Dosya Sistemleri Yapısı ve Analizi

Hafta 8 – Dosya Sistem Analizi Dersi

Giriş

- Dosya sistemi analizi bir volüme ait (yani bir bölüm veya disk) verileri inceler ve onları bir dosya sistemi olarak yorumlar.
- Bu işlemden elde edilen sonuçlar dosyaları bir dizinde listelemek, silinen içeriği kurtarmak ve bir sektörün içeriğini görüntülemek gibi örnekler içerebilir.

Dosya Sistemi Nedir?

- Bilgisayarlar, verilerin uzun süreli depolanması ve alınması için bir yöntem gerektirirler.
- Dosya sistemleri, kullanıcıların dosyaları ve dizinleri hiyerarşisine veri depolamak için bir mekanizma sağlar.
- Bir dosya sistemi, bilgisayarın nerede bulacağını bildiği şekilde düzenlenmiş yapısal ve kullanıcı verilerinden oluşur.
- Çoğu durumda, dosya sistemi herhangi bir bilgisayardan bağımsızdır

Örnek

- Bir doktor ofisinde bir dizi dosya dolabı düşünün.
- Ulusal *Tıbbi Kayıt Dosyalama Prosedürleri Birliği* (NAMRFP) tüm hasta kayıtlarının dosyalama dolaplarına düzenlenmesi ve hastanın soyadına göre sıralanması gerektiğini belirtebilir.
- Kaydı tanımlamak için kullanılan etiket, İngilizce olarak yazılmalı ve soyadın arkasında ilk ad olmalıdır.
- Bu prosedürde eğitim almış herhangi bir kişi, prosedürü kullanan bir büroda hasta kayıtlarını dosyalayabilir ve geri alabilir.
- Ofiste 100 hasta, bir dosya dolabı veya 100.000 hasta ve 25 dosya dolabı var mı önemli değildir.
- Önemli olan tek şey, dosyalama dolabının ne olduğunu tanımak, dosyayı nasıl açacağını bilmek ve etiketleri okumayı ve oluşturmayı bilmektir.

Dosya Sistemi Nedir?

- Dosya sistemleri bu kayıt saklama prosedürlerine benzerdir.
- Dosya sistemlerinde, bir dosyayı bir diskete veya bir depolama alanında saklamak için kullanılabilecek belirli yordamlar ve yapılar bulunur.
- Her bir dosya sistemi benzersiz bir boyuta sahiptir, ancak temel yapısı, dosya sisteminin türünü destekleyen herhangi bir bilgisayarın onu işleyebilmesine olanak tanır.
- Bazı veriler, dosyanın içyapısı ve organizasyonuna ihtiyaç duyar.

Veri Kategorileri

- Farklı dosya sistemlerinin daha kolay karşılaştırılabilmesi için temel bir referans modelinin bulunması gereklidir.
- Örneğin, bir referans modeli, FAT ve Ext3 dosya sistemleri arasındaki farkı karşılaştırmayı kolaylaştırır.
- Bu temel model için
  - **1- Dosya sistemi**
  - **2- İçerik**
  - **3- Metadata**
  - **4- Dosya Adı**
  - **5- Uygulama**
- olmak üzere beş kategori kullanılır.

1- Dosya Sistemi

- Tüm dosya sistemleri genel bir yapıya sahiptir, ancak dosya sisteminin her örneği benzersizdir, çünkü benzersiz bir boyuta sahiptir ve performans için ayarlanabilir.
- Dosya sistemi kategorisindeki veriler, size belirli veri yapılarının nerede bulunacağını ve bir veri biriminin ne kadar büyük olduğunu söyleyebilir.
- Bu kategorideki verileri, belirli bir dosya sistemi için bir harita olarak düşünebilirsiniz.

2- İçerik Kategorisi

- Bir dosyanın gerçek içeriğini oluşturan verileri içerir; bu da, öncelikle dosya sistemimize sahip olmamızın nedenidir.
- Bir dosya sistemindeki verilerin çoğu bu kategoriye aittir ve genellikle standart boyutlu yapılardan oluşan bir koleksiyon halinde düzenlenmiştir.
- Her dosya sistemi, clusters (kümeler) ve bloklar gibi alanlara farklı bir ad atar.

3- Metadata Kategorisi

- Bir dosyayı tanımlayan verileri içerir; Bunlar veri tanımlayan verilerdir.
- Bu kategori, dosya içeriğinin nerede saklandığı, dosyanın ne kadar büyük olduğu, dosyanın en son ne zaman okunduğu veya yazıldığı tarih ve saatler ile erişim denetim bilgileri gibi bilgileri içerir.
- Bu kategorinin dosyanın içeriğini içermediğini ve dosyanın adını içermeyebileceğini unutmayın.
- Bu kategorideki veri yapılarına örnekler, FAT dizin girdileri, NTFS Ana Dosya Tablosu (MFT) girdileri ve UFS ve Ext3 inode yapılarını içerir.

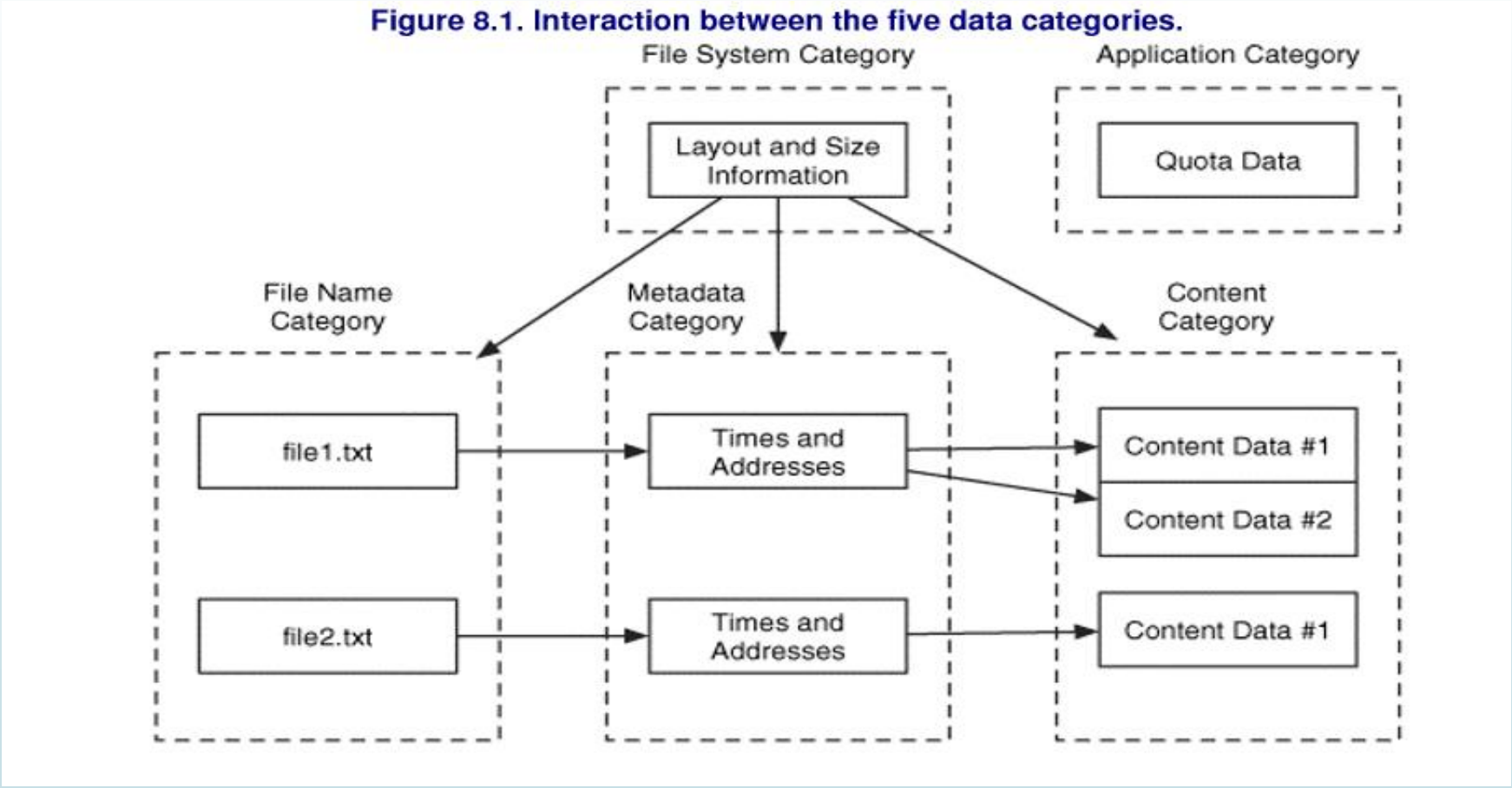
4- Dosya Adı Kategorisi

- Her dosyaya bir ad atayan verileri içerir.
- Çoğu dosya sisteminde, bu veriler bir dizinin içeriğinde bulunur ve karşılık gelen metadata adresine sahip dosya adlarının bir listesidir.
- Dosya adı kategorisi bir ağdaki bir ana makine adına benzer. Ağ cihazları, insanların hatırlayamayacağı IP adresleri kullanarak birbirleriyle iletişim kurar.
- Bir kullanıcı bir uzak bilgisayarın ana bilgisayar adına girdiğinde, iletişimin başlatılabilmesi için yerel bilgisayarın adı IP adresine çevirmesi gerekir.

5- Uygulama Kategorisi

- Özel özellikler sağlayan verileri içerir.
- Bu veriler bir dosyanın okunması veya yazılması işlemi sırasında gerekli değildir ve çoğu durumda dosya sistemi belirtimine dahil edilmeleri gerekmez.
- Veriler normal bir dosya için değil, dosya sistemi yapısına uygulanması daha verimli olabilir.
- Bu kategorideki verilere örnek olarak kullanıcı kotası istatistikleri verilebilir.
- Bu veriler bir soruşturma sırasında yararlı olabilir, ancak bir dosyayı yazmaya ve okumaya ihtiyaç duyulmadığı için diğer verilere göre daha kolay sahte olabilirler.

Kategoriler Arası İlişki



Temel ve Temel olmayan veriler

- Asıl ve asıl olmayan veriler arasında ayrım yapmak neden önemlidir?
- Önemlidir çünkü temel verilere güvenmek zorundayız, ancak temel olmayan verilere güvenmek zorunda değiliz.
- Örneğin, tüm dosya sistemleri, dosya içeriğinin saklandığı yere işaret eden bir kayıt tutar. Bu değer, gerçek olması gerektiği için önemlidir. Eğer yanlışsa, kullanıcı dosyayı okuyamayacaktır.
- Öte yandan, bir zaman değeri veya Kullanıcı Kimliği, gerçek olması gerekmediği için asıl veri değildir. Zaman değeri güncellenmezse, dosyayı okumaya veya dosyaya yazmaya çalışırken kullanıcıyı etkilemez. Bu nedenle, asıl verilere, asıl olmayan verilerden daha fazla güvenmeliyiz çünkü dosya sisteminin dosyaları kaydetmesi ve geri yüklemesi için gereklidir.

Temel ve Temel olmayan veriler

- Dosya sisteminin kullanıldığı işletim sisteminin bilinmesinin, dosya sisteminin türünü bilmek kadar önemlidir.
- Dosya kurtarma işlemini tartışırken bir dosyanın bir FAT dosya sisteminden nasıl kurtarılabileceğini sormak yeterli değildir.
- Bunun yerine, Windows 98'de bir FAT dosya sisteminde silinmiş bir dosyanın nasıl kurtarılabileceğini sormalıyız.
- Birçok işletim sistemi, FAT dosya sistemlerini uygular ve her biri farklı teknikleri kullanarak bir dosyayı silebilir.

Dosya Sistemi Kategorisi

- Dosya sistemi kategorisi, bu dosya sisteminin benzersizliğini ve diğer önemli verilerin nerede olduğunu belirten genel verileri içerir.
- Dosya sisteminin ilk sektörlerinde standart bir veri yapısında bulunur
- Bu bilgi ile dosya sisteminin boyutuna bağlı olarak değişebilecek diğer verilerin yerleri bulunabilir.
- Bu verilerden herhangi biri bozulursa veya kaybolursa, yedek kopyalar bulmanız veya değerlerin ne olduğunu tahmin etmeniz nedeniyle ek analiz gerekir.

Analiz Teknikleri

- TSK, bir dosya sistemi için dosya sistemi kategorisi verilerini görüntüleyen **fsstat** adlı bir araca sahiptir.
- Hangi bilgisayarda bir dosya sistemi oluşturulduğunu belirlemeye çalışıyorsanız, bir birim kimliği veya sürümü aranır.

İçerik Kategorisi

- İçerik kategorisi, dosyaları ve dizinlere tahsis edilen depolama konumlarını içerir; böylece veri kaydedebilirler.
- Bu kategorideki veriler genellikle eşit boyutlu gruplar halinde düzenlenir; her dosya sistemi kendileri için benzersiz bir ada sahiptir ve **Cluster** (küme), **Block** (blok) gibi veri birimi isimleri kullanılır.
- Bir veri birimi **allocated** (ayrılmış) veya **unallocated** (ayrılmamış) durumda olur.
- Genellikle, her veri biriminin tahsis durumunu takip eden bazı veri yapısı türleri vardır.
- Yeni bir dosya oluşturulduğunda veya var olan bir dosya daha büyük hale getirildiğinde, işletim sistemi ayrılmamış bir veri birimi arar ve bir dosyaya atar.
- Bir dosya silindiğinde, dosyaya ayrılan veri birimleri unallocated (ayrılmamış) duruma ayarlanır ve yeni dosyalara atanabilir.
- ***Çoğu işletim sistemi ayrılmamış veri biriminin içeriğini silemez.***

İçerik Kategorisi

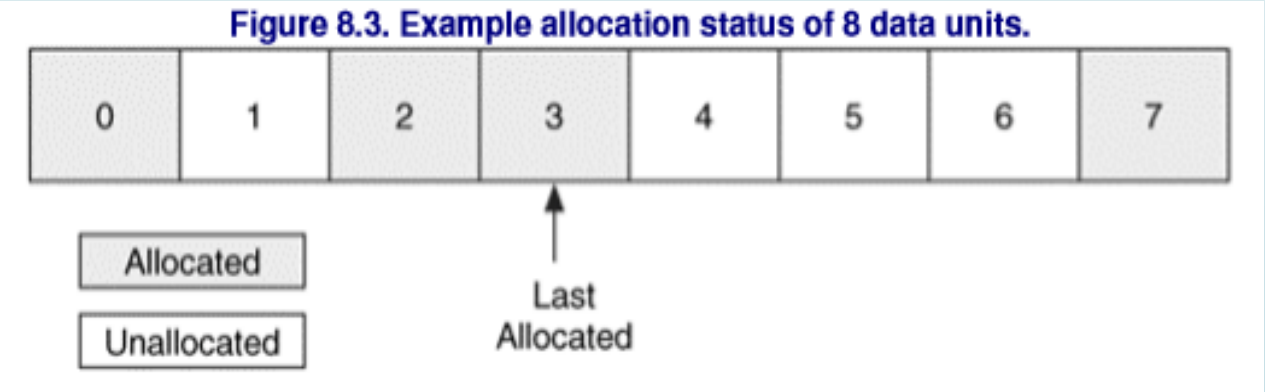
- İçerik kategorisinin analizi silinen verileri kurtarmak ve düşük seviye aramalar yapmak için gerçekleştirilir.
- Bu kategoride çok fazla veri vardır; bu nedenle, genellikle elle analiz edilmez.
- Referans olarak, bir araştırmacı beş saniyede 512 baytlık bir sektörü inceleyebilirse, günde 12 saat arama yapması halinde 40GB'lık bir sürücüyü 388 günde inceleyebilir.

Allocation (Tahsis) Yöntemleri

- Bir işletim sistemi, veri birimlerini tahsis etmek için farklı stratejiler kullanabilir.
- Tipik olarak, bir OS ardışık veri birimlerini tahsis eder, ancak bu her zaman mümkün değildir.
- Bir dosyada ardışık veri birimi yoksa **parçalı(fragmented)** olarak adlandırılır.

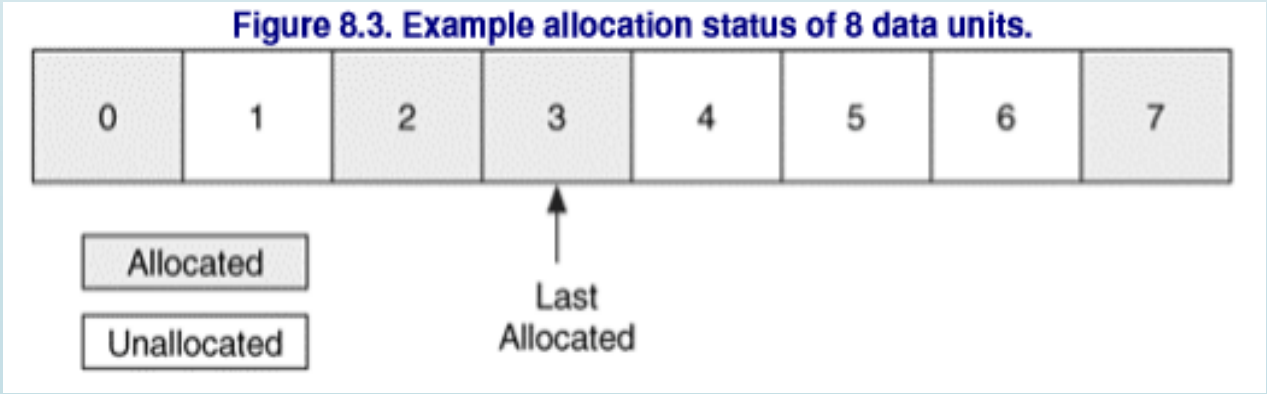
First Available (İlk Mümkün olan)

- Dosya sisteminin ilk veri biriminden başlayarak, kullanılabilir bir veri birimi arar.
- Bir veri birimi bu strateji kullanılarak tahsis edildikten sonra ikinci bir veri birimine ihtiyaç duyulur ve arama, dosya sisteminin başında tekrar başlar.
- Dosya bir bütün olarak tahsis edilmediğinden, bu strateji kolaylıkla parçalanmış dosyalar üretebilir.



Next Available

- Kullanılabilir bir alan aramaya dosya sisteminin en başından başlamak yerine, en son atama yapılan adresten başlar.
- Örneğin, Şekil 8.3.'deki veri birimi tahsis edilmiş olsa bir sonraki arama 0 yerine 4. Veri biriminden başlar.
- Bu algoritma veri kurtarma olasılığı bakımından daha dengelidir. Çünkü dosya sisteminin başındaki veri birimleri, en sonra kadar atama işlemi yapılmadan üzerine yazma işlemine tabi tutulmazlar.



Best Fit (En Uygun)

- ihtiyaç duyulan miktarda ardışık veri birimini bulmaya çalışır.
- Bu yöntem bir dosyanın ne kadar veri birimine ihtiyaç duyduğunun bilindiği durumlarda işe yarar.
- Ancak, dosyanın boyutu büyüdüğünde , yeni veri birimleri başka bir alana kaydedilir ve dosya yine parçalanmış olur.
- Eğer bu algoritma tüm verinin sığabileceği bir alan bulamazsa first veya next available yöntemleri kullanılabilir.
- Her işletim sistemi bir dosya sistemi için bir ayırma stratejisi seçebilir. Bazı dosya sistemleri hangi stratejinin kullanılacağını belirtir. Belirtilmeyen sistemlerin hangi yöntemi kullandığının bulunması için bir uygulama ile test edilmesi gerekir.

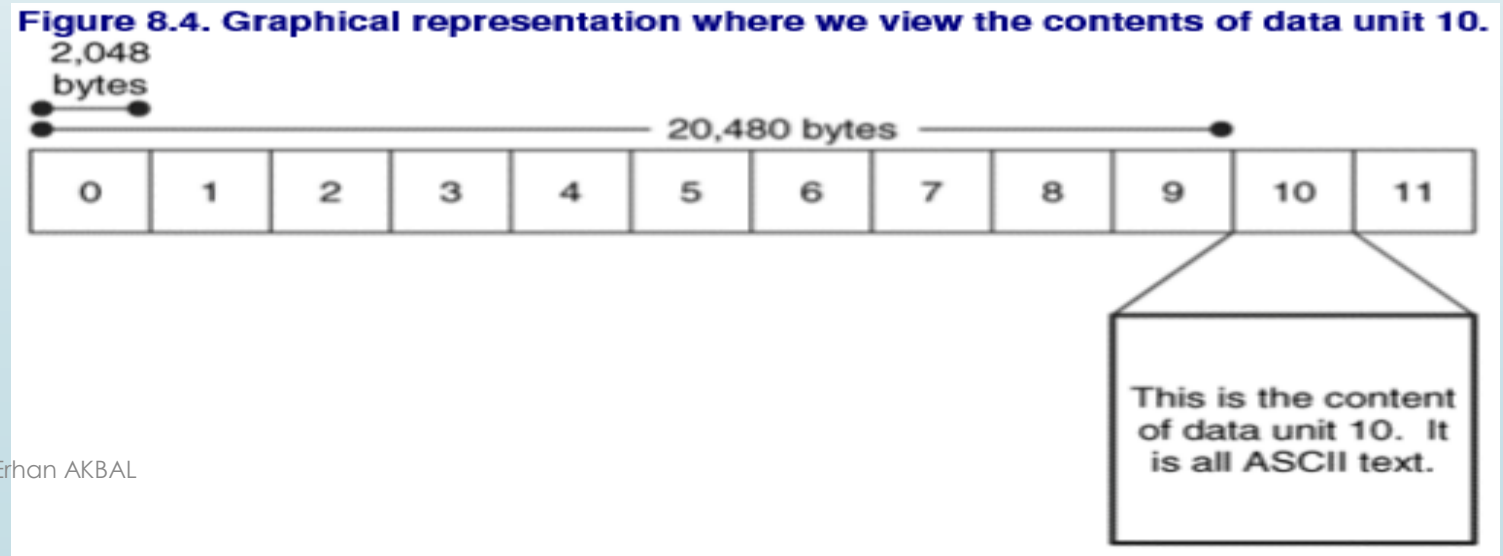
Kötü (Bad) Veri Birimleri

- Birçok dosya sistemi, bir veri birimini kötü olarak işaretleme yeteneğine sahiptir.
- Bu, hataları işleme kapasitesine sahip olmayan eski sabit diskler için gerekliydi.
- İşletim sistemi bir veri biriminin kötü olduğunu algılayacak ve bir dosyaya tahsis edilmeyecek şekilde işaretleyecektir.
- Günümüz modern sabit diskleri kötü bir sektör tespit edebilir ve yedekle değiştirir, böylece dosya sistemi işlevselliği yitirilmez.

Veri Birimi Görüntüleme

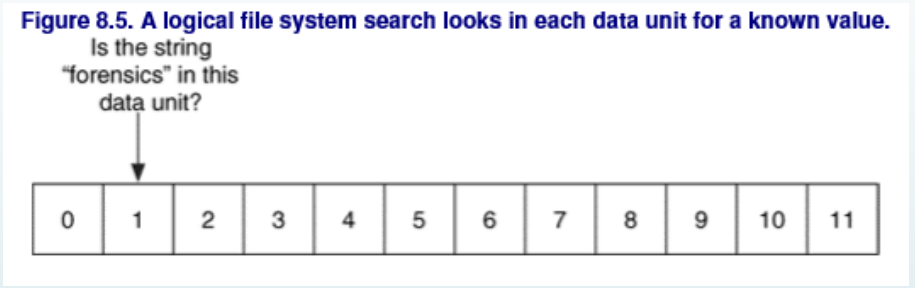
- Araştırmacı veri biriminin mantıksal dosya sistemi adresini girer ve bir araç veri biriminin bayt veya sektör adresini hesaplar.
- Araç daha sonra bu konuma giderek buradaki verileri okur. Örneğin, 0. Veri biriminin 0. Bayt ofsetten başladığı ve her veri biriminin 2,048 bayt olduğu düşünülürse; 10. Veri biriminin bayt ofset değeri Şekil'de görüldüğü gibi 20,480 bayt olur.

Figure 8.4. Graphical representation where we view the contents of data unit 10.



Dosya Sistem Düzeyinde Mantıksal Arama

- Bu teknikte, kanıtların hangi içeriğe sahip olacağını biliyoruz, ancak nerede olduğunu bilmiyoruz.
- Mantıksal dosya arama sistemi, her bir veri biriminde belirli bir cümle veya değer arar. Örneğin, "forensics" veya belirli bir dosya başlığı değeri aramak isteyebilirsiniz.
- Maalesef dosyalar her zaman ardışık olarak kaydedilmezler, eğer aranan değer ardışık olmayan iki veri birimine kaydedilmiş ise, mantıksal dosya sistemi araması bu değeri bulamaz.



Bu arama tekniği, sektörlerin fiziksel sıralamasını kullandığı için **fiziksel arama olarak adlandırılır.** Bu yöntem tek bir disk analiz edildiğinde düzgün çalışırken, disk spanning ve RAID kullanıldığı durumlarda düzgün çalışmamaktadır. Çünkü, bu gibi sistemlerde sektörlerin sırası fiziksel sıra değildir.

Data Unit Allocation Durumları

- Burada ilk 12 veri birimi için bitmap gösterilmektedir.
- Bitmap, her bir veri birimi için bir bite sahip olan bir veri yapısıdır.
- Eğer bitler 1 ise o birime atama yapılmıştır, eğer 0 ise yapılmamıştır.
- Eğer atama yapılmamış olan (tahsis edilmeyen) veri birimleri çıkarılırsa 2,6,7 ve 11 numaralı veri birimleri elde edilir.

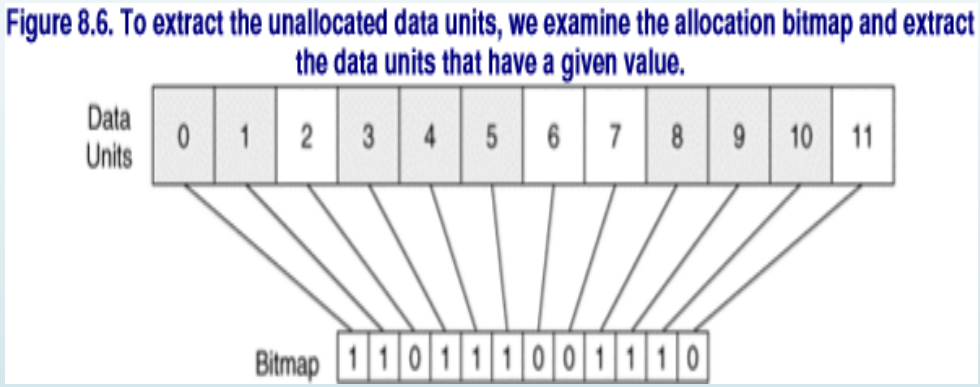


Figure 8.6. To extract the unallocated data units, we examine the allocation bitmap and extract the data units that have a given value.

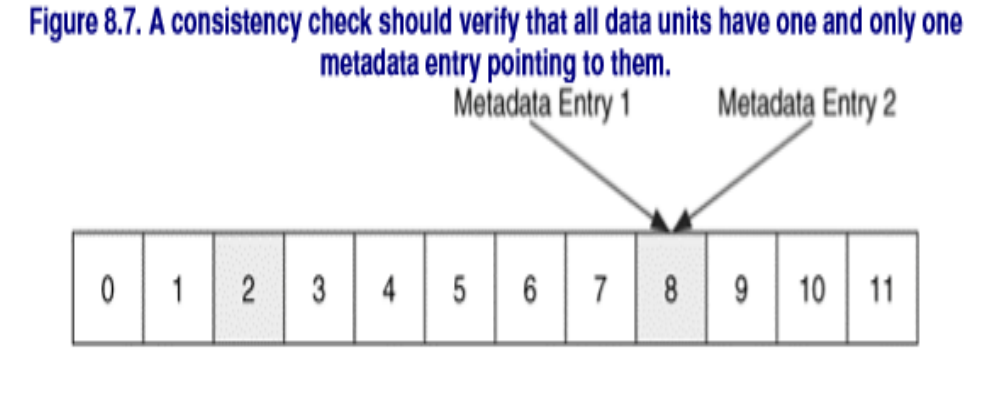
Analiz Araçları

- TSK'da, **dls** aracını kullanarak, ayrılmamış verileri bir dosyaya ayıklayabilirsiniz.
- Veriler bulunduktan sonra, dosya sistemindeki hangi veri birimini içerdiğini öğrenmek isteyebilirsiniz ve bunu belirlemek için **dcalc** aracını kullanabilirsiniz.
- TSK, tahsis edilmemiş verileri, tahsisat durumu ayrılmamış olarak ayarlanan veri birimleri olarak değerlendirir.
- Dosya sistemi tarafından kullanılan ve bir tahsisat durumuna sahip olmayan veri birimleri varsa, bunlar ayrılmış olarak kabul edilir.

Tutarlılık Kontrolleri

- Tutarlılık denetimleri her veri kategorisi için önemli bir analiz tekniğidir.
- Dosya sisteminin şüpheli bir durumda olup olmadığının belirlenmesine izin verirler.
- İçerik kategorisinde bir tutarlılık denetimi, metadata kategorisindeki verileri kullanır ve ayrılan her veri biriminin kendisine işaret eden tam olarak tahsis edilmiş bir metadata girişi olduğunu doğrular.
- Bu, bir kullanıcının, veri için bir ad olmadan, tahsisat durumunu bir veri birimine manuel olarak ayarlamasını önlemek için yapılır.

Tahsis edilmiş veri birimlerine karşılık gelen bir metadata kaydı yoksa bu alanlara orphan data unit – yetim veri birimi adı verilir.



Silme (Wipe) Teknikleri

- Çoğu silme(wipe) veya "güvenli silme" araçları içerik kategorisinde çalışır ve bir dosyanın tahsis edilen veri birimlerine veya kullanılmayan tüm veri birimlerine sıfır veya rastgele veri yazar.
- Ayrılmamış tüm veri birimleri sıfırlar veya rasgele değerler içeriyorsa, silme aracı ile silindiğinden şüphelenilmelidir.
- Araç rasgele değerler yazıyorsa veya mevcut diğer veri birimlerini kopyalıyorsa, uygulama düzeyinde bir araç kullanıldığına dair kanıt olmadan algılama neredeyse imkânsızdır.
- Tabii ki, sistemde bir silme aracı bulursanız, kullanılıp kullanılmadığını ve son erişim süresinin ne olduğunu kontrol etmelisiniz

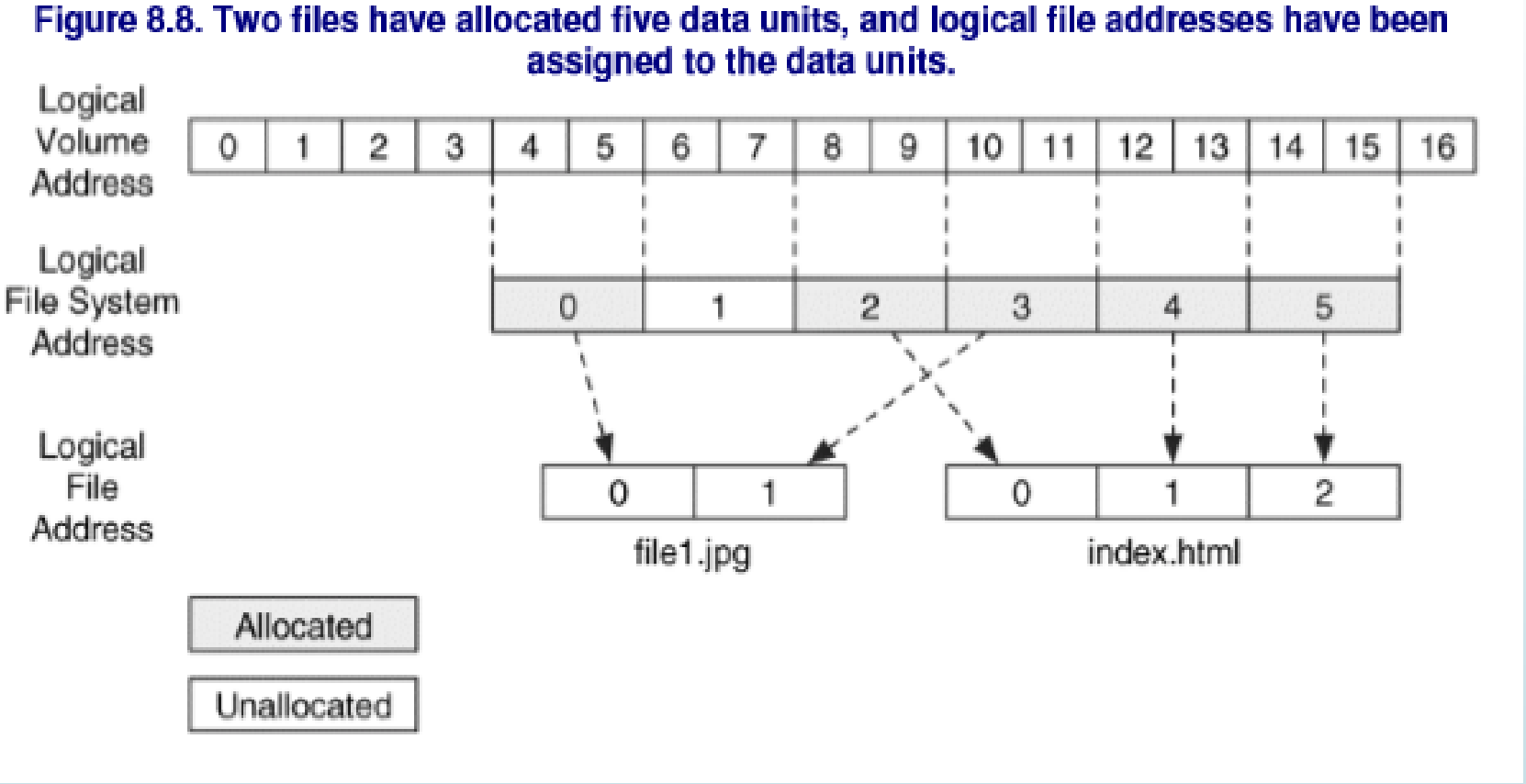
Metadata Kategorisi

- Metadata kategorisi, açıklayıcı verilerin bulunduğu yerdir.
- Burada, örneğin bir dosyanın tahsis ettiği veri birimlerinin adreslerini ve son erişim zamanını bulabiliriz.
- Birçok metadata yapısı sabit veya dinamik uzunlukta bir tabloda saklanır ve her kayıt bir adrese sahiptir.
- Bir dosya silindiğinde, metadata girişi ayrılmamış duruma ayarlanır ve işletim sistemi, girdideki bazı değerlerin silinmesine neden olabilir.
- Bu kategori, diğer kategorilere kıyasla daha önemli olmayan verilere sahip olma eğilimindedir.

Mantıksal Dosya Adresi

- Bir dosyaya ayrılan bir veri birimi de mantıksal bir dosya adresine sahiptir.
- Bir veri biriminin mantıksal dosya adresi, tahsis edildiği dosyanın başlangıcına göredir. Örneğin, iki veri birimi ayrılmış bir dosya varsa, ilk veri biriminin mantıksal dosya adresi 0 olur ve ikincisinin mantıksal dosya adresi 1 olurdu.
- Benzersiz bir mantıksal dosya adresi oluşturmak için dosyanın adı veya metadata adresi gereklidir.

Örnek Mantıksal Adres



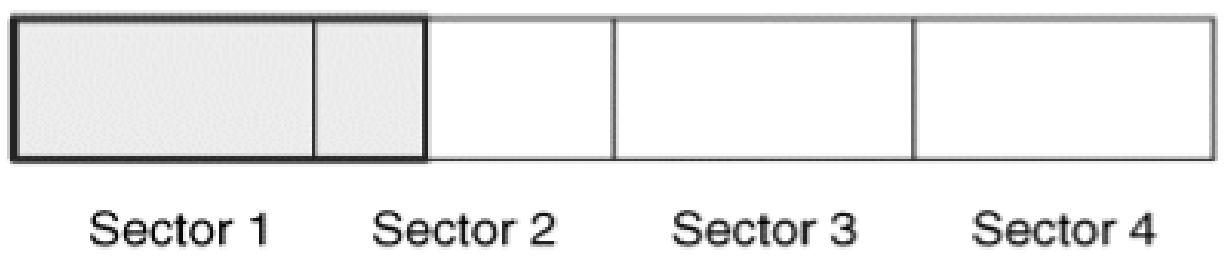
Slack Space (Artık Alan)

- Slack Space çoğu insanın bir şekilde duyduğu yaygın kullanılan bir adli bilişim tabiridir.
- Slack Space, bir dosya boyutu bir veri birimi boyutunun katları olmadığında oluşur.
- Bir dosyanın yalnızca küçük bir kısmı olsa bile tam bir veri birimi tahsis etmesi gerekir ve son veri biriminde kullanılmayan baytlara artık alan adı verilir.
- Bazı bilgisayarlar kullanılmayan baytları silmez, bu yüzden artık alan, önceki dosyalardan veya bellekten verileri içerir. Artık alan olarak iki alan vardır.
- İlk alan, dosyanın sonuyla dosyanın bittiği sektörün sonu arasında bulunur.
- İkinci alan, veri birimindeki kalan kullanılmayan sektörlerdir.

Slack Space – Örnek

- Dosyamız 612 bayt, bu yüzden kümedeki ilk sektörün tamamını ve ikinci sektörün 100 baytını kullanıyor.
- İkinci sektörün geri kalan 412 baytı, OS'lerin seçtiği veriyle doldurulmuştur.
- Üçüncü ve dördüncü sektörler OS tarafından sıfırlarla silinebilir veya dokunulmayabilir

Figure 8.9. Slack space of a 612-byte file in a 4096-byte cluster.

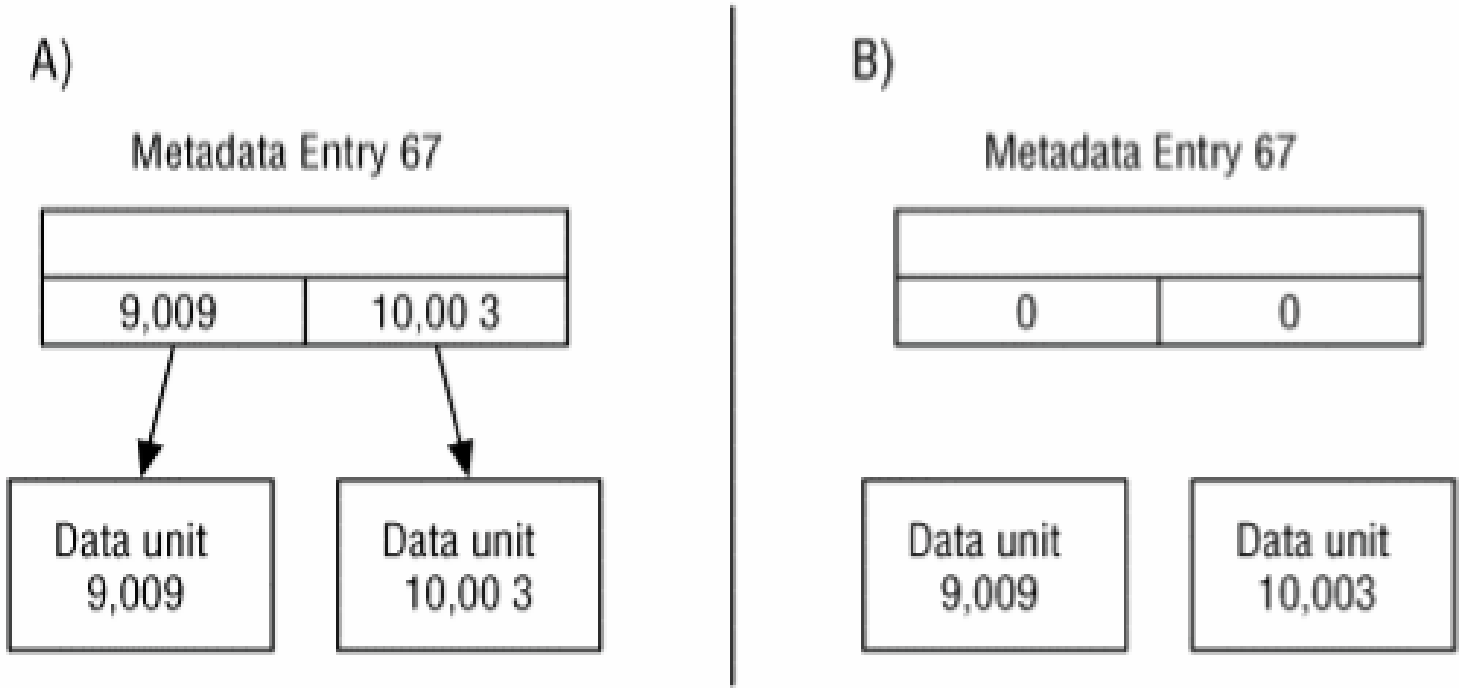


Metadata Tabanlı Dosya Kurtarma

- Silinen dosyaları kurtarmak için iki temel yöntem vardır:
- Metadata tabanlı ve uygulama tabanlı.
- Metadata tabanlı kurtarma, silinmiş dosyadaki metadatalar hala mevcut olduğunda çalışır.
- Metadatalar silinmişse veya metadata yapısı yeni bir dosyaya yeniden tahsis edilmişse, uygulama tabanlı tekniklere güvenmeniz gerekecektir.
- Dosyanın metadata yapısını bulduktan sonra kurtarma kolaydır. Ayrılan bir dosyanın içeriğini okumaktan farklı değildir.

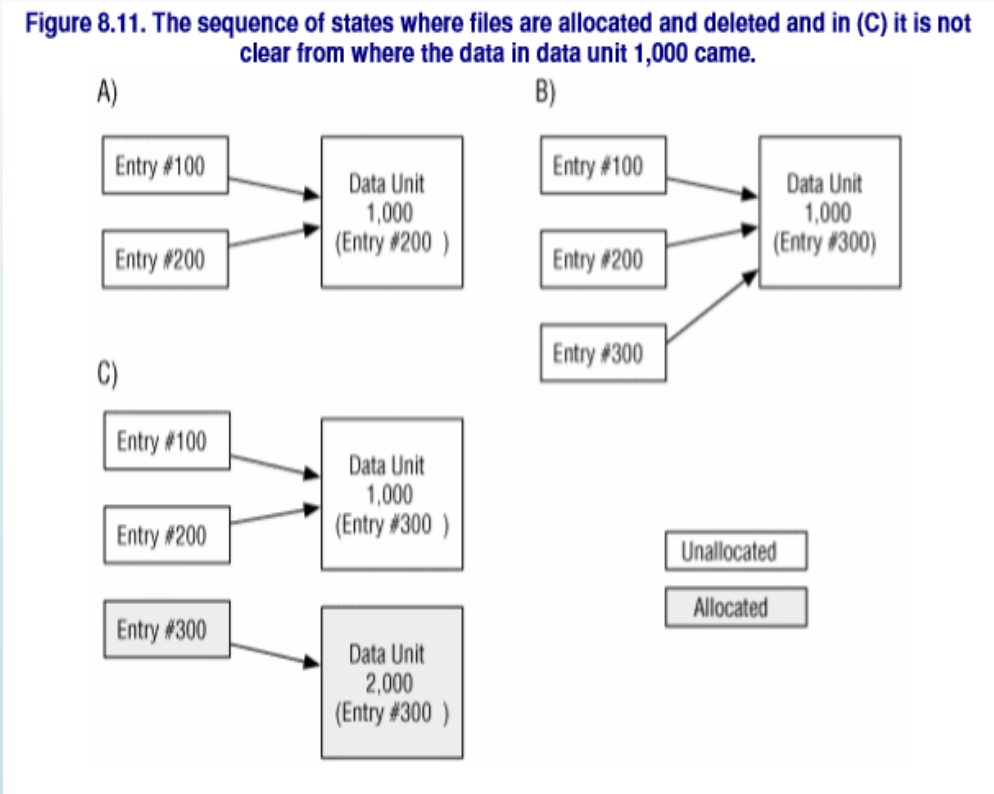
Örnek Yapı

Figure 8.10. Two scenarios where in (A) the data unit pointers are not wiped when the entry is unallocated and in (B) they are wiped.



Örnek Yapı -2

- Metadata kaydı 100, 1.000 veri birimini tahsis eder ve ona veri kaydeder.
- 100 kaydı için dosya daha sonra silinir ve hem kayıt 100 hem de 1000 veri birimi unallocate edilir.
- Metadata kaydı 200 oluşturulur ve 1.000 veri birimini yeniden ayırır.
- Daha sonra bu dosya da silinir. Bu sistem analiz edilirse aynı veri birimi adresine sahip iki ayrılmamış metadata kaydı bulunur.



Yorumlama

- Bu bölüme en son hangi dosyanın kaydedildiğini belirlememiz gerekir. Bunu yapmanın bir yöntemi, her kayıttaki zaman bilgilerini kullanmaktır, ancak buna güvenemeyebiliriz.
- Başka bir yöntem ise, metadatalar bu bilgiyi kaydederse, dosya türünü kullanmaktır.
- Örneğin, metadata girişi 200 bir dizine ait olabilir, bu nedenle bir dizinin biçimine sahip olup olmadığını görmek için veri birimi 1.000'in içeriğini analiz edebiliriz.
- Kayıt 200'ün giriş 100'den sonra olduğunu tespit edebilsek bile kayıt 200'ün tahsis edilen en sonuncu girdi olduğunu bilemeyiz.
- Bunu göstermek için kayıt 200 tarafından tanımlanan 1000 veri biriminin kayıt 300 tarafından tahsis edildiğini ve tekrar silindiğini düşünün.
- Daha sonra, yeni bir dosya yaratıldı ve kayıt 300, bunun için yeni veri birimi olan 2.000 veri birimi ile yeniden tahsis edildi.

Uygulama Tabanlı Dosya Kurtarma

- Uygulama tabanlı dosya kurtarma, bilinen bir dosya türlerinin başlangıcına ve bitişine karşılık gelen imzalar için veri yığınının arandığı bir süreçtir.
- Bu analiz sürecinin sonucu, imzalardan birini içeren bir dosya koleksiyonudur.
- Bu genellikle bir dosya sisteminin ayrılmamış alanı üzerinde yapılır ve araştırmacıya, kendilerine işaret eden herhangi bir meta veri yapısı olmayan dosyaları kurtarmasını sağlar.
- Örneğin, bir JPEG resmi standart üstbilgi ve altbilgi değerlerine sahiptir. Bir araştırmacı, silinmiş resimleri kurtarmak isteyebilir; bu nedenle, ayrılmamış alanı ayıklar ve JPEG üstbilgisine bakan bir carving aracı çalıştırır ve veriyi üstbilgi ve altbilgi arasında ayıklar.

Uygulama Tabanlı Dosya Kurtarma

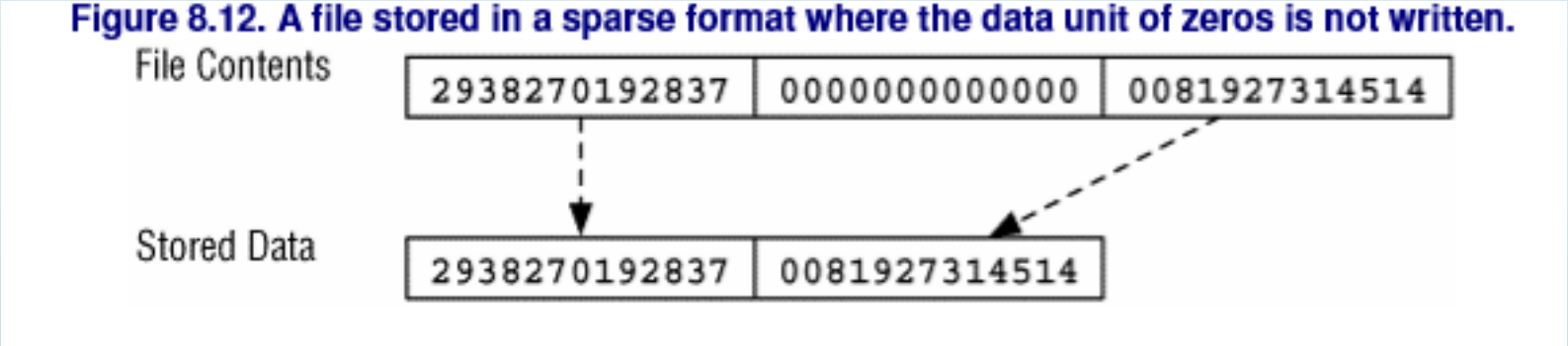
- Adli bilişim araçlarının birçoğu imza tabanlı dosya kurtarma işlemi gerçekleştirir.
- Ayrıca foremost isimli araç her imza için bir girişi olan bir konfigürasyon dosyasının içeriğine dayanarak bir ham dosya sistemi veya disk imgesini analiz eder.
- İmza, bilinen başlık değerini, dosyanın maksimum boyutunu, üstbilgi değerinin büyük / küçük harfe duyarlı olup olmadığını, dosya türünün tipik uzantısını ve isteğe bağlı altbilgi değerini içerir.

Sıkıştırılmış Dosyalar

- Bazı dosya sistemleri, verilerin diskte daha az veri birimi kullanacakları sıkıştırılmış bir biçimde depolanmasına izin verir.
- Dosyalar için sıkıştırma en az üç düzeyde gerçekleşebilir.
- En üst düzey bir dosya biçimi içindeki veriler sıkıştırıldığında gerçekleşir. Bir JPEG dosyası, resim bilgilerini depolayan verilerin sıkıştırıldığı, ancak dosya üstbilgisinin olmadığı bir örnektir.
- Sonraki düzey, harici bir program tüm bir dosyayı sıkıştırdığı ve yeni bir dosya oluşturduğu zamandır.
- Sıkıştırmanın son ve en düşük seviyesi, dosya sistemi veriyi sıkıştırdığı zamandır.
- Bu durumda, dosyayı yazan uygulama, dosyanın sıkıştırıldığını bilmez.

Dosya Sistemi Sıkıştırma Yöntemleri

- Dosya sistemleri tarafından kullanılan iki temel sıkıştırma tekniği vardır.
- **İlk teknik**, dosyalarda kullanılan aynı sıkıştırma tekniklerini kullanmak ve bunları dosyanın veri birimlerine uygulamaktır.
- **İkinci teknik** ise, tüm sıfırlarla doldurulan fiziksel bir veri birimini tahsis etmemektir. Sıfırlarla dolu veri birimlerini atlayan dosyaları, seyrek dosyalar olarak adlandırır.



Sıkıştırılmış dosyalar, soruşturma aracı sıkıştırma algoritmasını desteklememesi durumunda soruşturma için zorluk çıkarabilir. Ayrıca, sıkıştırılmamış veriler yerine sıkıştırılmış verileri incelemek için bazı anahtar kelime arama ve dosya kurtarma biçimleri etkisizdir

Şifreli Dosyalar

- Dosya içeriği yetkisiz erişime karşı korumak için şifrelenmiş bir biçimde saklanabilir.
- Şifreleme, dosyayı oluşturan uygulama tarafından, şifrelenmemiş bir dosyayı okuyan ve şifrelenmiş bir dosya oluşturan harici bir uygulama tarafından uygulanabilir veya dosyanın oluşturulduğu sırada OS tarafından uygulanabilir.
- Bir dosya diske yazılmadan önce işletim sistemi dosyayı şifreler ve şifre metnini veri birimlerine kaydeder.
- Dosya adı ve son erişim zamanı gibi içerik olmayan veriler genellikle şifrelenmez.
- Dosya içeriğini şifrelemek için bir başka yöntem ise tüm birimi şifrelemektir.
- Bu durumda, yalnızca içerik değil dosya sistemindeki tüm veriler şifrelenir. Genel olarak, OS içeren birim tamamen şifrelenmez.

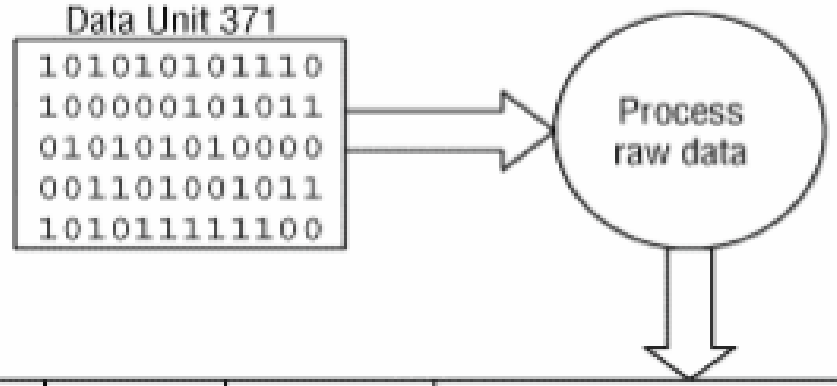
Şifreli Dosya Analizi

- Şifreli veriler, bir araştırmacıya meydan okumakla yükümlüdür, çünkü şifreleme anahtarını veya şifreyi bilmiyorsa dosyaların çoğuna erişilemez.
- Şifreleme tekniği bilinmiyorsa daha da kötüdür. Her anahtar veya şifre kombinasyonunu kaba kuvvet saldırısı olarak tahmin etmek için bazı araçlar bulunur, ancak algoritma bilinmiyorsa bunlar yararlı değildir.
- Yalnızca seçilen dosyalar ve dizinler şifrelenirse, şifrelenmemiş verilerin kopyaları geçici dosyalarda veya ayrılmamış alanda bulunabilir

Metadata Arama

- TSK'da **istat** aracı, metadata veri yapısından değerleri gösterir

Figure 8.13. Process for viewing the contents of a metadata entry.



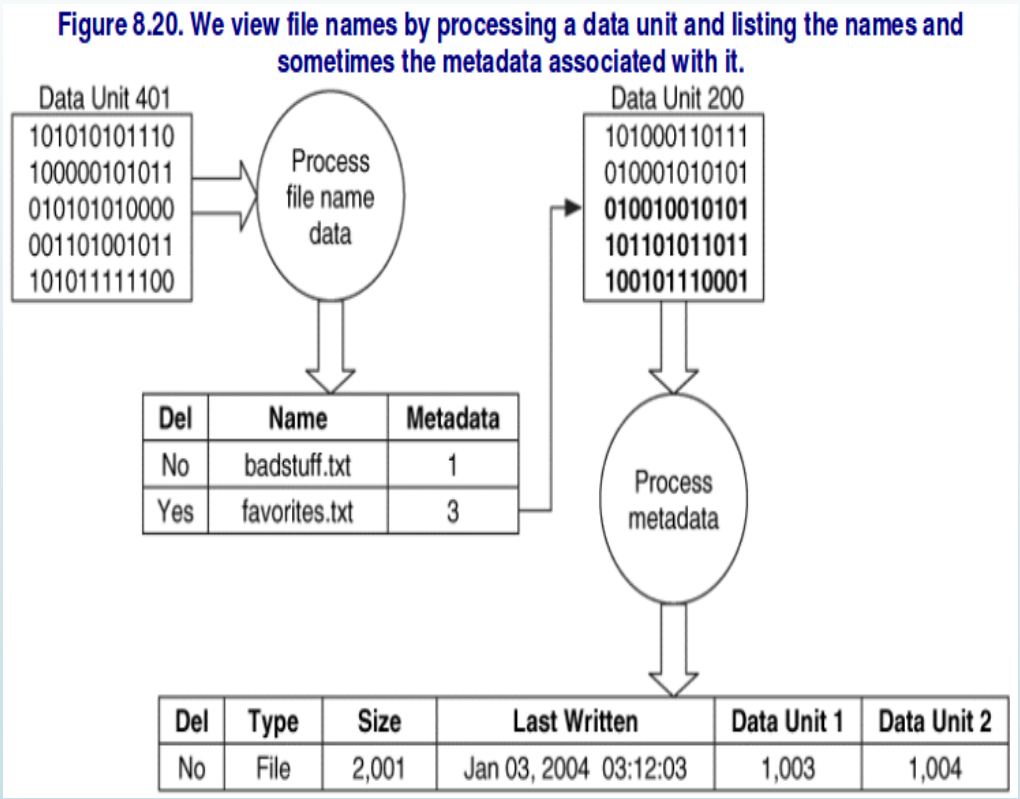
Del	Type	Size	Last Written	Data Unit 1	Data Unit 2
Yes	File	1,389	Jan 03, 2004 03:12:03	300,140	0
No	Dir	315,66 8	Jan 03, 2004 03:12:15	300,147	300,148

Dosya Adı Listeleme

- Dosya adı kategorisinin amacı, dosyalara ad atamaktır.
- Bir dosya kabul edildikten sonra, daha fazla bilgi edinmek için meta veri adresini kullanabiliriz.
- Bu teknikte yapılan değişiklikler, aynı bildirilen türdeki dosyaların gruplandırılabilmesi için dosyaları, dosya uzantılarına dayalı olarak sıralayacaktır.
- Çoğu dosya sistemi, silinmiş bir dosyanın dosya adını temizlemez; bu nedenle, silinen dosya adları listede de gösterilebilir. Bazı durumlarda, bir dosya silindiğinde meta veri adresi silinir ve daha fazla bilgi edinilemez.

Dosya Adı Listeleme

- Öncelikle dosya sisteminin kök dizinini bulunur.
- Kök dizin bilgisi metadata kaydında saklanır ve kayıt dizinin tahsis ettiği veri birimlerinin bulunması gerekir.
- Dizin içeriği bulunduktan sonra işlenir ve bir dosya ile ilgili metadata adres listesi elde edilir.
- TSK'da, **fls** aracı tahsis edilen ve silinen dosya adlarını listeler.



Dosya Sistemi Günlükleri (File System Journals)

- Bir bilgisayarın durması ve çökmesi nadir değildir.
- İşletim sistemi diske veri yazarken kilitlenme meydana geldiğinde veya istemsiz kapandığında dosya sistemi tutarsız bir durumda kalabilir.
- Tutarsızlıkları bulmak için bir OS, dosya sistemini tarayan ve eksik işaretçiler ve bozulma işaretleri arayan bir programı çalıştırır.
- Bu, büyük dosya sistemleri için çok uzun zaman alabilir.
- Tarama programının işini kolaylaştırmak için bazı dosya sistemleri bir günlük tutar. Bu nedenle, herhangi bir meta veri değişikliği dosya sisteminde yapıldığından, ortaya çıkacak değişiklikleri açıklayan bir günlük girişi vardır.
- Sistem çökerse, tarama programı günlüğü okur ve tamamlanmamış girdileri bulur.
- Program daha sonra değişiklikleri tamamlar veya orijinal durumuna geri döndürür.

Veri Türüne Göre Analiz Teknikleri

Table 8.2. The search methods and locations, depending on what evidence you are looking for.

Analysis Needs	Data Category	Search Technique
A file based on its name, extension, or directory	File name	File name search or listing directory contents
An allocated or unallocated file based on its time values	File name and metadata	Metadata attribute searching
An allocated file based on a value in its content	File name (using metadata and content)	Logical file search
An allocated file based on its SHA-1 hash value	File name (using metadata and content	Logical file search with hashes
An allocated file or an unallocated data unit based on a value in its content	File name (using metadata and content)	Logical file search with metadata-based file recovery and logical file system search
An unallocated file based on its application type	Application and content	Application-based file recovery of unallocated data units
Unallocated data based on its content (and not its application type)	Content	Logical file system search