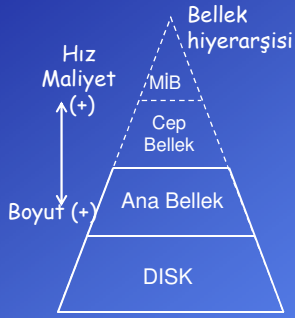


## 8 Bellek Yönetimi (Memory Management), Görüntü Bellek (Virtual Memory)

### Amaç:

- Kullanıcılara/programlara fiziksel belleğin (ana bellek) boyutundan bağımsız olarak büyük boyutta ve lineer (sürekli) bellek alanı sağlamak.  
Kullanıcılar/programlar, sistemde sadece kendilerine ait kesintisiz bir bellek alanı varmış gibi bir görüntü elde ederler.
- Bellek blokları üzerinde erişim denetimi (güvenlik/koruma) sağlamak.
- Çok kullanıcı / çok programlı sistemlerde ortak programların/verilerin paylaşılmasını sağlamak.



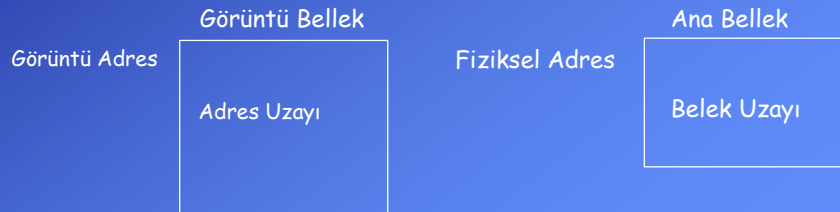
Cep bellek ile ana bellek arasındaki ilişki ana bellek ile disk arasında kurulur. (Yöresellik !)

Görüntü bellek sayesinde programlar, gerçekte var olan daha büyük bir bellek alanını kullanırlar.

Programlar ve veriler diskte tutulur; gerek duyulan komutlar ve veriler ana belleğe getirilir.

### Adresler ve bellekler:

- Programların yazılması ve adreslerin üretilmesi görüntü belleğe (*virtual bellek*) olur (sürekli ve büyük).
- Programların ürettiği adrese **görüntü** (sanal) (*virtual*) adres veya **mantıksal adres** (*logical address*) denir.  
Görüntü bellek kullanılan sistemlerde MİB görüntü adres üretir.
- Görüntü adreslerin kümesine (yani görüntü belleğin adres alanına) adres uzayı (*address space*) denir.
- Ana bellek adreslerine **fiziksel adres** (*physical address*) denir.
- Ana belleğin adres alanına bellek uzayı (*memory space*) denir.

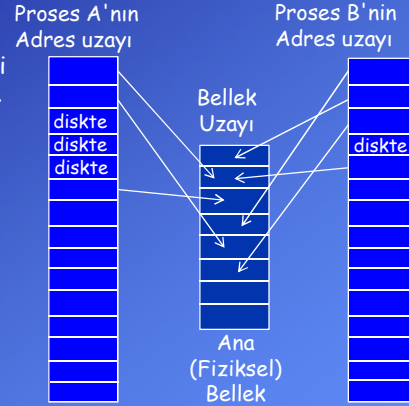


**Adresler ve bellekler :** (devamı)

Proseslerin adres uzaylarının değişik bölümleri fiziksel bellekte dağınık biçimde yer alırlar ve çoğunlukla peş peşe değildirler (Kullanılmayan veriler diskte).

İşletim sistemi görüntü adres uzayındaki bir verinin gerçek yerini belirleyebilmek için bir tablo tutar; veri diskte veya fiziksel (ana) bellekte olabilir (hangi adreste?).

Prosesler adres uzaylarının hangi bölümünün sistemde nerede (disk / bellek) olduğunu bilmezler.



Görüntü bellek yönetim sistemi büyük ve sürekli bir bellek sağladığından, çalışan her proses tüm sistem kendisine aitmiş gibi adresler üretir.

**Temel yararı:**

Programcıların ve derleyicinin işi kolaylaşır, çünkü programı oluştururken donanımın veya fiziksel bellek yapısının ayrıntılarını bilemeye gerek yoktur.

Programlar fiziksel belleğin yapısından bağımsız olarak derlenebilir.

**8.1 Sayfalı Dönüşüm (Paged Mapping):**

Coğrafi yöresellikten yararlanmak ve veri aktarımını DMA (IOP) ile yapabilmek için görüntü bellek (disk) ile ana bellek arasındaki veri aktarımı sayfalar ile yapılır.

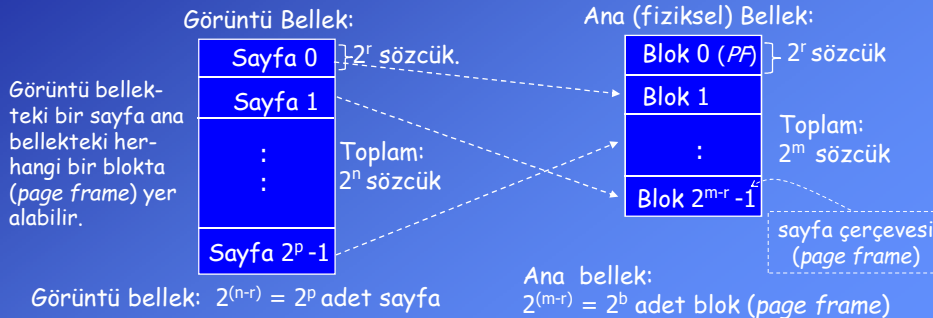
Görüntü bellek (adres uzayı) sabit boyuttaki **sayfalara** (sayfa) bölünür.

Ana belleğin sayfalarla aynı büyüklükte bloklardan (sayfa çerçevesi - page frame PF) oluştuğu düşünülür.

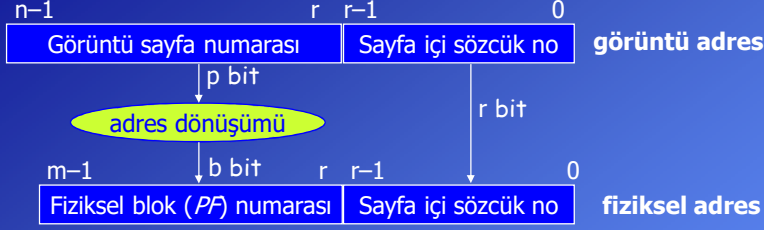
Sayfa boyu:  $2^r$

Adres uzayı:  $2^n$  Sayfa sayısı:  $2^p$ ,  $p = n - r$

Bellek uzayı:  $2^m$  Blok sayısı:  $2^b$ ,  $b = m - r$



MİB (bir program) bir görüntü adres ürettiğinde, bellek yönetim birimi (MMU) bu adresi fiziksel adrese dönüştürerek başvuru sayfanın ana bellekteki yerini belirler.



Hangi sayfanın hangi blokta yer aldığı bir **sayfa tablosunda** (page table -PT) tutulur. Sayfa tablosu fiziksel bellekte oluşturulur ve her prosesin kendi tablosu vardır. Eğer başvuru sayfa o anda ana bellekte yoksa bir sayfa hatası (page fault) oluşur. İşletim sistemi o prosesi bloke eder ve DMAC ilgili sayfayı görüntü bellekten ana belleğe getirecek şekilde koşullandırır.

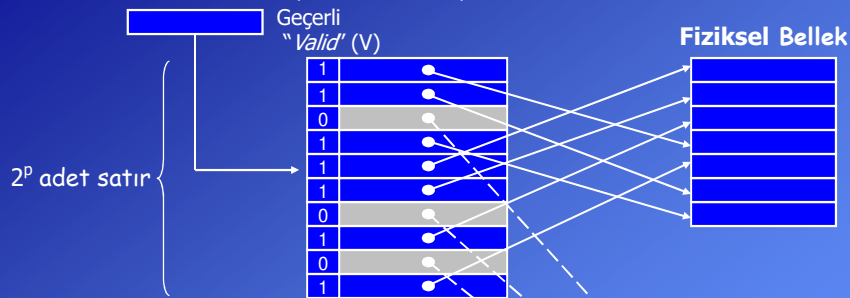
Sayfa ana belleğe getirilirken MİB başka bir prosesi çalıştırır.

Programın sadece gerekli olan sayfaları ana (fiziksel) belleğe getirilir (demand paging).

Yöresellik sayesinde zaten program belli bir süre aynı sayfaya erişmeye devam edecektir.

### Sayfa Tablosu (PT)

(fiziksel blok numarası (page frame number) ya da diskteki yeri)

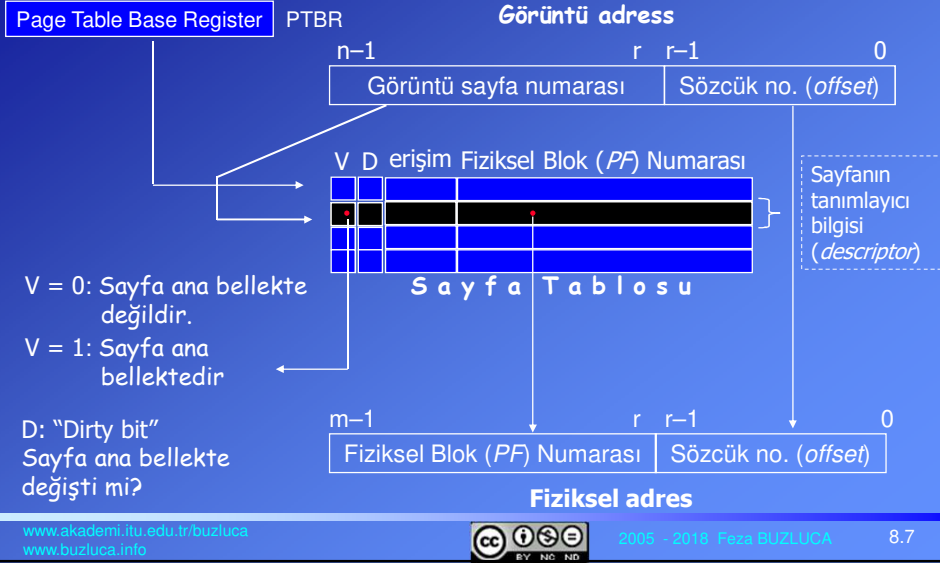


Sayfa Tablosunda o sayfayla ilgili ek bilgiler de yer alır:

- Sayfanın erişim hakları (RWX)
- Değişim biti D (dirty): Sayfanın ana bellekteyken değişip değişmediğini gösterir.

Sayfanın diskteki yeri (track-sector) sayfa tablosunda tutulabileceği gibi bu iş için işletim sistemi başka bir tablo da kullanabilir.

O anda işlemekte olan sürece (process) ilişkin sayfa tablosunun fiziksel bellekteki yeri bellek yönetim biriminde (MİB) bulunan bir saklayıcıda (Page Table Base Register - PTBR) tutulur.



### İsteğe bağlı sayfalama (Demand Paging):

Bir program çalışmaya başladığında işletim sistemi o andaki aktif sürecin (proses) adres uzayının küçük bir bölümünü diskten ana belleğe kopyalar.

Kopyalanan bu bölüm genellikle komutların yer aldığı ilk sayfayı ve programın ilk başta gerek duyacağı verileri içerir.

Program çalıştıkça prosesin adres uzayında sadece gerek duyulan sayfalar diskten ana belleğe getirilir.



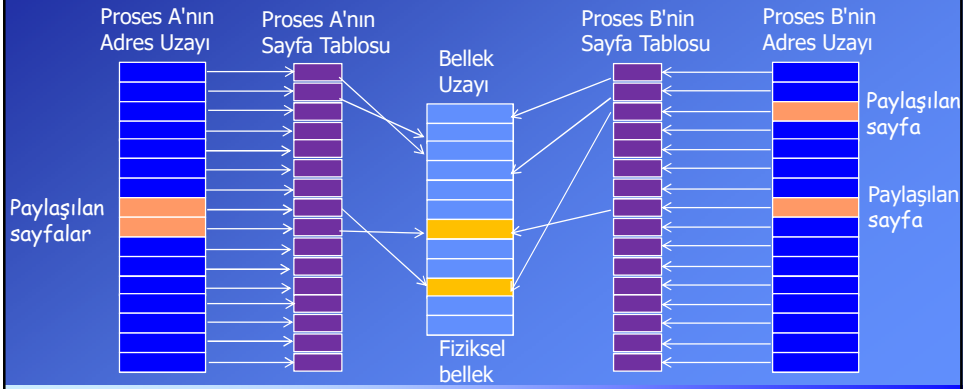
**Paylaşılan Bellek:**

Varsayılan çalışmada iki sürecin adres uzayları birbirinden ayrılır ve korunur.

Paylaşılan bellek yönteminde süreçler arası (*inter-process*) iletişim sağlamak için iki sürecin adres uzayları kesiştirilir.

Paylaşılan sayfalar ana bellekte aynı fiziksel çerçeveye yerleştirilirler.

Her iki sürecin sayfa tablosunda da paylaşılan sayfa için aynı çerçeve numarası yazılır. Farklı görüntü adresler aynı fiziksel adrese dönüşür.

**Adres dönüşümünün hızlandırılması (Translation Lookaside Buffer - TLB) :**

Her görüntü adres - fiziksel adres dönüşümü için sayfa tablosuna erişmek gerektiğinden fazladan bir bellek erişimi yapılmış olur.

Bu işlemi hızlandırmak için programların yöresellik özelliğinden yararlanılarak sık erişilen sayfalar ve onların deskriptörleri (ana bellekteki blok numarası, erişim hakları) bir çağrışimli bellekte tutulur (cep bellek mantığı).

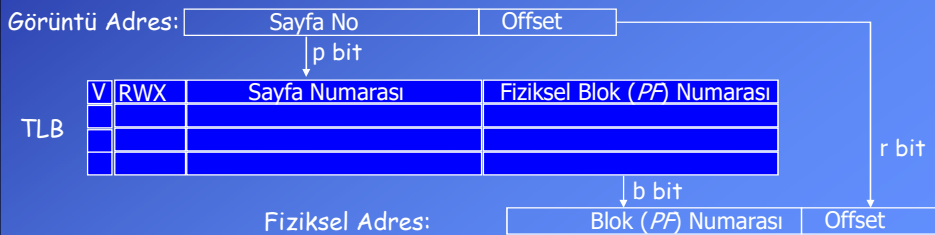
Bu belleğe **öngörü tablosu** (Translation Lookaside Buffer - TLB) denir.

Bir görüntü adres üretildiğinde donanım önce TLB'ye başvurur.

Eğer aranan sayfa numarası TLB'de varsa (vuru) donanım sayfa tablosuna gerek duymadan dönüşümü daha hızlı yapar.

Sayfa tabloda yoksa (TLB'de ıskı) gerekli dönüşüm bilgisi sayfa tablosundan alınır.

TLB dolduğunda bir yer değiştirme (*replacement*) algoritması uygulanır (FIFO, LRU).



**Sayfa Değişirme (Page Replacement)**

Ana bellek doluyken ana bellekte olmayan yeni bir sayfaya başvurulursa bir yer değiştirme algoritmasına göre ana bellekteki bir bloğun boşaltılarak yerine yeni sayfanın yerleştirilmesi gerekir.

LRU (*Least Recently Used*) yönteminde ana bellekte olan sayfalara (bloklara) yaşlanma sayacı (*aging counter*) atanır.

1. Bir sayfaya başvurulduğunda o sayfanın sayacı sıfırlanır.
2. Sadece başvuru olan sayfanın sayacından küçük olan sayaçlar bir arttırılır.
3. Bir yer değiştirme işlemi gerektiğinde en yüksek sayaç değerine sahip olan sayfa ana bellekten kaldırılır, yeni sayfa onun yerine koyulur sayacı sıfırlanır, diğer sayaçlar bir arttırılır.

**Örnek:** Başvuru zinciri: Başvuru olan sayfa numaraları: 0, 1, 2, 3, 0, 3, 4, 5  
Ana bellekte 4 blok yer var. Bu durumda 2 bitlik sayaçlar yeterli olur.

Sayaç	0	1	2	3	0	3	4	5
Başvuru olan sayfa	0	1	2	3	0	3	4	5
00	0	01	10	11	00	01	10	11
01	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1
11	2	2	2	2	2	2	2	2
00	3	3	3	3	3	3	3	3
01	4	4	4	4	4	4	4	4
10	5	5	5	5	5	5	5	5
11	3	3	3	3	3	3	3	3

**Kırıntılanma (Fragmentation)**

Hatırlatma: Coğrafi yöresellikten yararlanmak ve DMAC kullanabilmek için veriler sayfalar şeklinde aktarılır

Süreçlerin adres uzayı sabit boyutta sayfalara bölünür.

Bir süreç bir sayfanın (örneğin en sondaki) tamamını kullanmayabilir.

Örneğin bir sistemdeki sayfalar 1K x sözcük boyutundaysa, 5K+1 sözcük boyutunda bir program 6 adet sayfa yer kaplayacaktır ancak son sayfanın sadece 1 sözcüklük kısmı anlamlı olarak kullanılacaktır.

Aslında son sayfa sadece 1 sözcüklük veri içeriyor ama ana belleğe tüm bir sayfa olarak aktarılacak ve burada bir blok yer kaplayacaktır.

Diğer süreçler bu bloğu kullanamaz.

Bir süreç bir sayfadan daha küçük de olabilir, buna rağmen bellekte bütün bir blok yer kaplar.

Sayfalı sistemlerde sayfalar daha küçük parçalar bölünmezler, bir bütün olarak aktarılırlar.

Bir sayfadaki son bölümlerin kullanılmaması problemine iç kırıntılanma (*internal fragmentation*) denir.

Bunun yanı sıra bazı programlar ve veriler görüntü belleğin tamamına gerek duymazlar. Bu nedenle  $2^P$  satırlı sayfa tablosunun bazı satırları hiç kullanılmaz.

## 8.2 Segmanlı Sayfalı Dönüşüm (Segmented-Page Mapping):

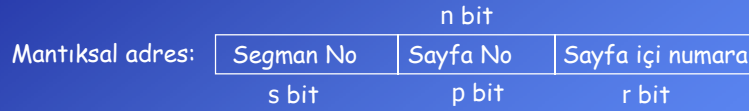
Mantıksal adres uzayı **segman** adı verilen değişken uzunluktaki mantıksal birimlere bölünür.

Her segman değişik sayıda sayfadan oluşabilir.

Bir segman, mantıksal bir program parçası veya veridir. Bütün bir program, bir alt program, dizi, matris bir segman oluşturabilir.

Bu yöntem ile mantıksal bir bütün oluşturan program parçaları ve veriler (segmanlar) üzerinde erişim ve paylaşım denetimi oluşturulabilir.

Mantıksal adres üç alandan oluşur:



Bu sistemim özellikleri:

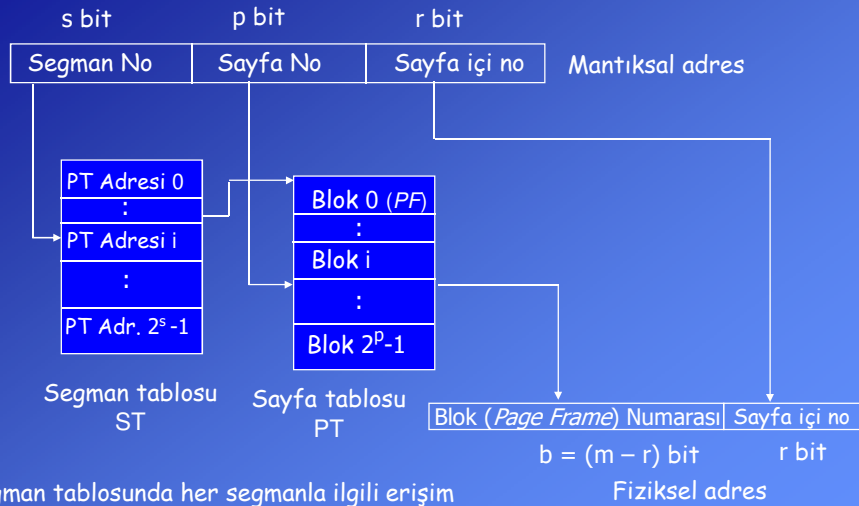
Mantıksal adres uzayı:  $2^n$  sözcük

Segman sayısı:  $2^s$

Bir segmandaki sayfa sayısı: 1,  $2^p$  arası

Bir sayfa boyu:  $2^r$  sözcük

## Sayfalı segmanlı yöntemde adres dönüşümü



**Adres dönüşümünün hızlandırılması (Translation Lookaside Buffer - TLB):**

Her görüntü adres - fiziksel adres dönüşümü için segman ve sayfa tablosuna erişmek gerektiğinden fazladan iki bellek erişimi yapılmış olur.

Bu işlemi hızlandırmak için sık erişilen segman no, sayfa no ve onların deskriptörleri (ana bellekteki blok numarası, erişim hakları) bir çağrışimli bellekte (TLB) tutulur. TLB segman ve sayfa tablolarının cep belleği gibi işlem görür.

Mantıksal Adres:

**8.3 Bellek erişimindeki tüm işler: Sayfalı Segman, TLB, Cep Bellek**

1. MİB mantıksal adres üretir : s, p, w (segman, sayfa, sözcük numarası (word) ).
2. TLB'de arama yapılır.
  - a) Aranılan segman (s) ve sayfa (p) TLB'de varsa ve erişim hakları uygunsa
    - TLB'den fiziksel blok numarası alınır ve fiziksel adres oluşturulur.
    - Cep bellek erişimi işlemlerine başlanır. Bkz. 3. madde.
    - LRU kullanılıyorsa TLB'de sayaçlar güncellenir.
  - b) Aranılan segman ve sayfa TLB'de yoksa
    - Segman tablosundan (s kullanılarak) ilgili sayfa tablosunun adresi alınır.
    - Sayfa tablosunda mantıksal adresteki sayfanın numarası p aranır.
    - i. Sayfa ana bellekte ise
      - Fiziksel blok numarası PT'den alınır ve fiziksel adres oluşturulur.
      - Cep bellek erişimi işlemlerine başlanır. Bkz. 3. madde.
      - TLB güncellenir : Yeni s, p ve fiziksel blok numarası kayıt edilir. Gerekirse yer değiştirme yapılır.
      - LRU kullanılıyorsa sayfa tablosunda yaşlanma sayaçları güncellenir.



- b) Aranana segment ve sayfa TLB'de yoksa (devamı)
- ii. Sayfa ana bellekte yoksa
- **Sayfa hatası** (*page fault*) oluşur.
  - Sayfa fiziksel belleğe getirilir.
  - Sayfa tablosu güncellenir : Adres, sayaçlar.
  - Fiziksel adres oluşturulur.
  - Cep bellek erişimi işlemlerine başlanır. Bkz. 3. madde.
  - TLB güncellenir : Yeni s, p ve fiziksel blok numarası. Gerekirse yer değiştirme.
3. Fiziksel adres kullanılarak ilgili erişim yöntemine (*mapping*) göre cep bellekte veri aranır (çağrışımlı, doğrudan, kümeli çağrışımlı).
- a) Aranana veri cep bellekte ise
- Veri cep bellekten alınır.
  - LRU kullanılıyorsa cep bellekte sayaçlar güncellenir.
- b) Aranana veri cep bellekte yoksa
- Veri ana bellekten okunur ve ana bellek ile cep bellek arasında blok aktarımı yapılır.
  - Cep bellekteki sayaçlar güncellenir.