Dosyalarda Farklı Yaklaşımlar

Giriş

- Şimdiye kadar öğrendiğimiz temel dosyalama komutlarıyla (fopen,flclose, fputs vb..) dosya oluşturabilmekte, kayıt ekleyebilmekte ve her bir kaydın içeriğini değiştirebilemekteyiz.

- Bir kaydın silinmesi gerektiği durumlarda ne olacak?

- Kayıt silinmesi için tanımlanmış bir fonksiyon bulunmamaktadır. Bunun için bazı stratejilerin izlemesi gerekmektedir.

Kayıt Silme Yaklaşımları

Bir kayıt nasıl silinir?

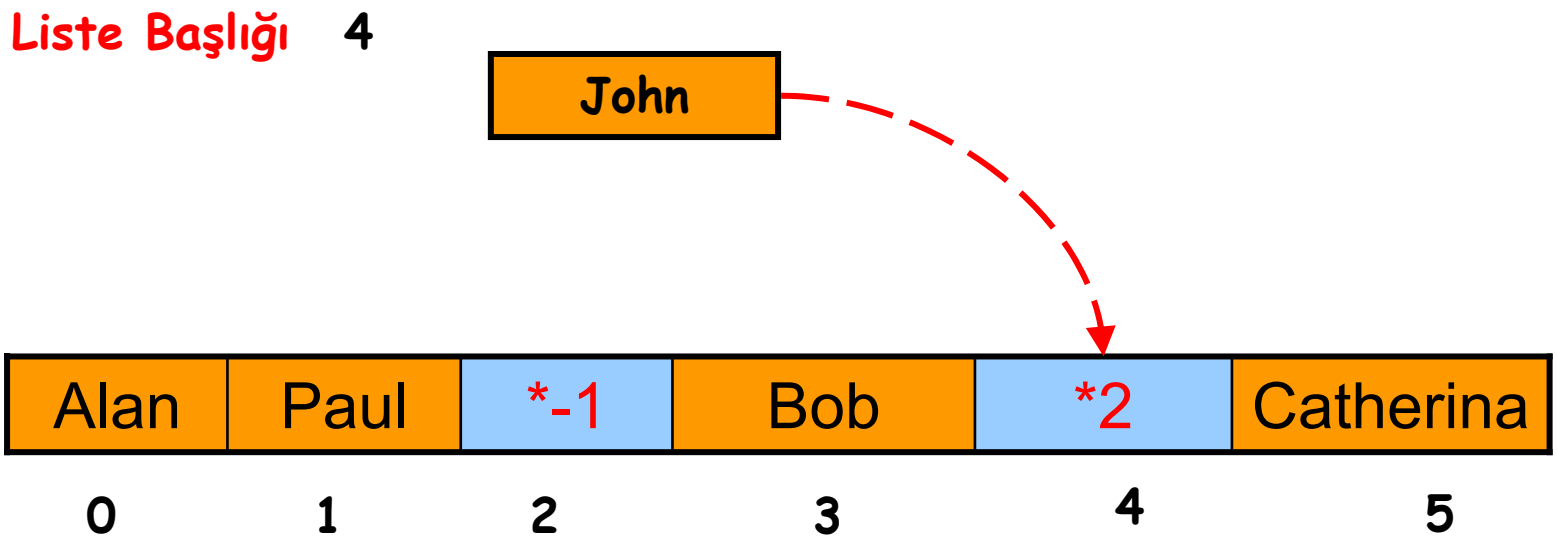
1. Kayıt Silinmesi ve Yeniden Dosya Organizasyonu

- Kayıt silme işlemi bir kaydın "**Silinmiş**" olarak işaretlenmesiyle yapılır.
- Boş alan serbest bırakılmadığından program içerisinde bir kaydın silinmiş olarak işaretlenip işaretlenmediğinin kontrolünün yapılması gerekir.
- Çok fazla kayıt silindiğinde özel bir program (fonksiyon) dosyayı sıkıştırır. Diğer bir deyişle silinmişleri atar, dosyayı yeniden organize eder.

Kayıt silme işleminde sonra boş kalan yerler daha sonra yeni kayıt eklemelerinde nasıl kullanılır?

2. Sabit Uzunluklu Kayıtların Silinmesi ve Boş Alanların Dinamik Olarak Kullanılması

- "Erişim Listesi" kullanılarak, boş alanlardan bir bağlı liste oluşturulur.
- Dosyanın başlangıcında bulunan bir başlık kaydı "Erişim Listesi" nin başlangıcını tutar.
- Bir kayıt silindiğinde "silinmiş" olarak işaretlenir ve "Erişim Listesi" ne ilave edilir. Bu işlem mantıksal olarak yapılır.



Yeni bir kayıt eklendiğinde, erişim listesinde bulunan ilk boş noktaya gider.

Not: -1 null pointer olarak gösterilmiştir.

3. Değişken Uzunluklu Kayıtların Silinmesi

- “Erişim Listesi” silinecek kaydın uzunluğunu da saklaması gerekir.

- Yeni kayıtların eklenmesi için “Erişim Listesi”nde yeterince genişlikte yer bulunmalıdır.

Yeni Kayıt Ekleme Yaklaşımları

Yeni kayıt ekleneceği zaman, “Erişim Listesi”nden bir kayıt seçmek için pek çok yaklaşım bulunmaktadır.

1. First Fit Yaklaşımı

- “Erişim Listesi” boyuta göre sıralanmaz.
- Yeni kaydı barındırabilecek büyüklükte olan ilk kayıt seçilir.

Örn:

Erişim Listesi: boyut=10, boyut=50, boyut=22, boyut=60 şeklinde ve eklenecek kaydın boyutu ise 20 byte olsun.

Bu durumda “Erişim Listesi” ndeki hangi kayıt eklenecek yeni kayıt için kullanılır?

2. Best Fit Yaklaşımı

- "Erişim Listesi" boyuta göre sıralanır.

- Yeni kaydı barındırabilecek büyüklükte olan en küçük kayıt seçilir.

Örn:

Erişim Listesi: boyut=10, boyut= 22, boyut=50, boyut=60 şeklinde ve eklenecek kaydın boyutu ise 20 byte olsun.

Bu durumda "Erişim Listesi" ndeki hangi kayıt eklenecek yeni kayıt için kullanılır?

3. Worst Fit Yaklaşımı

- "Erişim Listesi" boyuta azalan şekilde sıralanır.

- Yeni eklenecek kayıt için en büyük kayıt seçilir. Kullanılmayan kısımlar tekrar "Erişim Listesi" ne eklenir.

Örn:

Erişim Listesi: boyut=60, boyut= 50, boyut=22, boyut=10 şeklinde ve eklenecek kaydın boyutu ise 20 byte olsun.

Bu durumda "Erişim Listesi" ndeki hangi kayıt eklenecek yeni kayıt için kullanılır?

Binary Arama, Anahtar Sıralama ve Indexleme

Binary Search

• Anahtar değerlerine göre aranması gereken sabit uzunlukta kayıtlar olsun.

• Eğer, anahtar değeri ile tanımlanmış olan kaydın göreli kayıt numarasını (Relative Record Number-RRN) biliyorsak istenilen kayda fseek() fonksiyonunu kullanraka direkt olarak ilgili kayda erişebiliriz.

• Gerçekte böyle bir bilgiye sahip değilizdir ve bu anahtar değerini barındıran kayıt için arama yapmamız gerekir.

• Eğer, dosya anahtar değerine göre sıralanmamışsa, istenilen kayda ulaşmak için olası tüm konumlara teker teker bakmak gerekir.

• Bir diğer alternatif ise dosyayı anahtar değerlerine göre sıralamak ve sıralanan değerler üzerinde binary arama yapmaktır.

Binary Search

```
        proc binary_search

             /*  The n records of the file are ordered in in-
                 creasing order of the keys.   */

1            LOWER := 1
2            UPPER := n
3            while LOWER ≤ UPPER do
4                 MIDDLE := ⌊ (LOWER + UPPER) / 2 ⌋
5                 if key[sought] = key[MIDDLE]
6                      then terminate successfully.
7                      else if key[sought] > key[MIDDLE]
8                           then LOWER := MIDDLE + 1
9                           else UPPER := MIDDLE − 1
10           end
11           terminate unsuccessfully.
        end binary_search
```

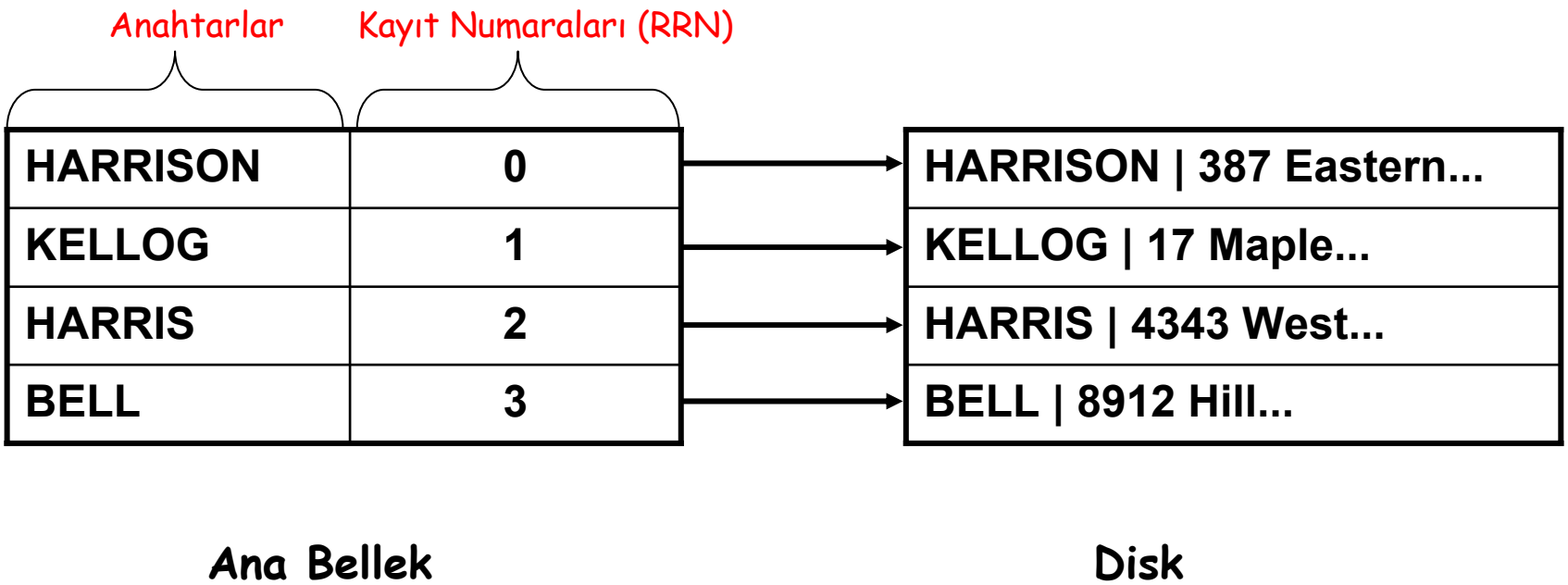
Anahtar Sıralama

- Bir dosyanın sıralanması gerekmektedir, fakat dosya ana belleğe sığamayacak kadar büyüktür.
- Dosyayı sıralamak için sadece anahtar değerlerine ihtiyaç vardır.
- İhtiyacımız olan bu anahtar değerleri ana belleğe sığmaktadır.

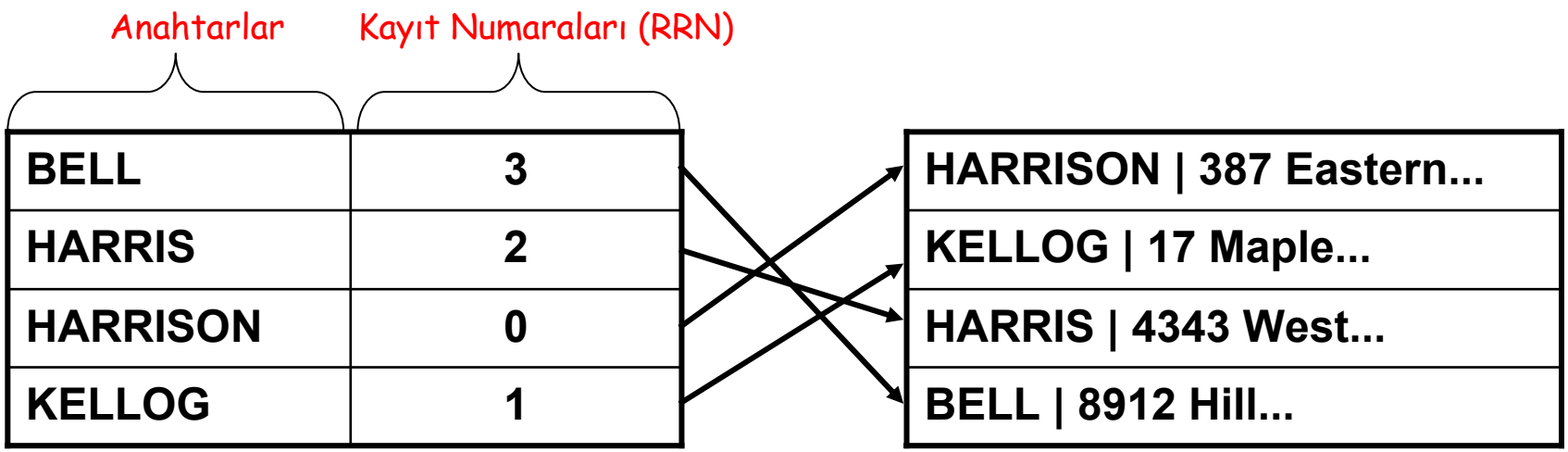
Çözüm:

- Anahtarları ve bu anahtarlara karşılık gelen göreli kayıt numaralarını (RRN) belleğe getirilir.
- Anahtarlar dahili olarak sıralanırlar.
- Sıralandıktan sonra dosya sıralı bir biçimde yazılır.

Örnek


Anahtarlar
Kayıt Numaraları (RRN)
HARRISON
0
KELLOG
1
HARRIS
2
BELL
3
HARRISON | 387 Eastern...
KELLOG | 17 Maple...
HARRIS | 4343 West...
BELL | 8912 Hill...
Ana Bellek
Disk

Örnek (Devam)



Ana Bellekte anahtarlar üzerinde gerçekleştirilen sıralama

Disk üzerinde herhangi bir değişiklik yok

Örnek (Devam)

Anahtarlar	Kayıt Numaraları (RRN)
BELL	3
HARRIS	2
HARRISON	0
KELLOG	1

BELL | 8912 Hill...
HARRIS | 4343 West...
HARRISON | 387 Eastern...
KELLOG | 17 Maple...

Sıralanmış yeni dosya oluşturulur ve öncekinin üzerine yazılır.

Öneri

- Neden sıralı dosyayı geri yazıyoruz ki?
- Bunun yerine oluşturduğumuz indexi kullanalım.

Index Dosyası

BELL	**3**
HARRIS	**2**
HARRISON	**0**
KELLOG	**1**

Kayıtlar

HARRISON | 387 Eastern...
KELLOG | 17 Maple...
HARRIS | 4343 West...
BELL | 8912 Hill...

Dosya değişmeden eski halinde kalır.

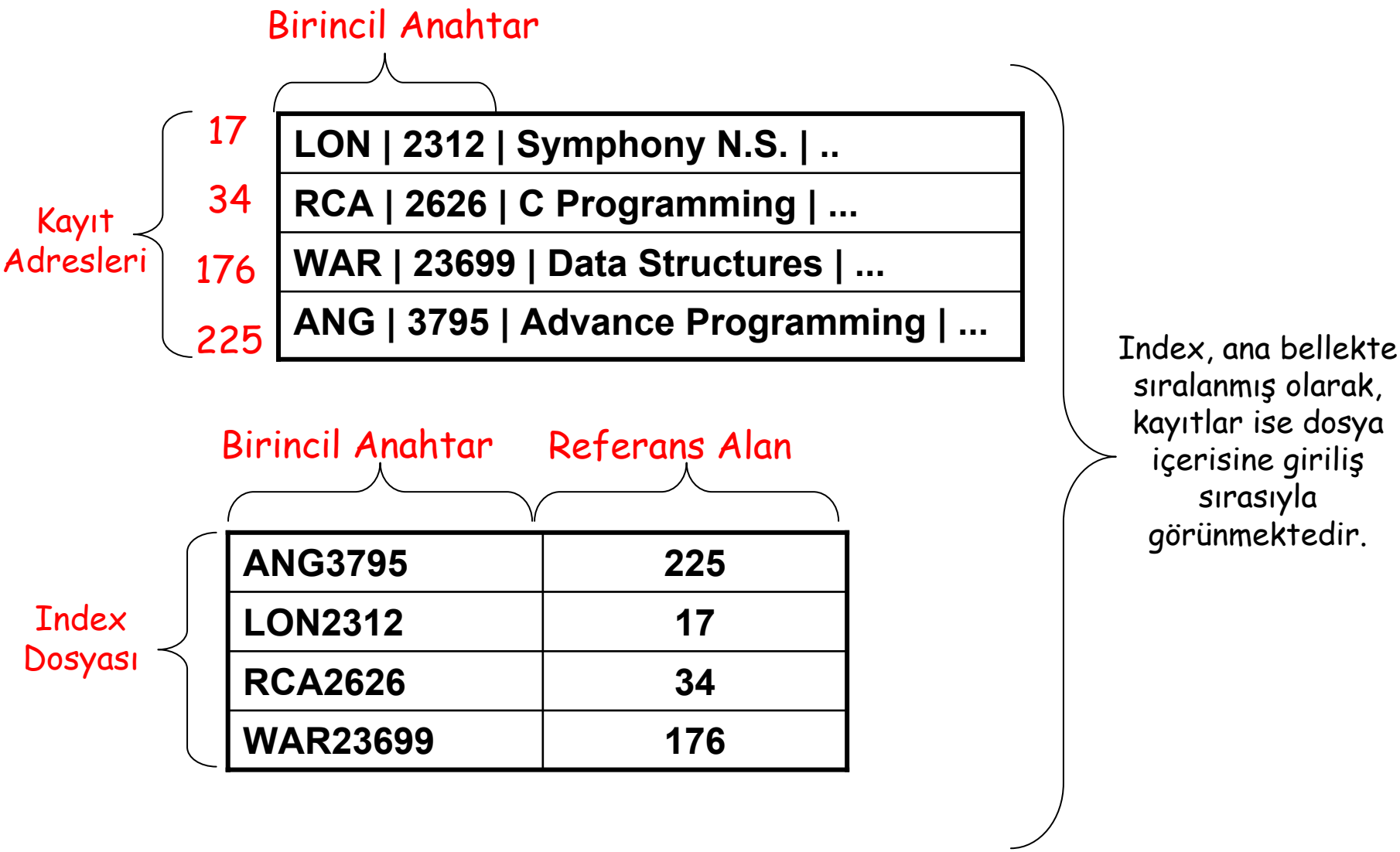
Buna Indexleme denir.

- Hatırlanacağı gibi silme işleminde dosya için mantıksal olarak oluşturulmuş bir “Erişim Listesi” kullanılmaktaydı.
- “Erişim Listesi” dosya üzerindeki kayıtların fiziksel bilgisini tutar.
- Eğer sıralam işleminde index dosyası kullanılırsa, “Erişim Listesi” ve dosyanın içindeki kayıtların konumları değişmeden aynen kalır.
- Bu sayede maliyet ve zamandan kazanılır.

Basit Indexleme

- Basit Indexler, dizileri kullanırlar. (Anahtarları ve alan referanslarını içeren bir tablodur) **Örn:** Bir kitabın indexi
- Genel olarak indexleme arama problemini ele alan başka bir yoldur.
- Indexleme, dosyayı yeniden organize etmden dosya üzerinde sıralama yapılmasını sağlar.
- Indexler, dosyalara çoklu erişim yolları (multiple access paths) sağlarlar. (**Örn:** Bir kütüphane katalog sisteminde, yazar adı, kitap adı ve başlık kullanılarak bir kitaba erişlebilir.
- Bir index değişken uzunluklu kayıtlara anahtarlı erişim sağlayabilir.

Giriş Sırasına Göre Düzenlenmiş Indexleme



Etiket ID'si Verilen Bir Kayıt Nasıl Aranır?

- İkili aram ile index içerisinden uygun referans bulunarak, onun gösterdiği yere konumlanılır.

Indexlenmiş Bir Dosya Üzerinde İşlemler

- İlk olarak boş bir index dosyası ve veri dosyası oluşturulur.

- Index dosyaysı kullanmadan önce ana belleğe yüklenir ve kullanıldıktan sonra yeniden yazılır.

  Hatırlanacağı gibi veri dosyası üzerinde:
  - Kayıt ekleme
  - Kayıt silme
  - Kayıtların güncellenmesi durumları söz konusudur.

- Veri dosyasındaki değişiklikler index dosyasına yansıtılarak güncellenir.

- Veri dosyası kapatıldığında, ana bellekteki index, index dosyasına yazılmış olmalıdır.

  Yeniden yazmanın gerçekleşmediği, elektrik kesintisi gibi, durumlarda ne olacaktır ?

  Çözüm: 2 faklı adım ile bu problem çözülebilr.

i) Index dosyasının başında bir durum belirteci bulundurulur.

ii) Eğer program indexin eski olduğunu tespi ederse, index dosyasını yeniden yapılandıran bir prosedürü çağırır.