



Karadeniz Teknik Üniversitesi

Mühendislik Fakültesi
Harita Mühendisliği Bölümü

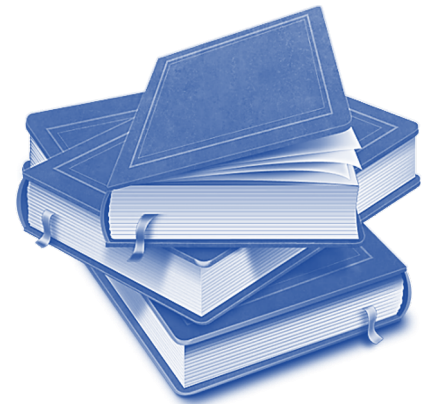
Dosya yapıları ve Erişim yöntemleri

Kartografya ABD



Ders İerięi

- Dosya yapıları ve Eriřim yöntemleri
 - Sıralı erişimli dosya yapıları (Sequential Access File-SAF)
 - Rastgele erişimli dosya yapıları (Random Access File)
 - Mantıksal dosyalar
 - İndeksli dosyalar
 - Hashed dosyalar



- **Dosya**, fiziksel depolama ortamlarında verilerin saklandığı mantıksal yapılardır.
- **Veri yapısı**, hangi verilerin depolandığını ve onların birbirleriyle nasıl ilişkili olduğunu belirler. Veri yapısını seçmek depolama gereksinimini, veri tabanında gerçekleştirilecek işlemlerin performansını ve veri tabanının bilgi içeriğini etkiler.
- **Dosya yapısı**: Bilgisayarda dosyaları düzenleme veya gruplandırma yoludur. Kalıcı veri depolama ortamlarında veri yapısının bir uygulamasıdır. Bir işi birden fazla icra edebiliriz. Örneğin 2D-3D ifade etmek gibi. Bunlara dosya yapısı denir.
- **Fark:**
 - Veri yapıları ana bellekte (main memory) verinin etkili bir şekilde depolanması ve işlenmesiyle ilgilidir.
 - Dosya yapıları ikincil depolama aygıtlarında (secondary storage) verinin depolanması organizasyonu ve işlenmesi ile ilgilidir.

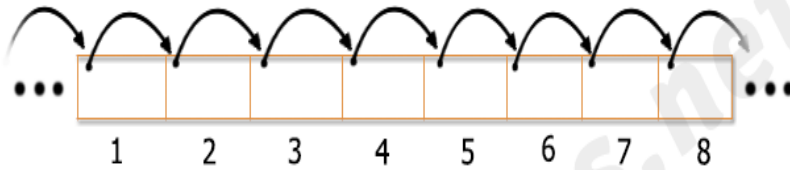
Dosyalar erişim yöntemine göre ikiye ayrılır:

1. Sıralı erişimli dosya yapıları

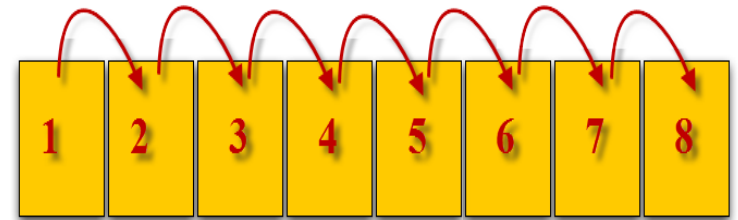
2. Rastgele erişimli dosya yapıları (Doğrudan erişimli dosyalar)

Dosya yapıları **alanlardan ve kayıtlardan** oluşur. Alanlar **sabit uzunluklu** (fixed length) ve **değişken uzunluklu** (variable length) olabilir.

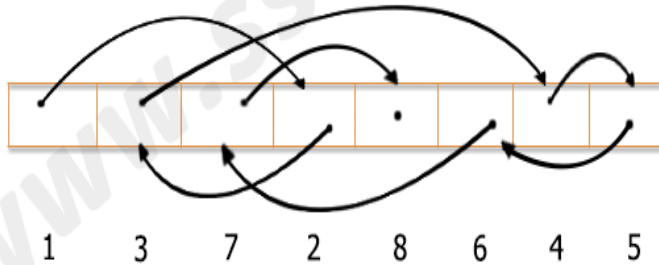
Sıralı Erişim (Sequential)



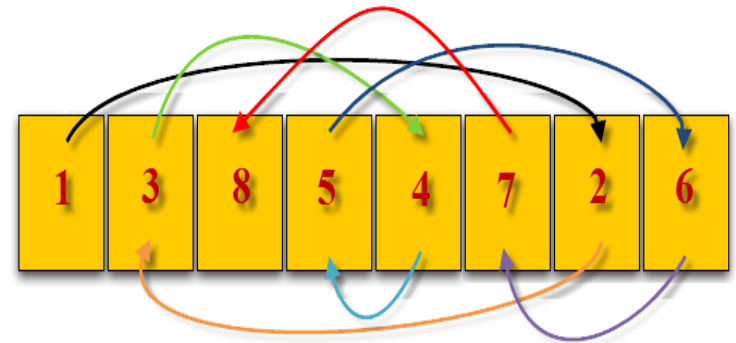
Sıralı erişim (Sequential)



Rastgele Erişim (Random)

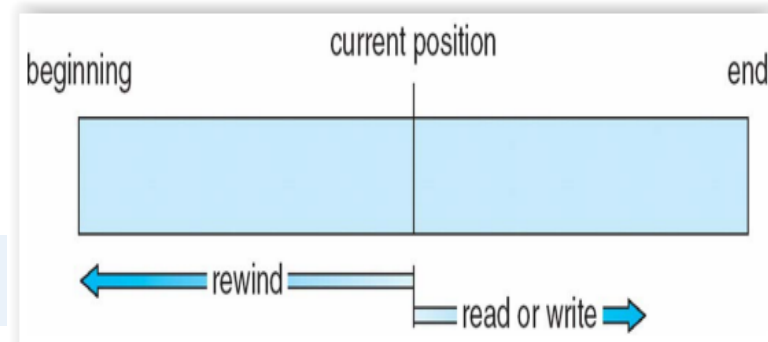


Rastlantısal erişim (Random)



Sıralı Erişimli Dosya (Sequential Access File-SAF)

- Bu tür erişimde bilgilere daha önce **belirlenen bir sıra** izlenerek **birbiri ardı sıra** ulaşılır. Sıralı erişim yönteminin en büyük dezavantajı, **herhangi bir bilgiye doğrudan ulaşmanın mümkün olmamasıdır**. Örneğin 500 kayıt bulunan bir dosyada 453. sırada yer alan bir kayda ulaşmak için bu kayıttan önce yer alan, 452 adet kaydın okunarak atlanması zorunludur. Bu nedenle sıralı erişim yöntemi genellikle aynı yapıdaki işlerin birleştirilerek kümeler halinde yapıldığı (batch processing) uygulamalarda kullanılır.
- En basit erişim yöntemidir.
- Bütün byte'lar ve kayıtlar baştan okunur. Dosyadaki bilgiler birbiri ardına kayıt edilir, sırayla işlenir.
- Atlama olmaz, geri sarma ve yedekleme olur.
- read next
- write next
- reset



- Dijital harita veri tabanı için basit bir veri yapısı, dosyada noktaları sıralı bir şekilde düzenlemektir. Bir haritada çizgi katmanını alalım ve onun bir dosyada depolamak istediğimizi varsayalım. Yol Haritası (Diğer bir örnek sınırlarının koordinatlarıyla birlikte poligonları depolamak olabilir.) Dosyalar kayıtlardan oluşur.

Çizgi 1
Kayıt 1

Çizgi 2
Kayıt 2

Çizgi 3
Kayıt 3

- Sıralı erişimli dosya yapısında bu çizgileri depolamak için bir alternatif:

çizgi 1

çizgi 2

çizgi 3

X,Y,X,Y,X,Y,.....,X,Y,**EOL**,
X,Y,X,Y,X,Y,.....,**EOL**, **EOF**

Sıralı erişimli dosyada kayıtlar arka arkaya fiziksel olarak devam eder.

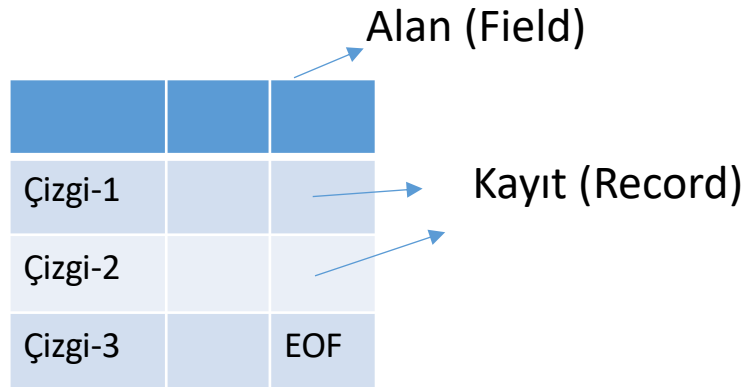
- **EOL** Her bir çizgi özel bir işaretçi ile sonlandırılır. (EOL « işaretçi » for End-of-Line) (örneğin -1 olabilir bütün koordinatlar pozitif ise)
- **EOF** dosya sonunu tanımlamak için çizgi sonunun hemen sonrasında yer alır. Genellikle bilgisayar sistemleri tarafından atanır. (EOF « işaretçi » for End-of-File)
- Dosyada 10. çizgiyi bulmak için program ilk 9 çizgiyi taramalıdır. Böyle veri yapıları sıralı arama olarak isimlendirilir.
- *.NCZ dosyası hangisidir? Rastgele veya sıralı?
- Kayıtların sonlarını anlaması için bir sembol, bir çizgi yerleştirilir.

- **Diğer bir alternatif** EOL işaretçileri yerine kayıtları takip etmek için her bir çizgi başlangıcında koordinatların numarasının sayısı kayıtın başına yerleştirilebilir. (**COUNT**)

COUNT, Y,X,Y,X,Y,.....X,Y, **COUNT**,X,Y,X,Y,X,Y,.....X ,Y ,**EOF**

- Bu dosyalar teyp kasetleri gibi algılanabilir. İstenilen şarkıya gelmek için öncelikle ileri veya geri sarmak gerekir. Aynı durum örneğin 10. çizgiyi almak için söz konusudur. 10. çizgiye ulaşmak için ilk 9 çizgi okunmalıdır.
- Hangi alternatifle çalışmak daha kolaydır?
- Böyle bir dosya yapısı sabit uzunluklu «fixed length» veya değişken uzunluklu «variable length» mudur?

1

Sabit uzunluklu dosya (Fixed length file)

Her bir kayıt aynı uzunluktadır.

2

Değişken uzunluklu dosya (Variable length file)

ID	Koord.
Çizgi 1
Çizgi 2
Çizgi 3

Her bir kayıt farklı uzunluktadır.

- GPS verilerinde sabit uzunluklu dosya ile çalışılmaz. Her çizgi farklı uzunlukta olduğundan değişken uzunluklu dosya ile çalışmak daha uygundur.

Sıralı Erişimli Dosya Yapısının Avantajları :

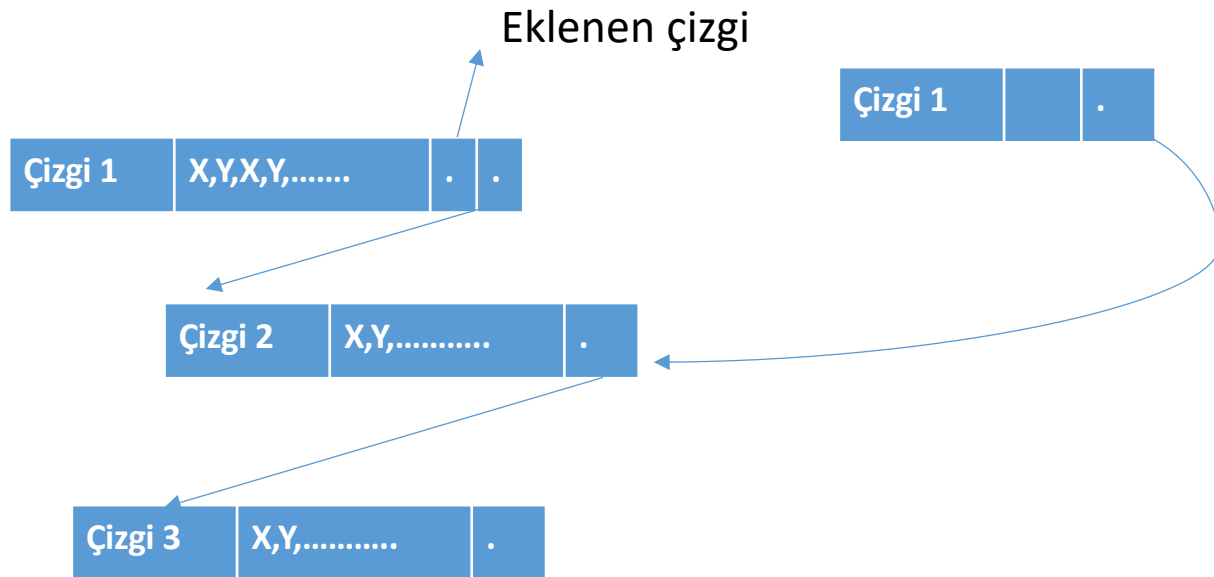
- Yeni bir çizgi (çizgi-4) eklemek kolaydır. Basitçe dosya sonuna eklenir. Örneğin yeni bir yol parçası mevcut yola eklenir.
- Bir çizgiyi **mantıksal olarak** silmek kolaydır. Örneğin bir yol parçası mevcut yoldan kaldırılır. Bu COUNT -1 yapılarak gerçekleştirilir. Çizgiyi fiziksel olarak silmek için veri kaydırılmalıdır.
- Çizginin şeklini değiştirmek kolaydır.
- Veri yapısı basittir, daha az yer kaplar ve anlamak daha kolaydır.

Sıralı Erişimli Dosya Yapısının Dezavantajları :

- **Veriye erişim tamamen sıralıdır.** Herhangi bir kayda erişmek için dosyanın başından itibaren o kayda kadar tüm kayıtların sırasıyla okunması gerekir. (Verilen bir noktaya en yakın çizgi nedir? Sorgusunu cevaplamak için dosyadaki bütün noktalar incelenmelidir.)
- **Bir çizgiye yeni bir nokta eklemek için sağ taraftaki bütün noktalar yeni noktaya yer açmak için hareket ettirilmelidir.** Yeni bir nokta eklemek çok zordur. Bu, verinin fiziksel olarak kaydırılmasını gerektirir.
- **Bir noktayı fiziksel olarak silmek için aynı problemle karşılaşılır.** bu kez soldaki kaydırılır.
- Sonuç olarak, sıralı bir dosya interaktif (etkileşimli) işlemler için uygun değildir fakat arşivde **veri depolamak için, aktif dosyaları yedeklemek için ve veri değişimi için** idealdir.

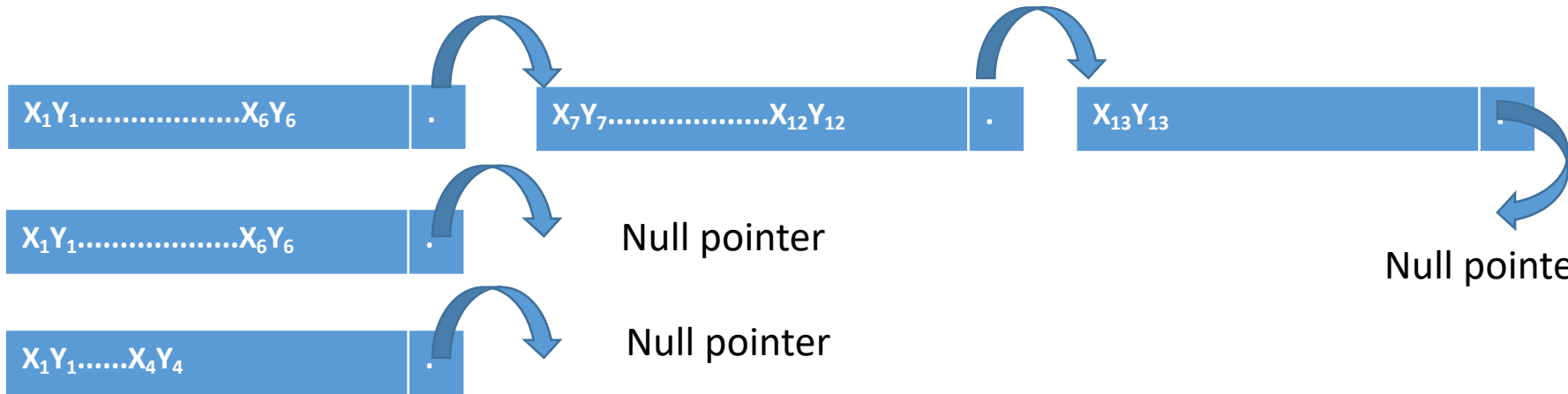
Rastgele (Doğrudan) Erişimli Dosya (Random Access File)

- Bu erişim yönteminde, ulaşmak istenen herhangi bir bilgiye, diğer kayıtlardan bağımsız olarak **doğrudan ulaşmak mümkündür**. Ancak, bunun için her kaydı diğer kayıtlardan ayıran özel bir tanımlayıcıya gerek duyulur. Bu özel tanımlayıcı **"anahtar" (key)** olarak adlandırılır. Böylelikle anahtar bilindiği zaman, diğer kayıtlar incelenmeksizin ve kayıtların kaydediliş sırasıyla ilgilenilmeksizin istenilen kayda doğrudan ulaşmak mümkündür.
- Dosya mantıksal kayıtlar olarak düzenlenir. Mantıksal kayıtların fiziksel olarak ardışık olmaları gerekmez. Çünkü her biri bağımsız olarak "adreslenebilir". **Mantıksal kayıtlar fiziksel olarak disk üzerinde farklı bir yeredir.**



Fixed length file
Kayıt boyutunu artıramıyoruz.

- Dosyadaki kayıtların sabit uzunluklu olması durumunda;
- Ortalama bir kayıt uzunluğu seçilir. (6-6)



- Rastgele erişimli dosya yapılarının çok sayıda farklı türleri vardır.
- Mantıksal dosyalar (Dosyadaki kayıtlar işaretçilerle birbirine bağlıdır.
- İndeksli dosyalar
- Hashed dosyalar
- ...

- **Avantajları**

- Yeni bir nokta eklemek daha kolay çünkü sadece kayıt içindeki verinin hareket ettirilmesini gerektirir. SAF taki gibi tüm dosya içinde değil sadece kayıt kaydırılıyor.
- Arama, ikili arama gibi rastgele erişim mekanizmasıyla SAF tan daha hızlıdır.

- **Dezavantajları**

- Yeni bir çizgi eklemek SAF tan daha zordur. Dosyanın nerede olduğu bulunmalıdır ve bulunan yere çizgi eklenmelidir.
- Depolama alanında boşluklar oluşur. (Bu açıdan SAF daha iyidir çünkü noktalar ve boşluklar yer kaplıyor)

Bazı uygulamalar açısından;

- **Ekrana haritayı çiz** (SAF bir parça dosya var ama RAF ta parçaları bir araya getirilecektir. SAF daha iyidir.)
- **Haritadaki çizgilerin uzunluğunu hesapla.** (SAF daha iyidir. Parçaların birleştirilmesi söz konusu uzunluk işleminde RAF)
- Genel olarak çizgilerin tamamını gerektiren uygulamalarda (çizgi parçalarını bir araya getirme) SAF daha iyidir.

POINTER (gösterge, işaretçi)

- POINTER: Depolama adresi gibi disk, mekanizma
- C diliyle tanıtılan bir yapıdır. O bir depolama adresidir. POINTER uygulamaları çok hızlıdır ancak programcılar açısından çok karışıktır.
- C dilinden örnek ip x adresini taşıyor. Y'ye taşıyarak bunu 1 yaptı.

```
Int x=1, y=2, Z[10]
```

```
Int * ip; /* ip bir tamsayı işaretçisidir*/açıklama satırı
```

```
ip=& x; /* ip ye x in adresini ver*/
```

```
y=*ip; /* y=1 */
```

```
*ip=0; /* x=0 */
```

```
ip=&Z(0); /* ip yeni noktaları Z(0) elemanın adresini ver*/
```

Bellek (Ram)

&: adres atama
Byte olarak ip adresi olarak atıyor



İndeksli dosya

Parsel Dosyası (Veri)

İndeks Dosyası

ID	Pointer
1200	.
1250	.
1300	
1350	

ID	Malik	Değer(m ²)
1200
	10 000
		
1249		

BLOK
512 B on
1024 Byte

ID	Malik	Değer(m ²)
1250

1299		

Blok içerisinde sıralı arama

ID	Malik	Değer(m ²)
1300			
1349			

- 1299 nolu parselin değerini göster.
- Bu aramanın adımları
 1. Uygun blokları bul
 2. Sıralı arama ile blokları ara (Baştan başlayarak tüm kayıtları arar ve 1299 u bulur.
- Indeksler: **Primary index**, secondary index, clustering index(sınıflandırma indeksi)
Primary key (indexing in based on)

Hashed dosyalar

Hashed file: En hızlı rastgele erişim yöntemidir. Hem veri girişi hem arama «hash fonksiyonu» ile yapılır. $F(T)=T \bmod(N)$ N: kayıt sayısı T:veri
eg: {3,7,8,11,15,12}

Kayıt	Veri
0	8,12 .
1	
2	
3	3,7,11,15 .

N=4 sayının 4 e bölümünden kalan kayıt numarası olacaktır.

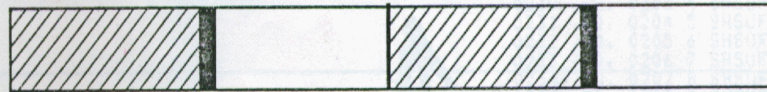
Bu dizide "9" var mı?
 $9 \equiv 1 \bmod 4$ 1 e bakıyoruz.
1 nolu kayıt içinde "9" yoktur.

Hashed file sabit uzunluklu çalışır. Fonksiyonu optimum dağıtır.

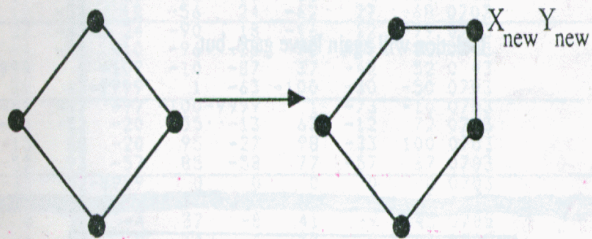


With the simple data structure, new lines are added to the end of the file.

← Shift left to close gap.



However, deletion will leave gaps in the file. A gap can be re-used by a shorter line but not by a longer one. Of course, gaps can be closed by shifting data.



→ Shift right to make room



X_{new} Y_{new}

Insertion, on the other hand, must involve data shift.

A Sequential Structure

