olduğunu, 2. clusterda olduğunu ve byte olarak 0xEE5000 offsetinde olduğunu hesaplamıştık.Eğer kök dizinden bir cluster ileri gidersek "brs0.txt" dosyasına erişmiş oluruz.Bu uygulamadaki sektör boyutunun 512 byte, clusterdaki sektör sayısının ise 8 olduğunu söylemiştik.Yani 1 cluster 4Kb tır.O halde 0xEE5000 ofsetinden 4Kb ileri gidersek 0xEE6000 ofsetinde "brs0.txt" dosyasına ulaşmış oluruz.Hemen ilgili adrese bakalım.



Şekil 11. brs0.txt dosyasının diskteki konumu ve içeriği

Yukarıdaki şekilde görüldüğü gibi offset numarası doğru hesaplanmıştır ve dosya içeriği notepad ile karşılaştırıldığında doğru olarak gözükmektedir.Dosya bitimi 0x00 okunması ile anlaşılır.

6.Kısa İsime Sahip Dosyaların Dizin Girdisi

Şimdi şekil 9 da gösterilen 32 bytelik kısa dosya isimlerine sahip olan dosyaların dizin girdilerini biraz daha detaylı olarak inceliyelim.Bu 32 byte kendi içerisinde 12 alana ayrılmıştır ve bu alanlar bize ilgili dosya veya dizin hakkında dosya adı, dosya boyutu,özelliği, ilgili dosyanın disk üzerinde nerede olduğu, son yazma zamanı, son erişim tarihi gibi bilgiler hakkında bilgi verir. Bu bilgiler aşağıdaki tabloda listelenmiştir.

Byte Offset	Uzunluk (Byte)	Açıklama
0x00	11	Kısa dosya adı.
0x0B		Dosya özelliği.
		Açıklama
		BIT
		Sadece okunabilir
		Gizli.Dizin listelenirken gözükmeyen
		Sistem dosyası
		Birim etiketi.
1		Alt dizin
		Arşiv dosyası
		Kullanılmıyor. 0
		Kullanılmıyor. 0
0x0C	1	WIN NT sistemlerde kullanılmak üzere rezerve. Bu byte 0 ise dosya sadece yaratılmış daha önce hiç değiştirilmemiştir.
0x0D	1	Dosyanın yaratılış zamanının, 10 milisaniye hassasiyetindeki milisaniye değeri. Zaman alanlarında saniye, iki saniyelik hassasiyet ile saklandığından bu değer 0-199 arası değer almaktadır. 99 değerinden sonra oluşturma zamanı için saniye 1 artar.Son erişim zamanındaki saniyeler daima çifttir.
0x0E	2	Oluşturulma zamanı
0x10	2	Oluşturulma tarihi
0x12	2	Son okuma tarihi
0x14	2	Dosyanın ilk clusterının yüksek değerlikli kısmı. (FAT12/16 için hep 0)
0x16	2	Son değiştirme zamanı.(Dosyanın oluşturulmasında değiştirme olarak kabul edilir)
0x18	2	Son değiştirme tarihi.(Dosyanın oluşturulmasında değiştirme olarak kabul edilir)
0x1A	2	Dosyanın ilk clusterının düşük değerlikli kısmı.
0x1C	4	Dosyanın byte cinsinden boyutu.

*Uzun dosya ismi formatı için ilk 4 bit set edilir (0x01 | 0x02 | 0x04 | 0x08)

Tablo 6. 32 byte lık dizin girdisinin içeriği

Tablo 6 da kısa isme sahip olan dosya veya klasörlerin dizin girdisi detaylı bir şekilde verilmiştir.Burda dikkat edilmesi gereken iki şey vardır.Özellik alanındaki ilk 4 bit uzun dosya ismi formatında set edilmeli yani 0x0F değerini almalıdır.Diğeri ise dosya adının ilk byte ının ifade ettiği özel anlamlardır.Bunlar aşağıdaki tabloda anlatılmıştır.

Dosya Adının İlk Karakteri	Açıklama
0xE5	Bu dizin girdisinde daha önceden silinmiş bir dosyanın artıkları vardır.Bu alanın boş olduğunu belirtmek, üzerine yeni dosya yazılabilir olduğunu anlatmak için dosya adının ilk karakteri 0xE5 yazılır.Şekil 10 da silinmiş Yeni Klasör için 0xE5 değerini görebilirsiniz.
0x00	0 değeri okunduktan sonra klasörün sonuna gelinmiştir.Bu değerden ileride herhangi bir girdi yoktur.
0x05	Dizin girdisinin boş olduğunu anlatmak için kullanılan 0xE5 karakterinin yerine geçecek 0x05 değeridir.0xE5 Japon karakteridir.Bu karakterin kullanılabilmesi için 0x05 değeri yazılmalıdır.

Tablo 7. Kısa dosya isminin ilk karakterinin özel anlamları

Ayrıca dosya ismi için geçerli olmayan bir dizi değer vardır. Bunlar 0x05 hariç 0x20den küçük olan değerler, diğerleri ise 0x22, 0x2A, 0x2B, 0x2C, 0x2E, 0x2F, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E, 0x3F, 0x5B, 0x5C, 0x5D, ve 0x7C dir. Bunlar " * + , . / : ; < = > ? [\] | dir.

Şimdi tablo 6 da anlatılanları şekil 10 ile kıyaslayarak karşılaştıralım. "brs0.txt" dosyası için dosya adını ve cluster değerini 5. bölümde anlatmıştık. Şekil 10 da dosya boyutu son 4 byte ta 0xC3 olarak görülmektedir. Desimal olarak 195 değerine karşılık düşmektedir. Dosya görünümünde Size sekmesi altında bu değer 195 byte olarak gözükmektedir. 0x0B ofsetindeki değer 0x20 olduğu görülmektedir. Tablo 6 dan karşılaştırdığımızda bu alanın özellik alanı olduğunu ve 0x20 nin arşiv dosyası anlamına geldiğini görürüz. Zaten şekil 10 da A diye işaretlenmiştir. Zaman ve tarih ise biraz farklı saklanmaktadır. Tarih bilgisi 2 byte lık alana şu şekilde yayılmıştır.

- Bit 0-4: Gün 1-31 arası
- Bit 5-8: Ay 1-12 arası, 1=Ocak
- Bit 9-15: 1980 den bu yana geçen yıl sayısı. 0-127 (1980-2107)

Örnek olarak dosyanın oluşturulma tarihi şekil 10 da 0x10 ofsetinde 0x3C5C olarak gözükmektedir. [4:0] bitleri "11100" dır. Bu değer 28. gün anlamına gelmektedir. [8:5] bitleri "0010" dır buda ikinci ay anlamına gelmektedir. [15:9] bitleri ise "11110" dır yani 1980 yılından beri 30 yıl geçmiştir. O halde yıl 2010 dur. Bu değerleri birleştirdiğimizde dosyanın 28.02.2010 tarihinde oluşturulduğunu öğreniriz. Şekil 10 da da bu gözükmektedir. Zaman bilgisi ise 2 byte ta aşağıdaki formatta saklanmaktadır.

- Bit 0-4: İki saniye hassasiyetindeki saniye değeri. 0-29 (0-58 saniye)
- Bit 5-10: Dakika 0-59 arası
- Bit 11-15: Saat 0-23 arası

Örnek olarak dosyanın oluşturulma zamanı şekil 10 da 0x0E ofsetinde 0x9084 olarak gözükmektedir. [4:0] bitleri "00100" dır. Bu değer 4 tür. Saniyenin iki saniyelik hassasiyette tutulduğunu tablo 7 den hatırlarsak bu 4 değeri 8 saniye anlamına gelir. Burda hesaba katılması gereken bir nokta daha vardır ve o da milisaniye değeridir. 0x0D ofsetinde saklanmaktadır. Şekil 10 da bu değer 0x8F olarak gözükmektedir. Desimal olarak 143 değerine karşılık gelmektedir. 10 milisaniyelik hassasiyet ile saklandığından bu değeri 10 ile çarpmamız gerekmektedir. Buradan 1430 milisaniye değerini elde ederiz. 1000 milisaniye 1 saniye yaptığı için bu 1 saniyeyi daha önce bulduğumuz 8 in üzerine ilave edip saniye değerini 9 olarak buluruz. Dolayısıyla da milisaniye değeri 430 kalır. Dakikayı ise [10:5] bitleri arasından "00100" olarak buluruz yani dakika 4 tür. Saat değerini [15:11] bitleri arasından "10010" olarak buluruz buda 18 saat demektir. Şimdi oluşturulma zamanını toplarsak 18:04:09:0430 u buluruz. Bu değeri şekil 10 dan karşılaştırdığımızda tutarlı olduğunu görürüz.

Burda milisaniye değeri dosya oluştururken kayıt olmaktadır. Son değiştirme zamanı için milisaniye değeri tutulmaz. Buda son değiştirme zamanlarının saniyelerinin hep çift olacağı anlamına gelmektedir. Yaratma zamanı ise milisaniye saklayıcısından dolayı tek saniyelere sahip olabilir. Ayrıca son okuma için sadece tarih tutulmaktadır.

Bu bölümde anlatacağım son konu ise alt dizinlere ait ilk iki girdinin özel durumudur. Bir dizin altında yeni bir klasör oluşturulduğunda, bu klasöre bir cluster tahsis edilir ve bu clusterın içerişi 0 değeri ile doldurulur. Yeni klasörü oluşturduğumuz dizinin dizin girdilerine, oluşturmuş olduğumuz klasörün bilgileri eklenir. Bu bilgiler dosya adı kısmına istediğimiz klasör adı (kısaca isim formatına göre 8 karakter), cluster numarası, dosya özelliği kısmına bu dosyanın alt dizin olduğunu belirten 0x10, tarih ve zaman bilgileri ile boyut bilgisi için 0 dır. Daha sonra oluşturduğumuz dizinin ilk girdisi üst dizinde kendisi ile ilgili olarak girdiğimiz verilerdir fakat tek farkı vardır oda dosya adıdır. İlk girdi için dosya adı “. “. olmalıdır. İkinci girdi için ise iki fark vardır. Bunlardan birincisi dosya adı “. “. olmalıdır ikincisi ise cluster alanında, üst dizinin cluster numarasının yazması gerektiğidir. Bu cluster değeri üst dizin kök dizin olduğunda 0 değerini alır. Örneğin kök dizindeki BRS klasörü için kök dizindeki görüntü şekil 12 de arkaplanda gözükmektedir. “BRS” klasörünün girdileri ise şekil 12 de ön planda gözükmektedir. Görüldüğü gibi “BRS” klasörünün ilk iki girdisi yukarıda anlatıldığı gibidir. İlk girdinin cluster değeri, “BRS” klasörünün kendi cluster değeridir. İkinci girdinin cluster değeri ise üst dizin kök dizin olduğundan 0 dır. Diskteki dosyalar şekil 6 da gösterilmektedir.

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
(Root directory)		4,0 KB					30504
BRS		4,0 KB	28.02.2010 18:04:24	28.02.2010 18:04:26	03.03.2010		30520

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000EE50E0	42	52	53	20	20	20	20	20	20	20	10	00	1F	8C	90		BRS
000EE50F0	5C	3C	63	3C	00	00	8D	90	5C	3C	04	00	00	00	00	00	\<c<...<
000EE5100	E5	74	00	30	00	33	00	2E	00	72	00	0F	00	5B	61	00	ât.0.3...r...fa.

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
..		4,0 KB	03.03.2010 02:43:42	03.03.2010 02:43:44	03.03.2010		30568
ALTDIZIN		4,0 KB	03.03.2010 02:43:42	03.03.2010 02:43:44	03.03.2010		30568

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000EE7000	2E	20	20	20	20	20	20	20	20	20	10	00	1F	8C	90		.
000EE7010	5C	3C	5C	3C	00	00	8D	90	5C	3C	04	00	00	00	00	00	\<\<...<
000EE7020	2E	2E	20	20	20	20	20	20	20	20	10	00	1F	8C	90		..
000EE7030	5C	3C	5C	3C	00	00	8D	90	5C	3C	00	00	00	00	00	00	\<\<...<
000EE7040	42	52	53	31	20	20	20	20	54	58	54	20	18	27	8F	90	BRS1
000EE7050	5C	3C	63	3C	00	00	96	90	5C	3C	05	00	07	00	00	00	\<c<...-0<
000EE7060	FF	59	00	65	00	6F	00	69	00	20	00	0F	00	26	4B	00	ÿ e n i c v

Şekil 12. Kök dizin altındaki BRS dizininin incelenmesi

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
..		4,0 KB	03.03.2010 02:43:42	03.03.2010 02:43:44	03.03.2010		30568
ALTDIZIN		4,0 KB	03.03.2010 02:43:42	03.03.2010 02:43:44	03.03.2010		30568

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000EE70A0	41	4C	54	44	49	5A	49	4E	20	20	20	10	00	32	75	15	ALTDIZIN
000EE70B0	63	3C	63	3C	00	00	76	15	63	3C	0A	00	00	00	00	00	c<c<...v.c<
000EE70C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
..		4,0 KB	03.03.2010 02:43:42	03.03.2010 02:43:44	03.03.2010		30568
ALTDIZIN		4,0 KB	03.03.2010 02:43:42	03.03.2010 02:43:44	03.03.2010		30568

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000EE0000	2E	20	20	20	20	20	20	20	20	20	10	00	32	75	15		.
000EE0010	63	3C	63	3C	00	00	76	15	63	3C	0A	00	00	00	00	00	c<c<...v.c<
000EE0020	2E	2E	20	20	20	20	20	20	20	20	10	00	32	75	15		..
000EE0030	63	3C	63	3C	00	00	76	15	63	3C	0A	00	00	00	00	00	c<c<...v.c<
000EE0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Şekil 13. BRS dizini altındaki ALTDIZIN dizininin incelenmesi

Şekil 13 te arkaplandaki “BRS” dizininin, “ALTDIZIN” e ait girdilerinden alt dizinin 0x0A. clusterda olduğunu görüyoruz.Yukarıda anlatılan bilgilere göre şekil 13 te ön plandaki “ALTDIZINE” ait görüntüde ilk girdinin adının “. ” ve cluster numarasının kendi cluster numarası olduğunu biliyoruz. Arkaplandaki görüntü ile karşılaştırdığımızda cluster numaraları tutuyor.”ALTDIZIN” e ait ikinci girdinin adı “. ” ve cluster numarasının, üst dizin olan “BRS” dizininin cluster numarası olan 04 olduğunu görüyoruz.

NOT: Herhangi bir bilgi olmamasına rağmen dizin oluştururken 8 karakterden uzun isme sahip dizinlerin,8 karakterden kısa ama adı içerisinde ASCII olmayan karakterlere sahip dizinlerin,8 karakterden kısa olmasına rağmen içerisinde küçük harf barındıran dizinlerin uzun dosya adı formatında saklandığını gördüm.

7.Uzun İsme Sahip Dosyaların Dizin Girdisi

İhtiyaçlar artıp bilgisayarlar çeşitli dilleri kullanan ülkeler tarafından kullanılmaya başladığında sekiz karakterlik dosya adı ve ASCII karakter seti yetmez oldu.Yeni bir geliştirme gerekiyordu.En önemliside bu geliştirmenin var olan sistemleri değiştirmeden eski sistemler ile birlikte çalışabilir olması gerekiyordu.Geliştiriciler bu sorunu küçük bir hileyle çözdüler.Normal girdilerden önce dosya adının uzunluğuna göre ek sayıda girdi ekleniyordu.Eklenen bu ek girdilerin özellik alanına ise tablo 6 da anlatıldığı gibi 0x0F yazılarak ilk 4 bit set ediliyordu.Bu sayede eski sistemler için bu ek girdiler sistemin çalışmasını bozmadan yeni sistemler için uzun dosya isimlerine bir çözüm oluşturmuştur. Karakter seti diğer dillerde göz önünde bulundurularak ASCII den UTF16 ya geçmiştir.Yani ek girdilerde saklanan dosya isimlerinin karakterleri artık bir byte değil iki byte kaplayacaktır.Şimdi bu girdilerin hangi formatta saklandığına bir göz atıp konuyu bunun üzerinden anlatalım.

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
Uzun dosya adı.txt	txt	13 bytes	02.03.2010 00:50:13	03.03.2010 12:38:10	03.03.2010	A	30584

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
2. Ek Girdi →	42	31	01	2E	00	74	00	78	00	74	00	0F	00	CE	00	00	B1...t.x.t...İ..
1. Ek Girdi →	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	yyyyyyyyyy...yy
Normal Girdi →	01	55	00	7A	00	75	00	6E	00	20	00	0F	00	CE	64	00	.U.z.u.n. ...İd.
	6F	00	73	00	79	00	61	00	20	00	00	00	61	00	64	00	o.s.y.a. ...a.d.
	55	5A	55	4E	44	4F	7E	31	54	58	54	20	00	AB	46	06	UZUNDO~1TXT .«F.
	62	3C	63	3C	00	00	C5	64	63	3C	0C	00	0D	00	00	00	b<c<..Ådc<.....
	42	52	53	4D	4F	43	20	20	41	50	47	20	08	04	0D	06	RRSMNC .JPG İT

Şekil 14. “Uzun dosya adı.txt” dosyasının disk üzerindeki girdilerinin görüntüsü.

2. ve son Ek Girdi →	42h	ı	.	t	x	t	0Fh	00h	chk-sum	0000h
	FFFFh	FFFFh	FFFFh	FFFFh	FFFFh	FFFFh	0000h	FFFFh	FFFFh	
1. Ek Girdi →	01h	U	z	u	n		0Fh	00h	chk-sum	d
		o	s	y	a		0000h		a	d
Normal Girdi →	U	Z	U	N	D	O	~	1	T	X
	Oluşturma Tarihi	Son Erişim Tarihi	Üst Cluster	Son Dğşme Zamanı	Son Dğşme Tarihi	Alt Cluster	20h	NT	Rsvd	Oluşturma Zamanı
	Dosya Boyutu									

Şekil 15. "Uzun dosya adı.txt" dosyasının girdileri.

Yukarıdaki şekillerden görüldüğü gibi "Uzun dosya adı.txt" dosyası 3 girdide saklanmıştır. Eğer dosya adı daha uzun olsaydı girdi sayısında artacaktı. Bu girdilerden birtanesi, kısa dosya adlarını saklarken kullandığımız normal girdidir ve bu girdi en sonra yer alır. Bu girdiden geriye doğru uzun dosya adı girdileri yer alır. Dikkat edilirse dosya adı, kısa isim alanına sığacak şekilde "UZUNDO~1TXT" olarak değiştirilmiştir. Bu dosyanın DOS adıdır. Uzun isim girdilerin yapısı aşağıdaki tabloda verilmiştir.

Byte Offset	Uzunluk	Açıklama
0x00	1	Girdi dizisi numarası. Son girdi ise 0x40 ile ORlanır.
0x01	10	5 adet UTF16 karakter
0x0B	1	Özellik. Uzun dosya adı olduğu için her zaman 0x0F
0x0C	1	Rezerve. Her zaman 0
0x0D	1	DOS adının kontrol toplamı (Checksum)
0x0E	12	6 adet UTF16 karakter
0x1A	2	Alt cluster. Her zaman 0x0000
0x1C	4	2 adet UTF16 karakter

Tablo 8. Uzun dosya adının girdisi

Tablo 8 de verilenleri şekil 15 ve şekil 14 tekiler ile karşılaştıralım. Görüldüğü gibi karakterler iki byte lıktır. ASCII karakterler bir byte olduğu için UTF16 formatında değerli byte 0x00 ile doldurulmaktadır. Şekil 14 te görülmektedir. Girdiler her zaman girdinin numarası ile başlar. Eğer bu son girdi ise 0x40 değeri ile OR işlemine tabi tutulur. Şekil 15 ile karşılaştırdığımızda ilk girdinin "0x01" ikinci ve son girdinin ise "0x02 | 0x040" olduğunu görürüz. Ardından 10 byte, yani beş karakterlik ilk isim alanı gelir. Şekil 15 te görüldüğü gibi bizim örneğimizde ilk girdi için bu alanda "Uzun " karakterleri saklanmaktadır. Şekil 14 ten ASCII karakterler için üst byte ların 0x00 ile doldurulduğunu görebilirsiniz. Bu alanı özellik alanı takip eder. Uzun dosya isimlerinin girdileri için özellik alanı ilk dört bitin set edilmesiyle elde edilen 0x0F tir. Özellik alanını bir byte lık rezerve alanı ile ardından bir bytelik checksum kontrol alanı takip eder. Checksum alanındaki değer DOS adının bir takım işlemlerden geçirilmiş halidir. Bu alandaki değer ile DOS adını aynı işlemde geçirip elde ettiğimiz değeri karşılaştırdığımızda değerlerin aynı olması gerekmektedir. Bu, uzun isim girdilerin doğru yazıldığını ve normal girdinin devamı olduğunu gösterir. Checksum için yapılan işlem aşağıda gösterilmiştir.

```
unsigned char lfn_checksum(const unsigned char *DOSAdi)
{
    int i;
    unsigned char sum=0;

    for (i=11; i; i--)
        sum = ((sum & 1) << 7) + (sum >> 1) + *DOSAdi++;
    return sum;
}
```

Bu fonksiyona parametre olarak normal girdideki "UZUNDO~1TXT" dosya adını geçtiğimizde sonuç olarak 0xCE değerini alırız. Bu değeri şekil 14 ten karşılaştırdığımızda doğru olduğunu görürüz. Checksum alanından sonra 6 karakterlik ikinci isim alanı gelir. Bu alan bizim örneğimizde ilk girdi için "dosya " dır. Bu alandan sonra iki byte lık ve değeri 0x0000 olan alt cluster alanı gelir. Bu alandan sonrada iki karakterlik 3. isim alanı gelir ve ilk girdi biter. İkinci girdinin dizi numarası 2 dir fakat bizim örneğimiz için bu son girdi olduğu için 0x40 ile OR işlemine tabi tutulur ve 0x42 değeri elde edilir. Bunu şekil 15 ve şekil 14 ten kontrol edebilirsiniz. Dosya adının son karakterinden sonraki ilk karakter alanına dosya adının bittiğini söyleyen 0x0000 yazılır ve boşta kalan diğer karakter alanları 0xFFFF ile doldurulur. Yine şekil 15 ve şekil 14 yardımı ile bunu görebilirsiniz.

8. FAT ve Büyük Dosyalar

Bir clusterın bu uygulama için 4Kb olduğunu ve 4kb tan küçük dosyaların bir cluster da saklandığını önceki bölümlerde anlattık. Büyük dosyalar ise birden fazla clusterda saklanırlar. Dosyaların hangi clusterda saklandıklarını File Allocation Table (FAT) denen tablodan öğreniriz. Fakat bu tabloyu kullanabilmek için dosyaların ilk cluster larının yerini bilmemiz gerekmektedir. Bunları da 5. bölümde anlatıldığı gibi kök dizinden yola çıkarak dizinlerin dizin girdilerinden elde ederiz. Dizin girdilerinde geçerli girdilerin cluster alanından o dosyanın ilk cluster numarasını elde ederiz. Daha sonra bu numarayı offset olarak kullanıp FAT tablosundan ikinci clusterın yerini öğreniriz. Aynı işlemi tekrar edip üçüncü clusterın yerini öğreniriz. Bu işlem işlem end of cluster chain (EOC) işaretini görene kadar devam eder. FAT girdileri FAT32 için dört byte yani 32 bittir. Bu yüzden cluster değerlerini ofset olarak kullanırken girdileri dört byte olarak değerlendirmemiz gerekmektedir.

FAT32 için girdilerin 32 bit olduğunu söyledik ama aslında bu 32 bitin üst 4 biti kullanılmaz, ilk 28 bit kullanılır. Bu üst dört bit kullanılmamalıdır. Kullanılırsa bilgilerin yanlış yorumlanmasına sebebiyet verebilir. FAT tablosunu doğru yorumlayabilmek için EOC gibi anlamlara gelen özel değerler kullanılmaktadır. Bu değerler ile FAT tablosundan işlem yapacağımız clusterın durumunu öğrenebiliriz. Bu değerler aşağıdaki tabloda verilmiştir.

FAT12	FAT16	FAT32	Açıklama
0x000	0x0000	0x00000000	Boş Cluster
0x001	0x0001	0x00000001	Rezerve değer,kullanmayın
0x002–0xFFE	0x0002–0xFFEF	0x00000002– 0x0FFFFFFF	Kullanılmakta olan cluster. Değer bir sonraki clusteri işaret eder.
0xFF0–0xFF6	0xFFFF0–0xFFFF6	0xFFFFFFF0– 0xFFFFFFF6	Rezerve değer
0xFF7	0xFFFF7	0xFFFFFFF7	Clusterda bad sektör var veya rezerve
0xFF8–0xFFF	0xFFFF8–0xFFFFF	0xFFFFFFF8– 0xFFFFFFF	Dosyadaki son cluster. EOC

Tablo 9. FAT alanında kullanılan değerler

Şimdi görsel bir örnek ile tablo 9 daki verileri test edelim.Kök dizin altındaki 163Kb lık “brsmnc.jpg” dosyasının birden fazla clusterda nasıl saklandığına bakalım. 163Kb / 4Kb işleminden bu dosyayı saklamak için 41 cluster gerektiğini görürüz.Bölme işlemi kalan olduğu için yukarı yuvarlanmaktadır. Denemeler yaparken diske format atıp dosyaları tekrar oluşturduğum için zaman, boyut ve cluster numaraları daha önce verdiğim görüntüler ile tutmayabilir.

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
(Root directory)		4,0 KB					30504
BRS		4,0 KB	03.03.2010 20:38:24	03.03.2010 20:38:26	03.03.2010		30520
brs0.txt	txt	3 bytes	03.03.2010 20:38:05	03.03.2010 20:38:16	03.03.2010	A	30512
brsmnc.jpg	jpg	163 KB	03.03.2010 20:42:32	31.12.2009 01:51:30	03.03.2010	A	30536
Uzun dosya adı.txt	txt	0 bytes	03.03.2010 20:42:49	03.03.2010 20:42:50	03.03.2010	A	

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000EE5080	42	52	53	30	20	20	20	20	54	58	54	20	18	7B	C2	A4	BRS0
000EE5090	63	3C	63	3C	00	00	C8	A4	63	3C	03	00	03	00	00	00	TEXT
000EE50A0	65	59	00	65	00	6E	00	69	00	20	00	0F	00	26	4B	00	...
000EE50B0	6C	00	61	00	73	00	F6	00	72	00	00	00	00	FF	FF	00	...
000EE50C0	65	45	4E	49	4B	4C	7E	31	20	20	20	10	00	0A	CC	A4	...
000EE50D0	63	3C	63	3C	00	00	CD	A4	63	3C	04	00	00	00	00	00	...
000EE50E0	42	52	53	20	20	20	20	20	20	20	20	10	00	0A	CC	A4	BRS
000EE50F0	63	3C	63	3C	00	00	CD	A4	63	3C	04	00	00	00	00	00	...
000EE5100	42	52	53	4D	4E	43	20	20	4A	50	47	20	08	4F	50	A5	BRSMNC
000EE5110	63	3C	63	3C	00	00	6F	0E	9F	3B	06	00	68	8C	02	00	JPG
000EE5120	42	31	01	2E	00	74	00	78	00	74	00	0F	00	CE	00	00	...
000EE5130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	...
000EE5140	01	55	00	7A	00	75	00	6E	00	20	00	0F	00	CE	64	00	...
000EE5150	6F	00	73	00	79	00	61	00	20	00	00	00	61	00	64	00	...
000EE5160	55	5A	55	4E	44	4F	7E	31	54	58	54	20	00	68	58	A5	UZUNDO~1TXT
000EE5170	63	3C	63	3C	00	00	A1	A6	63	3C	30	00	03	00	00	00	...

Şekil 16. Kök dizin altındaki dosya ve dizinlerin girdileri ve ilk cluster numaraları

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
..							
ALTDIZIN		4.0 KB	03.03.2010 20:49:44	03.03.2010 20:49:46	03.03.2010		30864
brs1.txt	txt	4 bytes	03.03.2010 20:39:25	03.03.2010 20:39:34	03.03.2010	A	30528

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
000EE70A0	42	52	53	31	20	20	20	20	54	58	54	20	18	C6	EC	A4	BRS1	TXT	.Ei
000EE70B0	63	3C	63	3C	00	00	F1	A4	63	3C	05	00	04	00	00	00	e<c<..ħ	e<.....	
000EE71C0	41	4C	54	44	49	5A	49	4E	20	20	20	10	00	46	36	A6	ALTDIZIN	..F6}	
000EE71D0	63	3C	63	3C	00	00	37	A6	63	3C	2F	00	00	00	00	00	e<c<..7;	e</.....	

Şekil 17. BRS dizini altındaki dosya ve dizinlerin girdileri ve ilk cluster numaraları

Şekil 16 ve şekil 17 de tüm dizinlerin ve dosyaların girdileri verilmiştir ve cluster numaraları işaretlenmiştir.Şimdi FAT alanına gidip bu cluster değerlerini inceleyelim.FAT alanının, tablo 1 de rezerve sektörlerden sonra başladığını ve rezerve sektör sayısının boot sektörde 0x0E ofsetinde saklandığını tablo 2 den biliyoruz. Şekil 7 den bu alanı kontrol ettiğimizde 0x22 yani 34 olduğunu görürüz. Yani FAT tablosu desimal olarak 34. sektörden başlamaktadır.Bu sektöre bir göz atalım.

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
..							
ALTDIZIN		4.0 KB	03.03.2010 20:49:44	03.03.2010 20:49:46	03.03.2010		30864
brs1.txt	txt	4 bytes	03.03.2010 20:39:25	03.03.2010 20:39:34	03.03.2010	A	30528

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000004400	F8	F1	FF	0F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿ.ÿÿÿÿÿÿ.ÿÿÿ.
000004410	FF	FF	FF	0F	FF	FF	FF	0F	07	00	00	00	08	00	00	08	ÿÿÿ.ÿÿÿ.....
000004420	09	00	00	00	0A	00	00	00	0B	00	00	0C	00	00	00	00
000004430	0D	00	00	00	0E	00	00	00	00	00	10	00	00	00	00	00
000004440	11	00	00	00	12	00	00	00	00	00	14	00	00	00	00	00
000004450	15	00	00	00	16	00	00	00	00	00	18	00	00	00	00	00	...
000004460	19	00	00	00	1A	00	00	00	00	00	1C	00	00	00	00	00
000004470	1D	00	00	00	1E	00	00	00	00	00	20	00	00	00	00	00
000004480	21	00	00	00	22	00	00	00	00	00	24	00	00	00	00	00	!...".#...\$...
000004490	25	00	00	00	26	00	00	00	00	00	28	C	00	00	00	00	%...&...'...(...
0000044A0	29	00	00	00	2A	00	00	00	00	00	2C	00	00	00	00	00)...*...+.....
0000044B0	2D	00	00	00	2E	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF	-.....ÿÿÿ.ÿÿÿ.
0000044C0	FF	FF	FF	0F	00	00	00	00	00	00	00	00	00	00	00	00	ÿÿÿ.....
0000044D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Şekil 18. FAT – Dosya Tahsis Tablosu

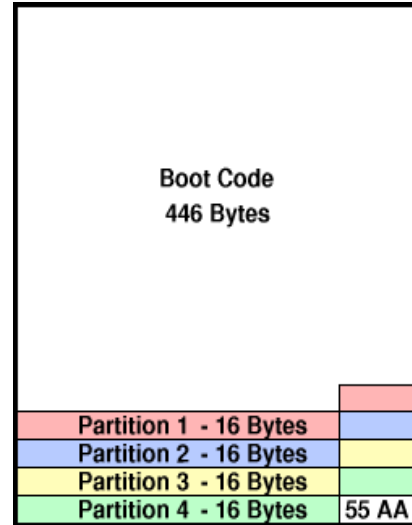
Şekil 18 de FAT tablosunun başladığı sektör gösterilmiştir.Burda girdiler dört byte ıktır. Yani her dördlüyü 1 olarak saymamız gerekmektedir.Ayrıca değerlerin little endian olarak tutulduğunu bir daha hatırlayalım.Şekil 17 ve şekil 16 da işaretlenen dosyalar şekil 18 de de işaretlenmiştir.Tablo 9 dan 0x0FFFFFFF değerinin dosyadaki son clusterı belirttiğini hatırlayarak devam edelim. Kök dizinin cluster numarasının iki olduğunu daha önceki konularda anlatmıştık. Şimdi şekil 18 de üçüncü dördlü ye yani ikinci indise bakalım.Burdaki değer 0x0FFFFFFF olduğunu görürüz.Bu kök dizindeki verilerin bir cluster a sığıdığı anlamına gelir.Şekil 16 da “brs0.txt” dosyasının cluster numarasının 3 olduğu gözükmekte. Şekil 18 de üçüncü indise, yani dördüncü dördlüye bakalım.Değer 0x0FFFFFFF tir. Yani

“brs0.txt” dosyasıda 4Kb tan küçük olduğu için bir clusterda saklanmıştır.Aynı şekilde “BRS” dizini ile “brs1.txt” dosyasıda bir clustera yerleşmiştir.Şimdi 41 cluster yer kaplayan “brsmnc.jpg” dosyasına bakalım.Şekil 16 da brsmnc.jpg dosyasının ilk cluster numarasının 6 olduğu gözükmektedir.Şekil 18 de 6. indise baktığımızda 0x00000007 değerini görürüz.Tablo 9 dan bu değeri karşılaştırdığımızda bu değer brsmnc.jpg dosyasının ikinci clusterını işaret ettiğini görürüz. O halde 7. indisteki, bizim dosyamızın ikinci clusterıdır ve değeri 0x00000008 dir.Aynı şekilde üçüncü clusterımızın 8. indiste olduğunu gösterir. Bu şekilde devam ederek 0x2E. indiste 0x0FFFFFFF değerini okuruz.Bu değer 0x2E. clusterın “brsmnc.jpg” dosyasının son clusterı (EOC) olduğunu ifade eder ve dosya burda bitmiştir. 0x2F clusterında “ALTDIZIN” in ve 0x30 clusterın da ise “Uzun dosya adı.txt” dosyasının saklandığını ve bunların bir clusterlık dosyalar olduğunu görürüz.

Burda dikkat edilmesi gereken bir nokta vardır.Bizim örneğimizde disk temiz olduğu için “brsmnc.jpg” dosyası clusterlara ardışıl olarak yerleşmiştir.Fakat kullanılmış bir diskte ardışıl olarak dizileceği garanti edilemez.Veriler cluster boyutlarında bölünerek boş bulunan clusterlara yerleştirilir ve numaraları şekil 18 de görüldüğü gibi FAT tablosuna yerleştirilir.

9.Master Boot Record

Master Boot Record disk üzerindeki ilk sektördür ve ilk partition oluşturulurken oluşturulur.Bu bölüm işletim sisteminin başlatılması için gereken küçük bir kod ve disk üzerindeki bölümlerin (partition) bilgilerini bulundurur.Bu sektördeki ilk 446 Byte içerisinde bootstrap code denilen boot kodu bulunur.Daha sonra bölümlere ait bilgilerin saklandığı kısımlar gelir.Bu kısımlar 16 byte tan meydana gelmektedir ve son iki byte 0x55-0XAA sektör imzasıdır.Şekil 19 da anlatılanlar görülmektedir.Aynı şekilde MBR nin maksimum dört bölüm bilgisi saklayabileceğini görebilirsiniz.Eğer dört bölümden fazla bölümümüz varsa genişletilmiş bölümlendirme (extended partitioning) adında bir teknik kullanılır.Dört bölüm olması şart değildir genelde iki bölümden fazla bölümü olan disklerde bu yöntem kullanılmaktadır. Genişletilmiş bölümlendirmede bölümlerin bilgileri normal bir şekilde MBR de yer alır fakat bölümlerin ilk sektörleri MBR de olduğu gibi diğer bölümlerin bilgilerini tutar.



Şekil19. Master Boot Record

Bölüm bilgilerinin 16 byte tan oluştuğunu söylemiştik.Şimdi bu 16 byte in bölüm bilgilerini ne şekilde sakladığına bakalım.Bu 16 byte altı alandan oluşmaktadır.Şekil 20 ve tablo 10 da bu alanlar gösterilmiştir.İlk alan bir byte lık boot bayrağıdır.Bu byte 0x80 ise bilgisayar bu bölümden açılmaktadır. Aksi halde 0x00 olmadır.Diğer değerler geçersizdir.Daha sonra 3 byte lık CHS (Cylinder Head Sector) başlangıç alanı gelmektedir.Diğer alanlar ve içerikleri tablo 10 ve şekil 20 de verilmiştir. Burada HDD içindeki diskler ve okuyucu kafalardan bahsedilmiştir.



Şekil 20. Bölüm bilgisi alanları

Offset	Uzunluk	Açıklama
0x00	1	0x80 yüklenebilir, 0x00 yüklenemez. Diğer değerler geçersizdir.
0x01	3	Bölümdeki ilk bloğun CHS adresi. İçeriği sonraki 3 satırda anlatılıyor.
0x01	1	Kafa (okuyucunun)
0x02	1	[5-0] arası bitler sektör bilgisi; [7-6] arası bitler silindir bilgisi
0x03	1	[7-0] arası bitler silindir bilgisi.
0x04	1	Bölüm tipi. 0x0B veya 0x0C FAT32 için kullanılır.
0x05	3	Bölümdeki son bloğun CHS adresi. İçeriği sonraki 3 satırda anlatılıyor.
0x05	1	Kafa (okuyucunun)
0x06	1	[5-0] arası bitler sektör bilgisi; [7-6] arası bitler silindir bilgisi
0x07	1	[7-0] arası bitler silindir bilgisi.
0x08	4	Bölümdeki ilk sektörün LBA (Lojik Blok Adresi) adresi
0x0C	4	Bölümdeki sektör sayısı

Tablo 10. Bölüm bilgisi alanları

Yukarıda anlatılanlardan önemli olanları aşağıda şekil 21 de işaretlenmiş olarak görebilirsiniz. Görüldüğü gibi bölümlerden bir tanesi 0x80 ile işaretlenerek yüklenebilir olduğu belirtilmiştir. Aynı şekilde 0x0C ile bölümün FAT32 biçimlendirildiği belirtilmiştir. LBA alanında bölümün sektör numarası 0x3F olarak verilmiştir. Bu sayıyı bir sektördeki byte sayısı (0x200) ile çarparsak 0x7E00 buluruz. Bu byte cinsinden ofsettir. 0x7E00 adresine gidip bakarsak şekil 3 teki tablo ile karşılaşırız. Aynı hesaplamaları yaparak bölümdeki FAT tablosu ve dosyalara erişip işlem yapabiliriz.

29	000000180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1	000000190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	0000001A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
iB	0000001B0	00 00 00 00 00 00 2C 43 62 D7 C0 D7 C0 00 00 80 01,Cb×À×À..€.
es	0000001C0	01 00 0C FE FF FF 3F 00 00 00 04 0F A9 04 00 00	...şÿÿ?.....©...
iB	0000001D0	C1 FF 0F FE FF FF 43 0F A9 04 C1 91 A8 04 00 00	Áÿ.şÿÿC.©.Á'...
es	0000001E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
iB	0000001F0	00 00 00 00 00 00 00 00 00 00 00 00 00 55 AAU²
es			

Şekil 21. MBR Bölüm bilgileri

10.Sonuç

Bu dökümanda FAT32 ile ilgili bilgiler verilmiştir. Bu bilgiler ile kendi kodlarınızı yazıp en alt düzeyde FAT32 biçimlendirilmiş disk ve taşınabilir medyalar ile haberleşebilirsiniz. Faydalı olması dileğiyle.