Volüm (Birim) Analizi

Dosya Sistem Analizi Hafta 4

Yrd. Doç. Dr. Erhan AKBAL

Amaç

- Disk Volüm analizi, bölümlere (partition) ayırma ve depolama aygıtlarındaki baytları bir araya getirme ile ilgili veri yapılarına inceleme konularını içerir.
- Disk birimleri dosya sistemini ve diğer yapısal veriyi depolamak için kullanılır.

Giriş - I

- Dijital depolama ortamları, verilerin verimli bir şekilde alınmasına izin vermek için düzenlenmiştir.
- Disk volüm sistemi ile Microsoft Windows'u yüklerken ve sabit disk bölümleri oluştururken sıklıkla karşılaşırız.
- Yükleme süreci, kullanıcıyı birincil ve mantıksal bölüm oluşturma süreci boyunca yönlendirir ve sonunda bilgisayar, veriyi depolayacağı "sürücüler" veya "birimler" listesine sahip olur.
- Bir UNIX işletim sistemi yüklerken benzer bir süreç oluşur ve büyük depolama ortamlarında, birden fazla diskin sanki bir tane büyük diskten oluşmuş gibi görünmesi için volume yönetimi yazılımları kullanılır.

Giriş - II

- Adli inceleme sırasında, bütün bir disk imajını almak ve imajı analiz araçlarına alarak incelemek genel bir yaklaşımdır.
- Birçok inceleme aracı disk imajını partitionlara otomatik olarak ayırır, ancak bazen de sorunlara olabilir.
- Bu nedenle bir aracın ne yaptığının ayrıntılarını ve bir disk bozulduğunda neden sorun yaşadığını anlamak önemlidir.
- Örneğin, disk partition yapısı silinmiş veya şüpheli tarafından değiştirilmiş olabilir veya araç yalnızca bir partitionu bulamayabilir.

Volüm Kavramları

- Volüm (Birim) sistemleri iki temel kavramda açıklanmaktadır.
- İlki bir saklama biriminde birden fazla farklı volüm oluşturmak,
- ikinci ise saklama alanını bağımsız partitionlara bölmektir.
- Partition ve volüm terimler sıklıkla birlikte kullanılır. Ancak aralarında belirli farklar bulunmaktadır.
- Volüm bir işletim sisteminin veya uygulamanın depolama için kullanabileceği adreslenebilir sektörlerin toplamıdır.
- Bir volümdeki sektörler bir fiziksel depolama aygıtında ardışık olmak zorunda değildir.
- Bunun yerine sektörlerdeki iz bilgilerini vermesi istenir. Sabit disk ardışık sektörlerde bulunan bir birime örnek olarak verilebilir. Bir volüm daha küçük volümlerin bir araya getirilmesi veya birleştirilmesi ilede ortaya çıkabilir.

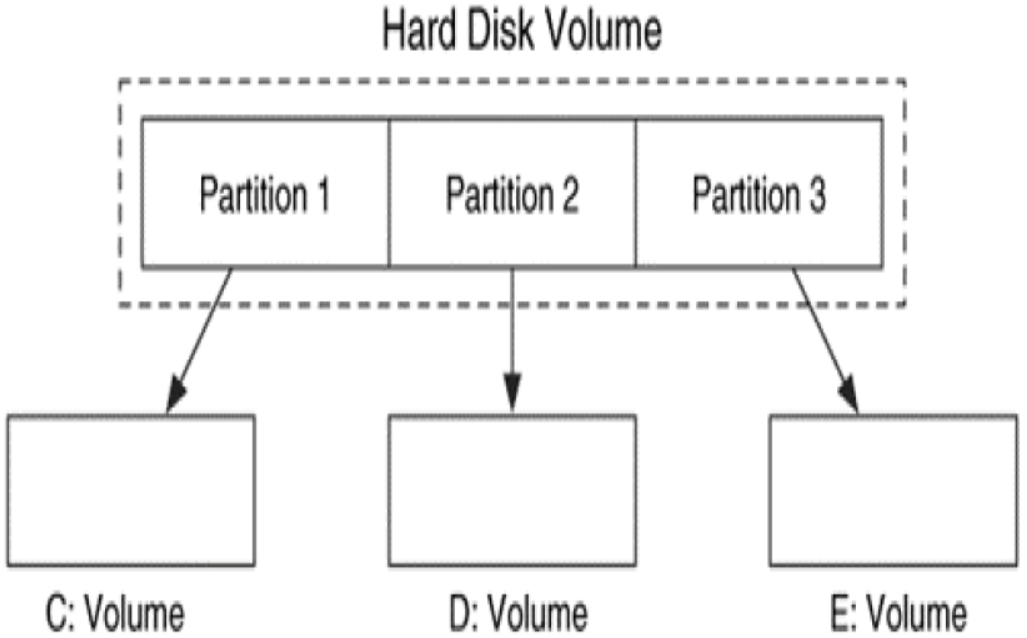
Partition (Bölüm) Kavramı

- Bir partition bir volümdeki (birimdeki) ardışık sektörlerin toplamıdır.
- Tanımı gereği bir partitionda bir volümdür. Bu nedenle bu terimler birbirine karıştırılmaktadır. Volüm partitionların üst birimi olarak düşünülebilir.
  - Bazı dosya sistemleri hard disk boyutundan daha küçük maksimum boyuta sahip olabilir.
  - Birçok diz üstü bilgisayar, uyku durumuna geçtiğinde bellek içeriğini saklamak için özel bir bölüm kullanır.
  - UNIX sistemler dosya sisteminden kaynaklı hataların minimuma indirmek için farklı dizinler için farklı partitionlar kullanır.
  - Windows, Linux gibi birçok işletim sistemine sahip IA32 tabanlı sistemler, her işletim sistemi için ayrı partitonlar gerektirebilir.

Örnek Yapı

- Bir sabit diskli Microsoft Windows sistemi düşünelim.
- Sabit disk volümü, daha küçük üç partition haline bölünmüştür ve her birinin bir dosya sistemi vardır.
- Windows, her birime C, D ve E adları atar.

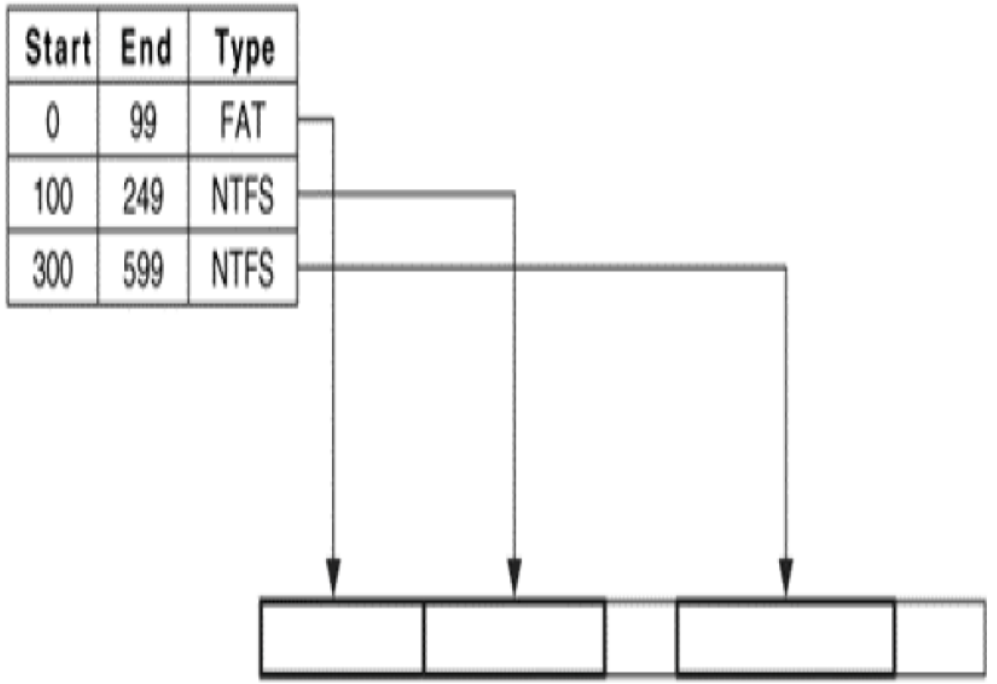
Figure 4.1. An example hard disk volume is organized into three partitions, which are assigned volume names.



Partition Türleri

- Her işletim sistemi ve donanım platformu, genellikle farklı bir partition yöntemi kullanır.
- PC Tabanlı Partitionlar ve Sunucu tabanlı Partitionlar da farklı uygulamaları kullanmaktadır, ancak belirli temel kısımları bulunmaktadır.
- Ortak partition sistemleri bir veya daha fazla tabloya sahiptir ve her tablo girişi bir partitionu açıklar.
- Girişteki veriler, partition başlangıç sektörüne, bitiş sektörüne (veya uzunluğa) ve türüne sahiptir.

Figure 4.2. A basic table with entries for the start, end, and type of each partition.



Partition Yapısı

- Bir partition sisteminin amacı, bir volümün düzenini belirlemektir.
- Bu nedenle, yalnızca gerekli veriler, her partition için başlangıç ve bitiş konumları arasındadır.
- Bir partition sistemi, bu değerler bozulmuş veya mevcut değilse amacına hizmet edemez.
- Açıklama gibi diğer tüm alanlar önemsizdir. Çoğu durumda, bir partitionun ilk ve son sektörü herhangi bir şey içermez.
- Partition sistemi yapıları eksik olduğunda, Partition sınırları bazen Partition içinde saklanan bilgiler kullanarak tahmin edilebilir. Bu, peyzaja bakılarak mülk sınırlarını tahmin etmeye benzer.
- Bir partition sisteminin, sabit diskteki arabirim türüne değil, işletim sistemine bağlı olduğunu unutmayın.
- Bu nedenle, bir ATA / IDE veya SCSI kullanıyorsa, bunun için Windows sistemi bir disk bölümleme sistemi kullanır.

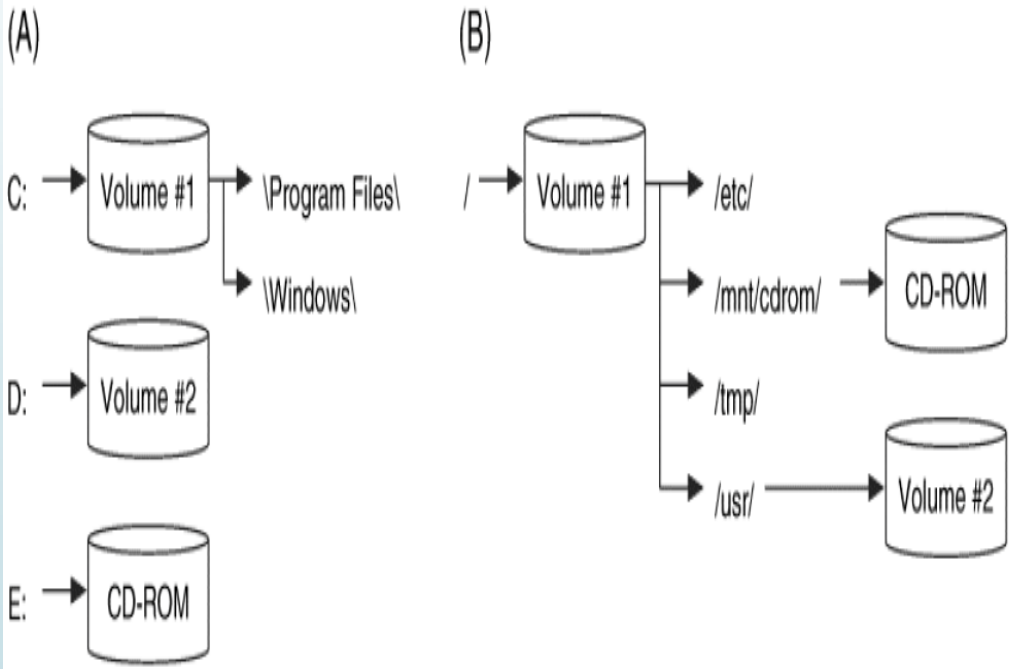
Unix de Volümlerin Kullanımı

- UNIX sistemleri genellikle Microsoft Windows sisteminin yaptığı gibi partitionları kullanmaz.
- UNIX'de, kullanıcıya C: ve D: gibi çeşitli "sürücüler" sunulmaz.
- Bunun yerine, kullanıcıya, kök dizininden başlayan veya / dizinleri içeren bir dizi alt dizin sunulur. "/ " Alt dizinleri ya aynı dosya sistemindeki alt dizinlerdir ya da yeni dosya sistemleri ve birimler için mount noktalarıdır.
- Örneğin, bir CD-ROM'a Windows'da E: sürücüsü olarak görülür, ancak Linux'ta /mnt/ cdrom üzerine mount edilmiş olabilir. Bu, kullanıcıların dizinleri değiştirerek sürücüleri değiştirmelerini sağlar ve çoğu durumda kullanıcılar bunu yaptığından habersizdir.

Unix ve Windows Disk Erişim Yöntemleri

- Sürücü bozulmasının etkisini en aza indirgemek ve verimliliği artırmak için, UNIX genelde her diski birkaç bölüme ayırır.
- Kök dizini **(/)** için bir birim, temel bilgileri saklar;
- Kullanıcının giriş dizinleri için ayrı bir birim **(/usr/)** olabilir ve uygulamalar kendi **(/home/)** biriminde bulunur.
- Tüm sistemler benzersizdir ve tamamen farklı volüm ve düzenine sahip olabilir. Bazı sistemler kök dizin için yalnızca bir büyük volüm kullanır ve sistemi parçalamaz.

Figure 4.3. Mount points of two volumes and a CD-ROM in (A) Microsoft Windows and (B) a typical UNIX system.



Volüm Oluşturmanın Genel Teorisi

- Daha büyük sistemler, birden fazla diski tek diske benzetmek için volüm oluşturma teknikleri kullanır.
- Veriler birden fazla diske yazılırsa ve bir disk başarısız olursa bir yedek kopya var demektir.
- Bunun bir diğer nedeni, daha fazla depolama alanı elde etmeyi kolaylaştırmaktır.
- Birim Volümleri, birden fazla partitionun toplam depolama alanını bir araya getirerek çalışır; böylece bir büyük volüm oluşturulur.
- Var olan verilere herhangi bir etkisi olmadan ek disklerle daha büyük volüm elde edilebilir.

Örnek Yapı

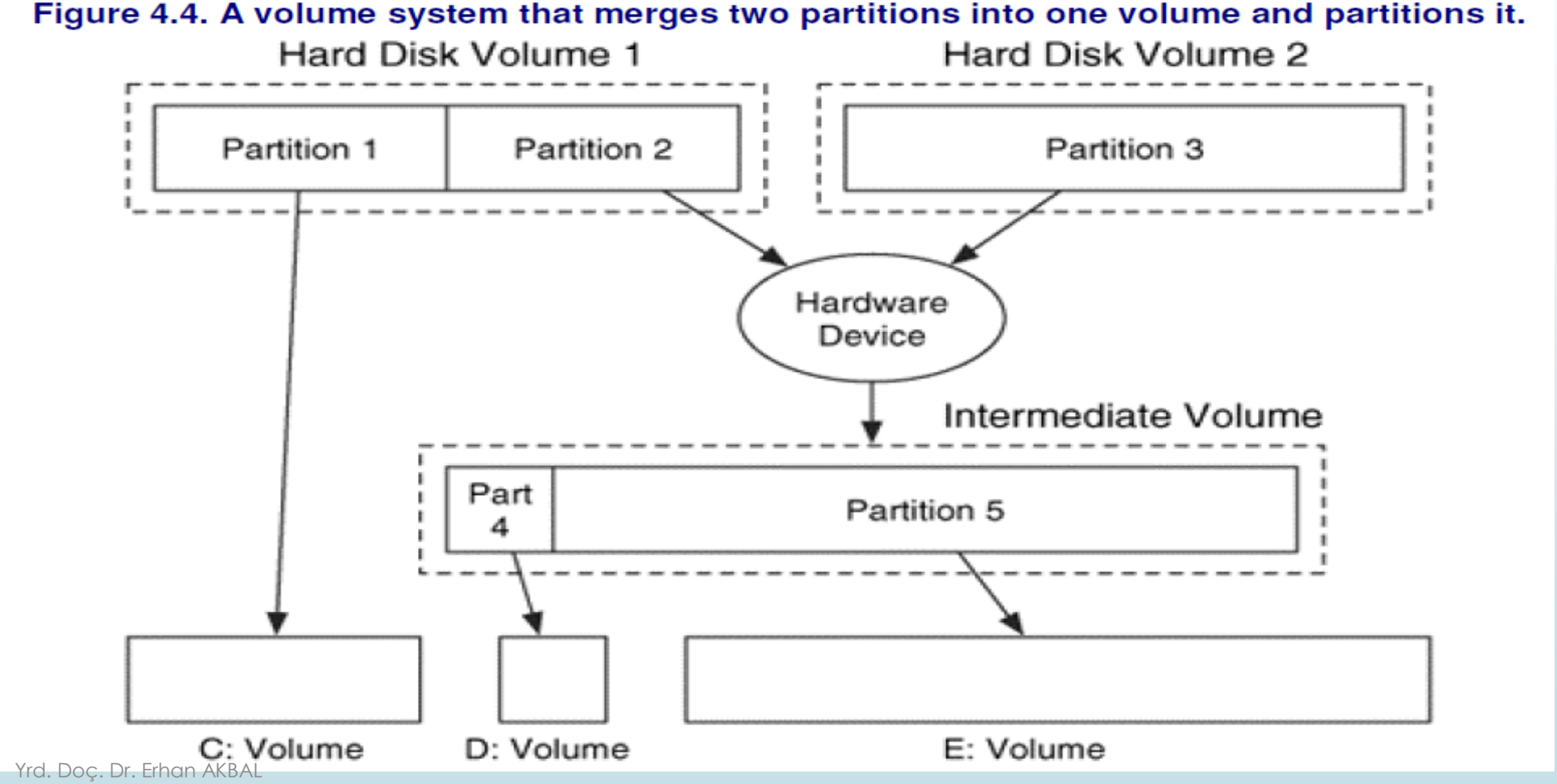


Figure 4.4. A volume system that merges two partitions into one volume and partitions it.
Hard Disk Volume 1
Hard Disk Volume 2
Partition 1
Partition 2
Partition 3
Hardware Device
Intermediate Volume
Part 4
Partition 5
C: Volume
D: Volume
E: Volume

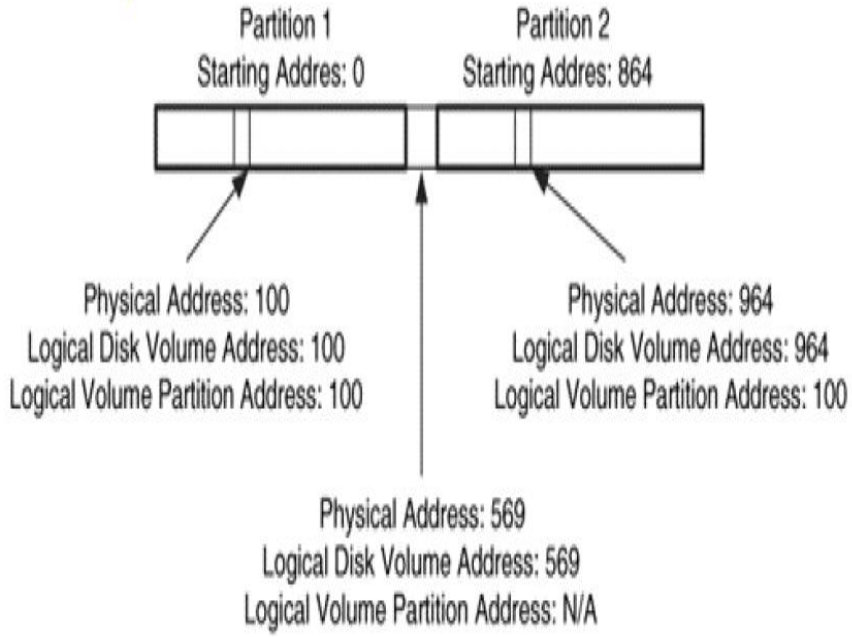
Sektör Adresleme

- Önceki bölümde disk geometri ve adresleme yapılarını gördük. En yaygın yöntem, LBA adresini kullanmaktır, bu da diskin ilk sektöründe 0 ile başlayan bir sayıdır. Bu adres, bir sektörün fiziksel adresidir.
- Bir volüm, sektörlerin toplamıdır ve onlara bir adres atamamız gerekir.
- Mantıksal volüm adresi, volümün başlangıcıyla ilgili sektörün adresidir.
- Bir disk bir volüm olduğu için fiziksel adres, disk volümü için mantıksal bir volüm adresi ile aynıdır.
- Bölümlerin başlangıç ve bitiş konumları, genellikle mantıksal volüm adresi kullanılarak tanımlanır.

Sektör Adresleme

- Bir sektör bir partitiona ayrılmazsa, logical partition volüm adresi olmayacaktır.
- Şekilde, iki partition olarak ayrılan ve partition olarak ayrılmayan alan bulunan bir örnek göstermektedir.
- İlk partition, sektör 0'dan başlar, dolayısıyla logical partition volüm adresleri logical disk volüm adresleriyle aynıdır.
- İkinci bölüm sektör 864'den başlar ve bu sektörlerin logical disk volüm adresi,logical partition volüm adresinden 864 sektör daha büyüktür.

Figure 4.5. The logical partition volume address is relative to the start of the partition while the logical disk volume address is relative to the start of the disk.



Analiz Temelleri

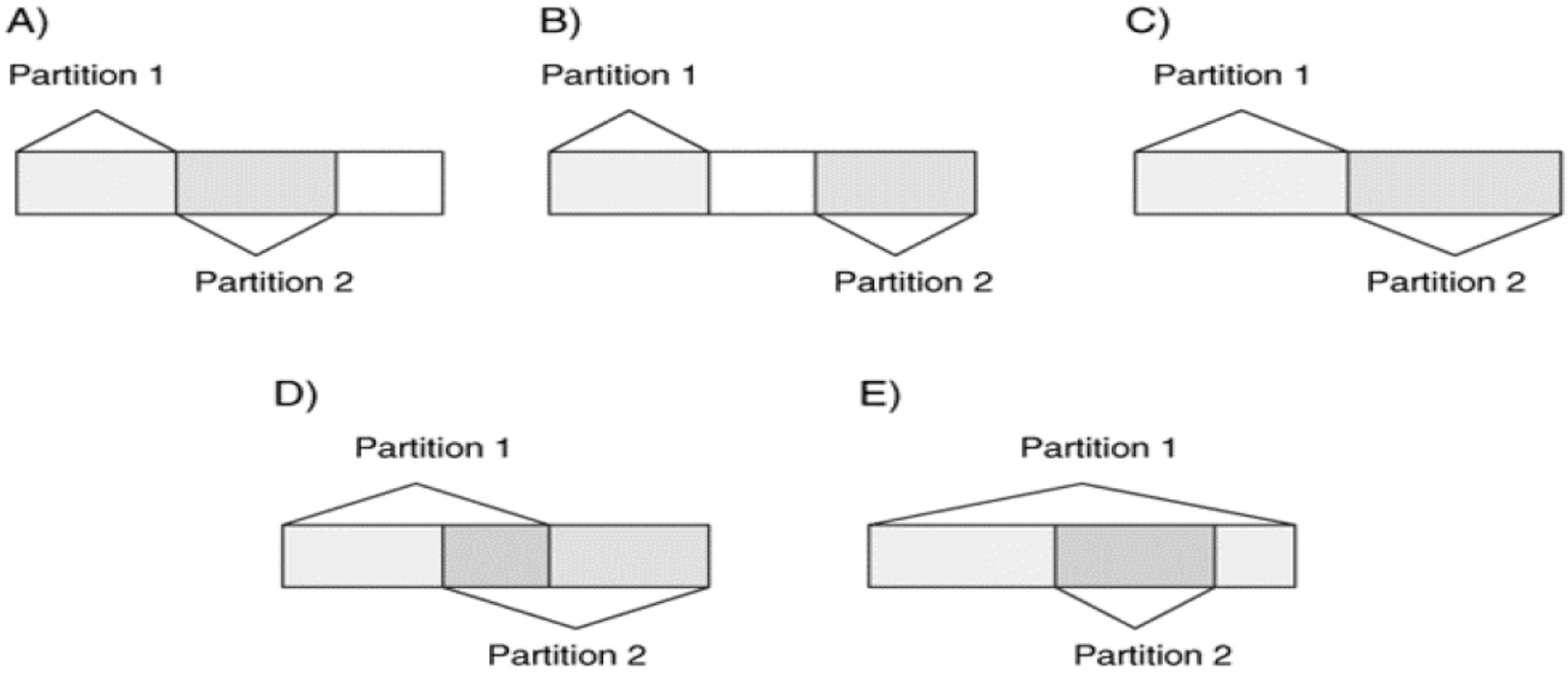
- Birçok araştırmacı bunu fark etmese de, volüm analizi sıkça yapılır.
- Çoğu durumda, bir araştırmacı, tüm sabit diski alır ve dosya sistemi içeriğini görüntülemek için imajı analiz yazılımına aktarır.
- Dosya sisteminin nerede başladığını ve bittiğini belirlemek için, bölüm tabloları analiz edilmelidir.
- Ayrıca, tüm bölümlerin bir bölüme atanması gerekmeyen ve önceki bir dosya sistemindeki verileri içerebileceği veya şüphelinin gizlemeye çalıştığı bölümün bölüm düzenini analiz etmek de önemlidir.
- Bazı durumlarda, bölümleme sistemi bozulabilir veya silinebilir ve otomatikleştirilmiş araçlar çalışmayabilir.

Analiz Teknikleri

- Volüm analizinin temel teorisi basittir. Partition sistemleri için, partition tablolarını bulmak, düzeni tanımlamak ve onları işlemek gerekir.
- Bir bölüm içindeki verileri analiz etmek için, ne tür verilere sahip olduğumuzu düşünmemiz gerekir. Genellikle, bu bir dosya sistemidir.
- Birleştirme, sistem bileşenlerini bir birim sistemde analiz etmek için, hangi birimleri birleştirip birleştirmediklerini açıklayan veri yapılarını bulmak ve işlemek zorundayız.

Tutarlılık Kontrolleri

Figure 4.6. Five examples of how two partitions can be organized relative to each other. The first three are valid, and the last two are not.



Partition İçeriklerinin Açılması ve Ayıklanması

- Bazı inceleme araçları girdi olarak bir partition imajı gerektirir veya partitionlar verilerini ayrı bir dosyaya çıkarmak isteyebiliriz.
- DOS tabanlı bir bölüm sistemine bir örnek inceleyelim.
- Partition tablosunun içeriğini listelemek için The Sleuth Kit'ten mmls aracı kullanılır.

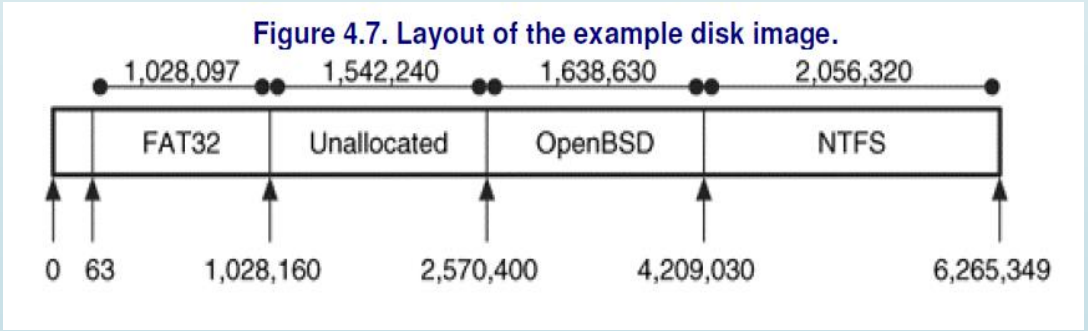
DD komutları

- Dd aracı komut satırı tabanlıdır ve birkaç bağımsız değişken alır.
- Partition verilerini ayıklamak için aşağıdakilere ihtiyacımız olacaktır:
- **If**: Okunacak disk imajı
- **Of**: Kaydedilecek çıktı dosyası
- **Bs**: Her defasında okunacak blok boyutu, 512 bayt varsayılan değerdir.
- **skip**: Okumadan önce atlanacak blokların sayısı, bs boyutlarının her biri
- **Count**: Girdiden çıktıya kopyalanacak blok sayısı, bs boyutunun her biri

Örnek Çıktı

- Mmls aracı, başlangıç sektörüne göre bölüm tablosu girdilerini düzenler ve bir bölüme ayrılmayan kesimleri tanımlar.
- 00 ve 01 numaralı ilk iki satır, birincil partition tablosu ve bölüm tablosu ile ilk bölüm arasındaki boş alanlardır.
- Çıktının 02 numaralı satırda FAT32 dosya sistemine sahip bir bölüm olduğunu,
- Satır 04'ün OpenBSD için bir bölüm olduğunu
- Satır 05'in bir NTFS dosya sistemi olan bir bölüm olduğunu görüyoruz.

```
# mmls -t dos disk1.dd
Units are in 512-byte sectors
     Slot    Start        End          Length       Description
00:  -----   0000000000   0000000000   0000000001   Table #0
01:  -----   0000000001   0000000062   0000000062   Unallocated
02:  00:00   0000000063   0001028159   0001028097   Win95 FAT32 (0x0B)
03:  -----   0001028160   0002570399   0001542240   Unallocated
04:  00:03   0002570400   0004209029   0001638630   OpenBSD (0xA6)
05:  00:01   0004209030   0006265349   0002056320   NTFS (0x07)
```



Partitionların Ayrılması

- Disk imajında dosya sistemi bölümlerini tam olarak ayırmak için, her partitionun başlangıç sektörünü ve boyutunu alır ve bunları aşağıda gösterildiği gibi dd'ye bağlarız.
- Bu komutlar, disk1.dd dosyasını girdi olarak alır ve çıktıyı, part1.dd, part2.dd ve part3.dd adlı dosyalara kaydeder.
- İlk bölüm, 1,028,097 blok kopyalanacak 63 blok atlayarak çıkarılır.
- Mmls çıktısında, bölüm 63'te başlamıştı sektör adreslerinin 0'dan başladığını düşünerek 63 atlatılması gerekiyor.

```
# dd if=disk1.dd of=part1.dd bs=512 skip=63 count=1028097
# dd if=disk1.dd of=part2.dd bs=512 skip=2570400 count=1638630
# dd if=disk1.dd of=part3.dd bs=512 skip=4209030 count=2056320
```

Bölümlendirilmiş Alanların Alınması

- Bu dd işlemi, bölümler arasındaki verileri ayıklamak için de kullanılabilir.
- Örneğin, mmls çıktısından, 1.028.160 ile 2.570.399 arasındaki sektörlerin kullanılmadığını biliyoruz.
- **# dd if=disk1.dd of=unalloc1.dd bs=512 skip=1028160 count=1542240**
- Hex editörler gibi diğer düşük seviye araçlar da sıralı sektörleri bir dosyaya kaydetme olanağı sağlar.

Silinen Partitionların Kurtarılması

- Bir adli soruşturmayı engellemek için kullanılan yaygın bir teknik, bir diski yeniden bölümlere ayırmak veya bölme yapılarını temizlemektir; böylece orijinal yapı gitmiş olur.
- Bazen bölüm yapıları bozulmuş bir sistemi kurtarmaya çalışılabilir.
- Bölüm kurtarma araçları, her bölümde bir dosya sisteminin bulunduğunu varsayarak çalışır.
- Birçok dosya sistemi sabit bir "magic" veya imza değeri olan bir veri yapısıyla başlar. Örneğin, bir FAT dosya sistemi, ilk sektörün 510 ve 511 numaralı baytlarında 0x55 ve 0xAA değerlerine sahiptir.
- Bölüm kurtarma araçları, bu imza değerlerini arar ve bir bölümün nereden başlamış olabileceğini belirler.

Silinen Partitionların Kurtarılması

- Arama aracı bir imza bulduğunda, belirli bir veri yapısı için geçerli olan değer aralığı üzerinde ek testler yapabilir.
- Örneğin, bir FAT dosya sistemi, bir kümede kaç sektör olduğunu tanımlayan bir alana sahiptir ve 1, 2, 4, 8, 16, 32, 64, 128 gibi 2'nin katlarına sahip olmalıdır.
- Her aracın arama mekanizması değişiklik gösterebilir. Bazı araçlar her sektörü inceler ve bilinen imzalara kıyaslar.
- Diğer araçlar yalnızca silindir sınırlarını aramaktadır çünkü partitionlar genellikle silindir sınırları üzerinde oluşturulmuştur.
- Diğerleri, dosya sisteminin ne kadar büyük olduğunu öğrenmek ve daha bilinen veri yapılarını aramadan önce sonuna atlamak için dosya sistemi veri yapılarındaki verileri kullanabilir.

Kullanılan Araçlar

```
# gpart -v disk2.dd
* Warning: strange partition table magic 0x0000.
[REMOVED]
Begin scan...
Possible partition(DOS FAT), size(800mb), offset(0mb)
   type: 006(0x06)(Primary 'big' DOS (> 32MB))
   size: 800mb #s(1638566) s(63-1638628)
   chs:  (0/1/1)-(101/254/62)d (0/1/1)-(101/254/62)r
   hex:  00 01 01 00 06 FE 3E 65 3F 00 00 00 A6 00 19 00

Possible partition(DOS FAT), size(917mb), offset(800mb)
   type: 006(0x06)(Primary 'big' DOS (> 32MB))
   size: 917mb #s(1879604) s(1638630-3518233)


   chs:  (102/0/1)-(218/254/62)d (102/0/1)-(218/254/62)r
   hex:  00 00 01 66 06 FE 3E DA E6 00 19 00 34 AE 1C 00

Possible partition(Linux ext2), size(502mb), offset(1874mb)
   type: 131(0x83)(Linux ext2 filesystem)
   size: 502mb #s(1028160) s(3839535-4867694)
   chs:  (239/0/1)-(302/254/63)d (239/0/1)-(302/254/63)r
   hex:  00 00 01 EF 83 FE 7F 2E 2F 96 3A 00 40 B0 0F 00
```

- Manipulate file systems
  - btrfs
  - ext2 / ext3 / ext4
  - fat16 / fat32
  - hfs / hfs+
  - linux-swap
  - lvm2 pv
  - nilfs2
  - ntfs
  - reiserfs / reiser4
  - ufs
  - xfs

- Çıktıdan iki FAT bölümünün ve bir Ext2 bölümünün muhtemel olduğunu görüyoruz. 'Size:' satırının sonundaki alan, bölümün sektörlerdeki yerini gösterir. -v bayrağı belirtilmemişse, sektör konumu yazdırılmayacaktı.
- Benzer bir araç Christophe Grenier'in TestDisk'idir
- Bu analiz aracı yalnızca temel silme veya bölüm tablosu bozulması oluştuğunda çalışır.