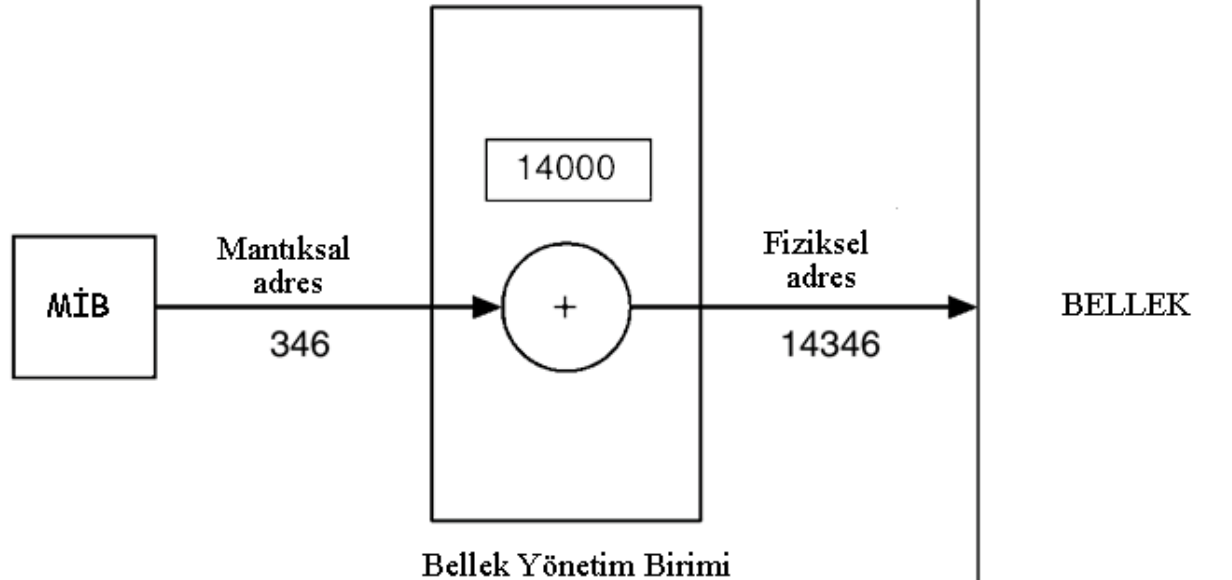


[Bellek Yönetimi]

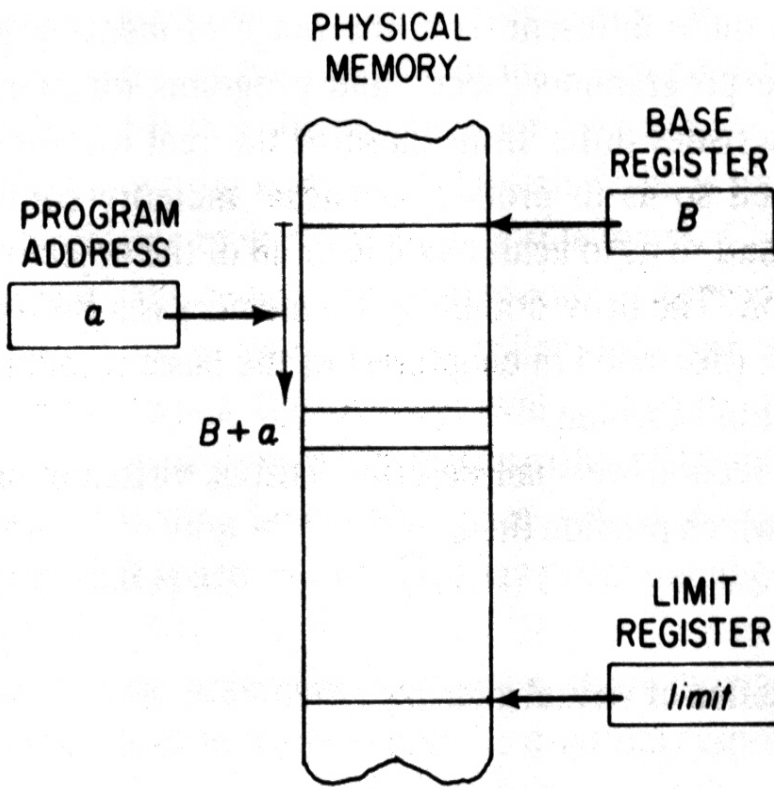
- Bellekteki herhangi bir işlemi başka bir yere aktarabilmelidir.
- Birden fazla işlem veya kullanıcı olduğunda bir kullanıcının diğer kullanıcıyı alanlarına girmeleri önlenmelidir.
- Kullanıcılar arası kaynak paylaşımını sağlamalıdır.
- Belleğin mantıksal alanlara bölünmesini sağlayarak bilgiye erişimi kolaylaştırmalıdır.
- Belleğin yetmediği durumlarda fiziksel başka bellek alanlarını yani hard diskleri kullanabilmelidir.

Taban kaydedicisini kullanarak belleğe dinamik yerleşim



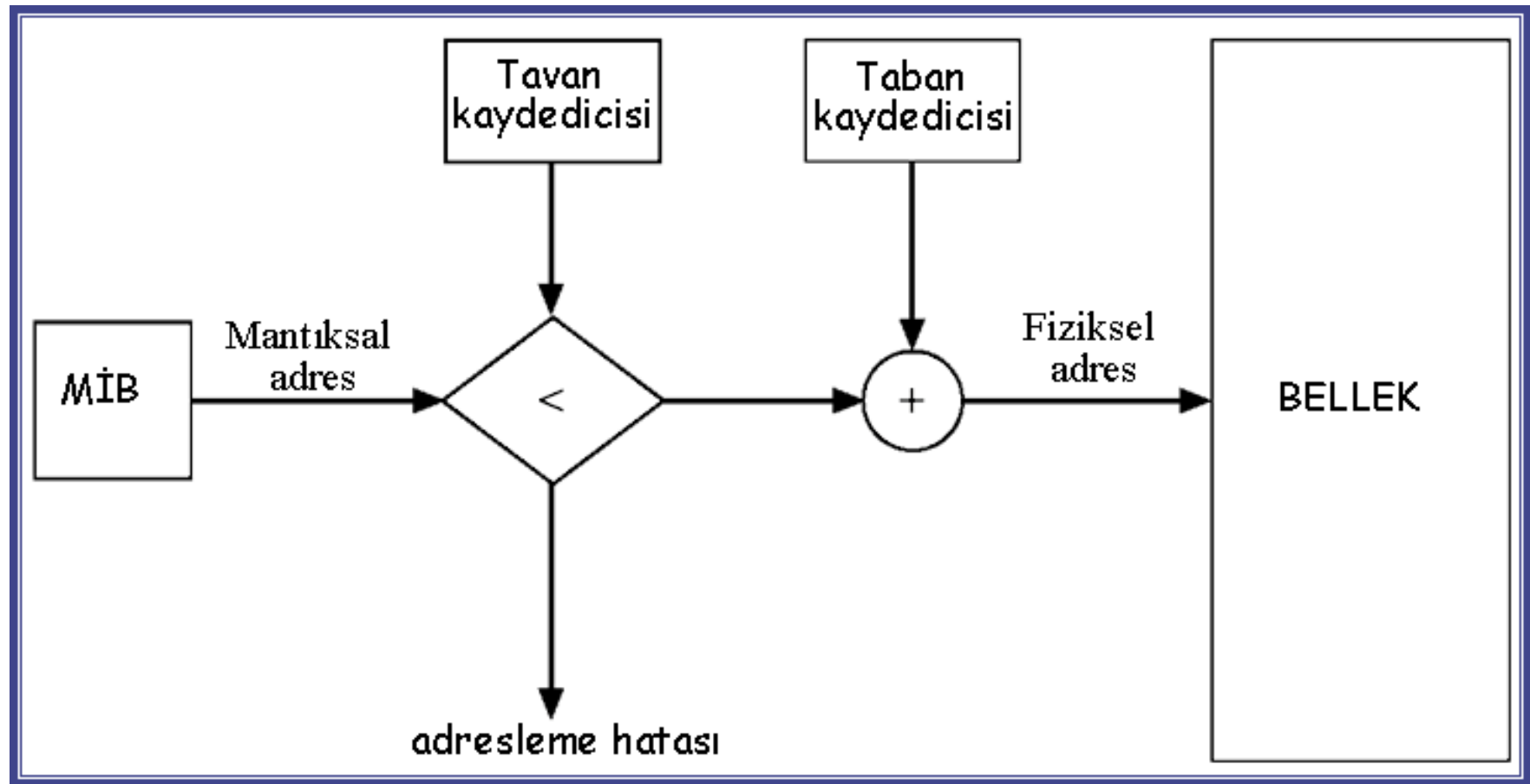
- MİB tarafından oluşturulan adrese mantıksal (veya sanal) adres,
- Bellekte yer alan adrese ise fiziksel adres denilmektedir.
- Bir kullanıcı işlemi belleğe yerleşirken tekrar yerleştirme (relocation) de denilen taban (base) kaydedicisi kullanılır.

Tavan(Limit) ve Taban(Base) kaydedicileri



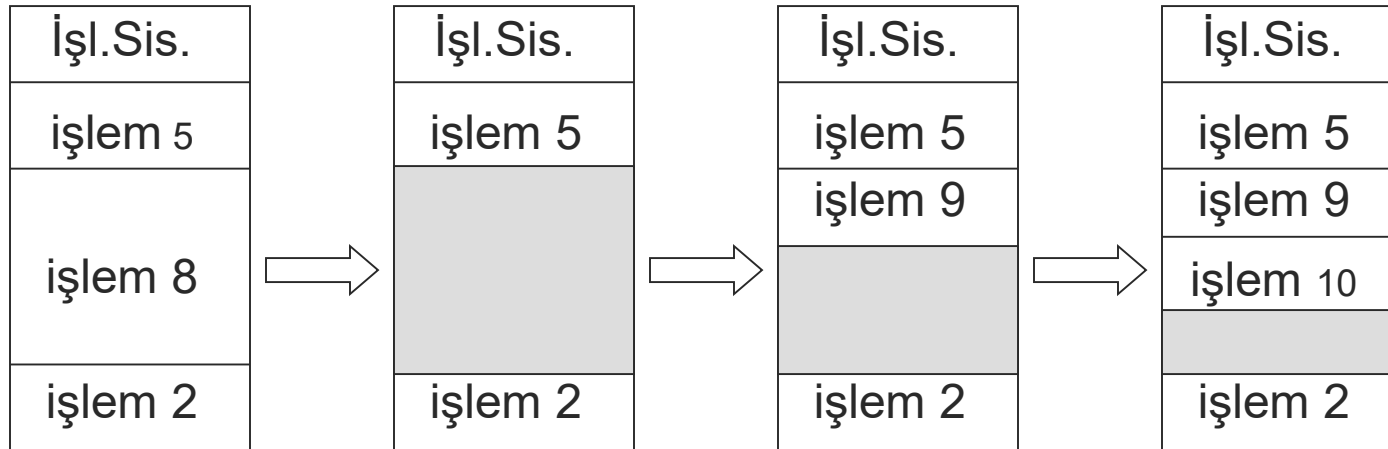
- Bu adres haritası aşağıdaki kriterlere göre donanımsal olarak adreslenir;
 - eğer $a < 0$ ise bellek koruma hatası
 - $a' = B + a$
 - eğer $a' > \text{Tavan}$ ise bellek koruma hatası
 - a' burada istenilen bellek yeridir.

Bellek koruma



Bitişik Yerleşim

- Ana bellekte hem işletim sisteminin hem de kullanıcı işlemlerinin yer almaktadır.
- Bellekte kullanıcı işlemleri boşluklara (hole) bloklar halinde yerleştirilmektedir.
- Yeni bir işlem geldiğinde bu belleğe işlemin yeteceği kadar bir alan ayrılarak konumlandırılmakta ve işletim sistemi boş alanlar ve dolu alanlar ile ilgili bilgileri kayıt altına almaktadır.



İşletim sistemi bellekteki n adet boşluğun bulunduğu listeden işlem için en uygun boşluğu belirlemede 3 strateji kullanır;

- İlk uyan (First-fit): Listedен işlem için yeterli büyüklükte olan ilk boşluğun seçilerek işlemin yerleştirilmesidir.
- En iyi uyan (Best-fit): Listedен işleme uygun büyüklükteki boşlukların aranarak bulunması ve bunların içinden en küçüğünün seçilerek işlemin yerleştirilmesidir.
- En kötü uyan (Worst-fit): Listedен araştırma yapıldıktan sonra en büyük boşluğun seçilerek işlemin yerleştirilmesidir.
 - İşlemin belleğe yerleştirilme hızı ve belleğin etkili kullanımı açısından ilk uyan ve en iyi uyan stratejileri en kötü uyan stratejisinden çok daha kullanışlıdır.

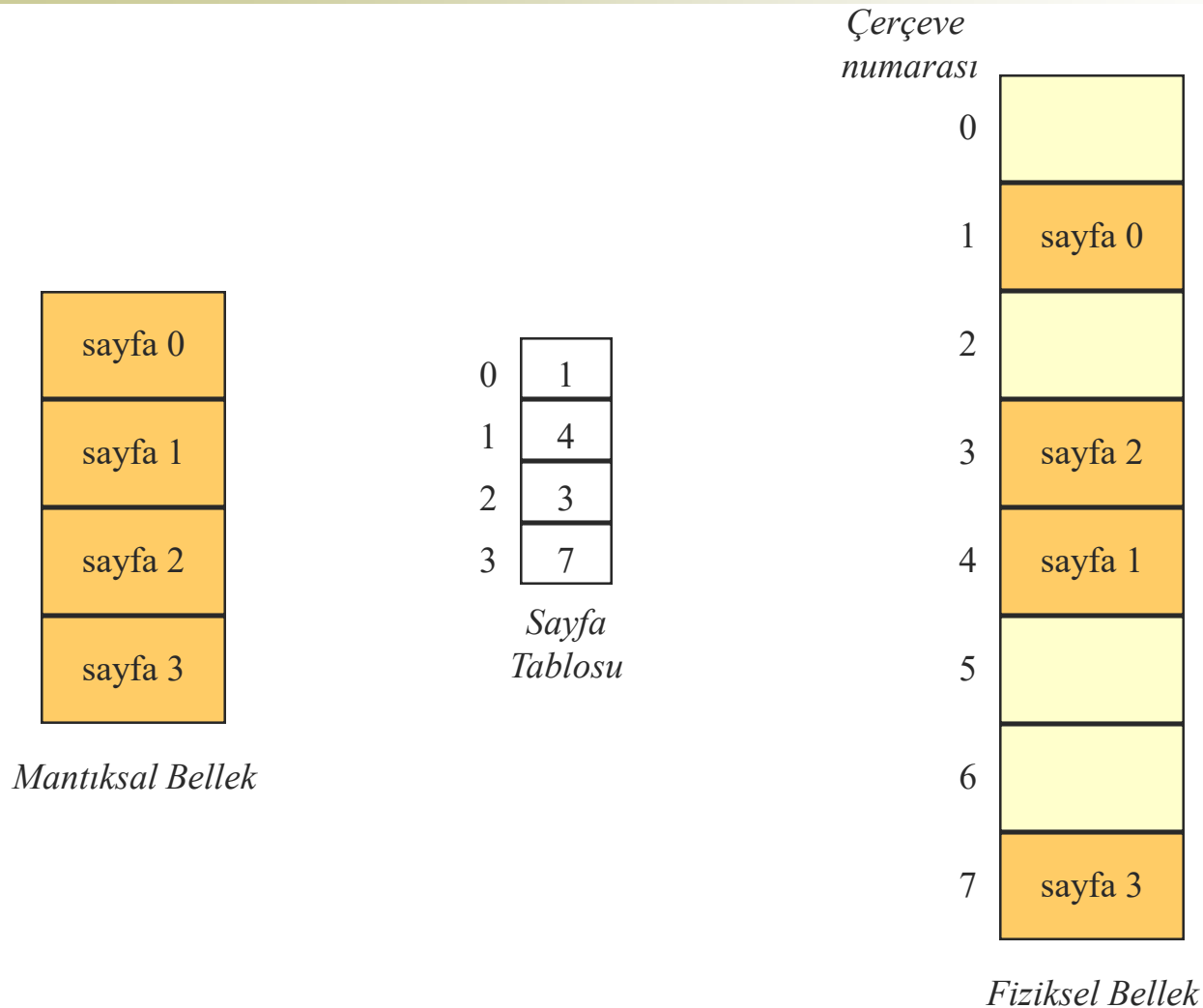
[Sayfalama (Paging)]

- Fiziksel bellek üzerindeki aynı uzunluktaki bloklara 'çerçeve (frame)',
- Mantıksal bellek üzerindeki aynı uzunluktaki bloklara ise 'sayfa (page)' denilmektedir.
- Sayfaların uzunlukları donanım tarafından belirlenmektedir. (2'nin katları; 512 byte, 16 MB,...)

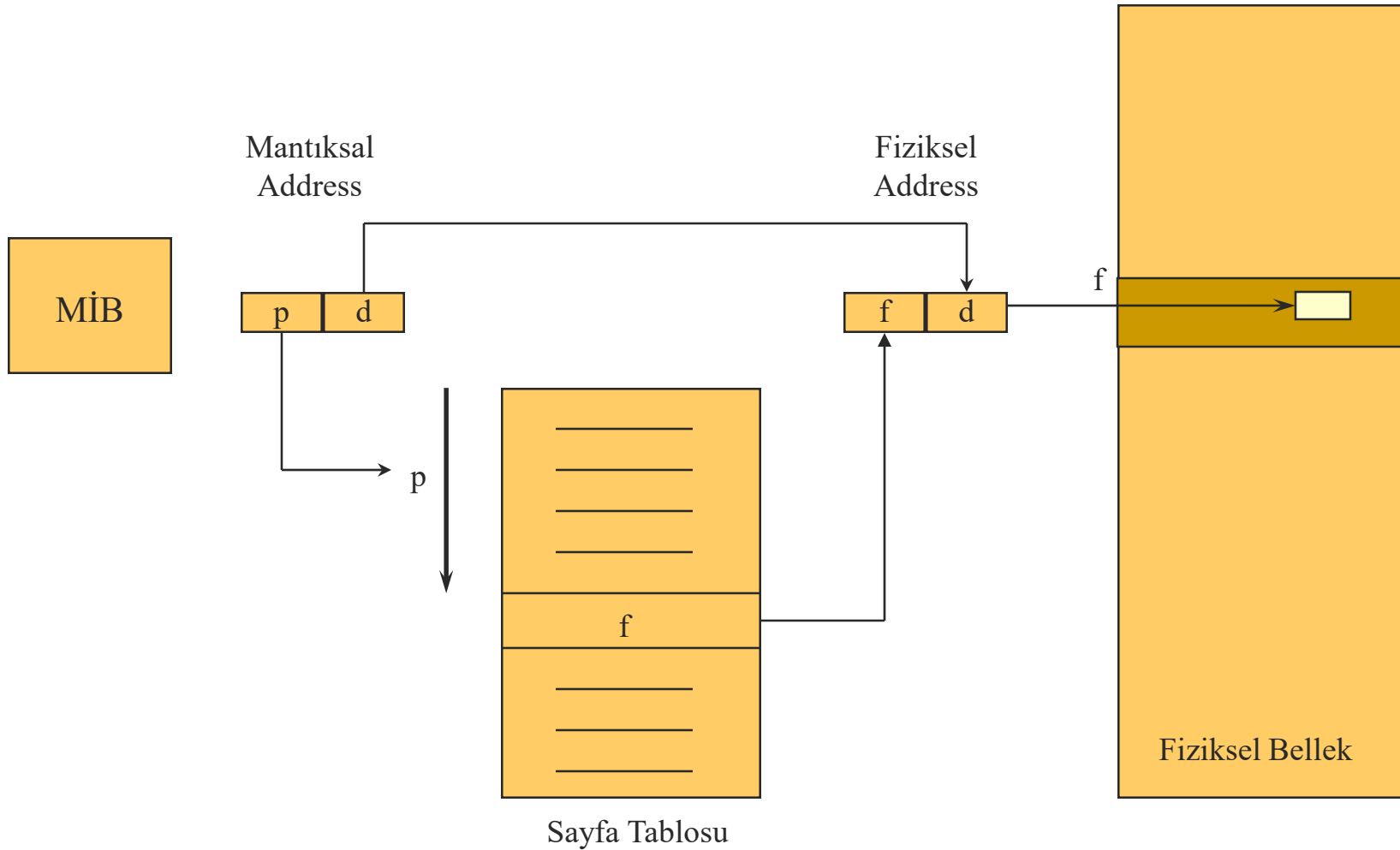
[Sayfalama (Paging)]

- MİB'nin oluşturduğu adres iki bölümden oluşmaktadır;
 - sayfa numarası (p): fiziksel bellekteki her bir sayfanın taban adresini tutan sayfa tablosundaki gösterge.
 - sayfa ofseti (d): taban adresi ile birleştirilerek fiziksel bellekte sayfanın içerisindeki yerin belirlenmesinde kullanılır.
- Sayfa tablosunda hem sayfa hem de ofset adresi bulunmaktadır.

Fiziksel ve mantıksal belleğin sayfalama modelleri



[Sayfalama]



Sayfalama örneği; 4 byte uzunluğunda sayfalardan oluşan 32 byte'lık bir bellek olduğunu düşünelim.

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

Mantıksal
bellek

0	5
1	6
2	1
3	2

Sayfa
tablosu

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

Fiziksel bellek

- Mantıksal adresi 0'da sayfa ve ofset adresi 0'dır. Sayfa tablosuna bakıldığında sayfa 0'ın çerçeve 5'de olduğu görülmektedir.
- Böylece mantıksal adres 0, fiziksel bellekte 20'ye $((5 \times 4) + 0)$ karşılık gelmektedir.

Sayfalama örneği; 4 byte uzunluğunda sayfalardan oluşan 32 byte'lık bir bellek olduğunu düşünelim.

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

Mantıksal
bellek

0	5
1	6
2	1
3	2

Sayfa
tablosu

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

Fiziksel bellek

- Mantıksal adres 3 (sayfa 0, ofset 3) fiziksel bellekte 23'ü $((5*4))+3$ adreslemektedir.

Sayfalama örneği; 4 byte uzunluğunda sayfalardan oluşan 32 byte'lık bir bellek olduğunu düşünelim.

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

Mantıksal
bellek

0	5
1	6
2	1
3	2

Sayfa
tablosu

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

Fiziksel bellek

- Mantıksal adres 4 (sayfa 1, ofset 0), sayfa tablosuna bakıldığında çerçeve 6'dadır ve fiziksel bellekte 24'ü $((6*4))+0$ adreslemektedir.

Sayfalama örneği; 4 byte uzunluğunda sayfalardan oluşan 32 byte'lık bir bellek olduğunu düşünelim.

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

Mantıksal bellek

0	5
1	6
2	1
3	2

Sayfa tablosu

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

Fiziksel bellek

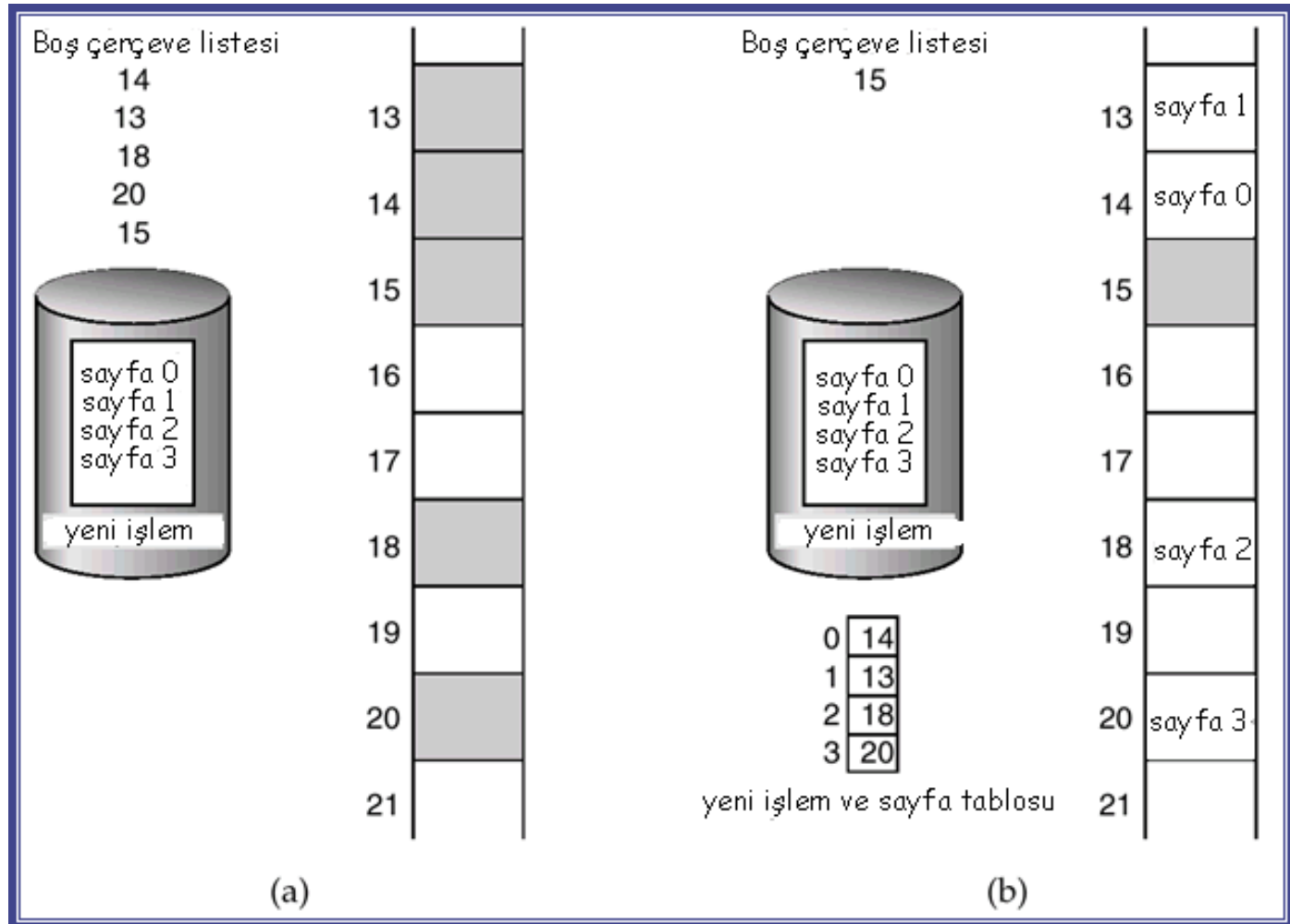
- Mantıksal adres 13, fiziksel adreste 9'u $((2*4)+1)$ adreslemektedir.

[Boş çerçeveler]

- Bir işlem sistem tarafından çalıştırıldığında, işlemin kullanmak istediği kadar sayfa sayısına göre bellekte gerekli sayıda çerçeve ayrılmaktadır.
- İlk sayfa ayrılan bir çerçeveye yerleştirilir ve bu çerçevenin numarası sayfa tablosuna konulur, diğer bir sayfa bir çerçeveye yerleştirildiğinde o çerçevenin numarası da sayfa tablosuna işlenmektedir.

Boş çerçeveler

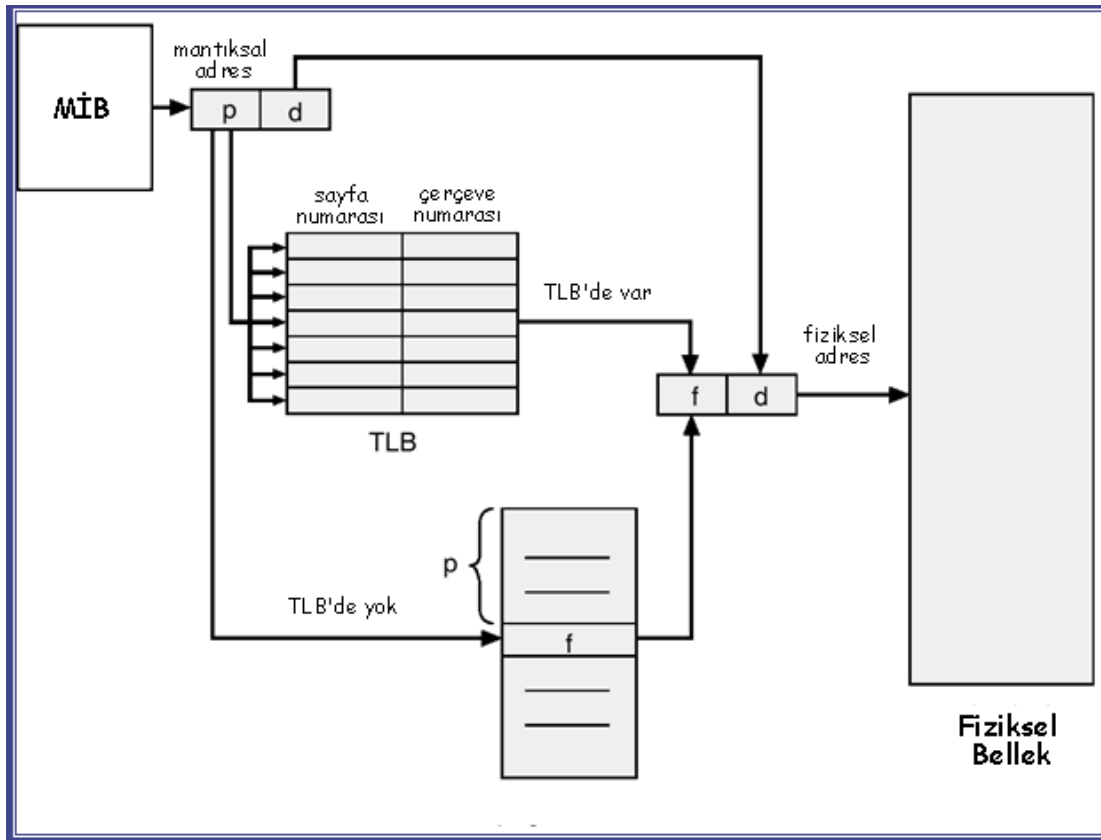
(a) Yerleşimden önce ve (b) Yerleşimden sonra



TLB (Translation Look-aside Buffer)

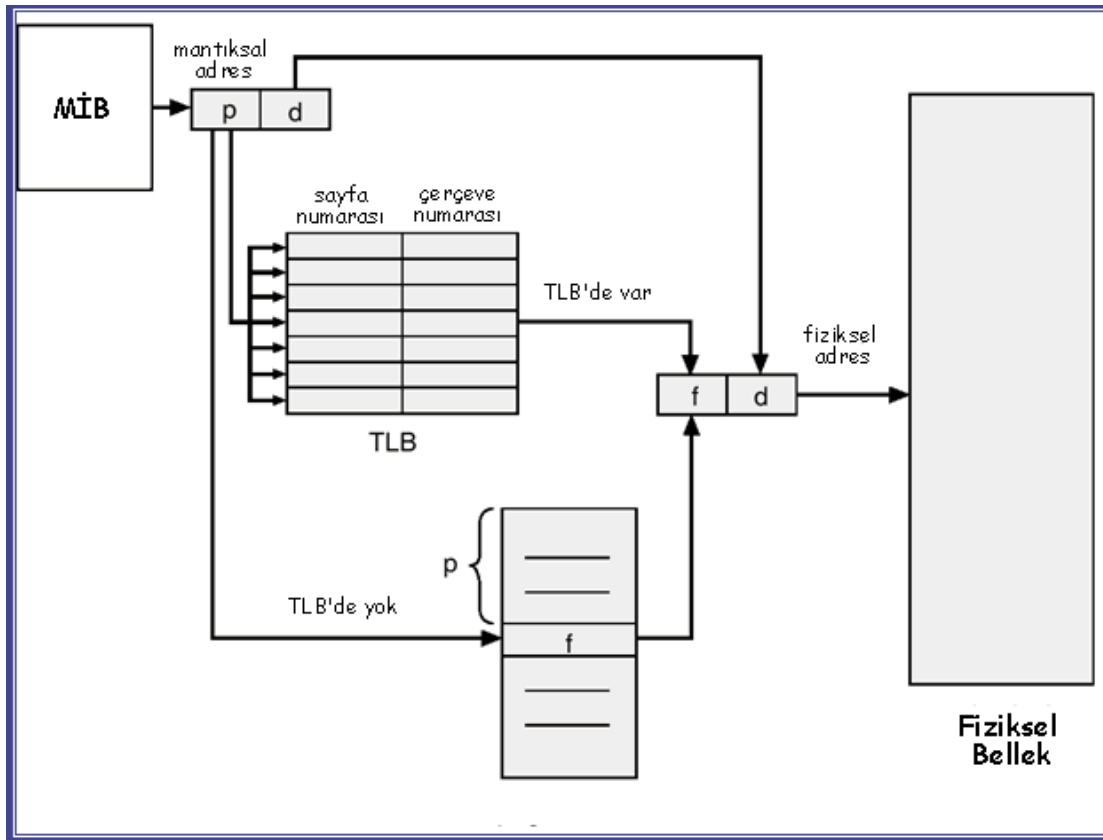
- Sayfa tablosunda çerçeve adresini ararken kaybedilen süreyi önlemek için kullanılan özel, küçük ve hızlı bir ön bellektir.
- Burada bir anahtar ve değer bulunmaktadır. Bulunmak istenen bilgi için tüm anahtarlara aynı anda hızlı bir biçimde bakılmaktadır.
- TLB sayfa tablosu ile birlikte kullanılır.
- TLB sayfa tablosunun sadece bir kısmını kapsar.

TLB (Translation look-aside buffer)



- MİB tarafından mantıksal bir adres tanımlandığında bu sayfa numarası TLB'ye iletilir.
- Sayfa numarası burada bulunursa çerçeve adresi hızlı bir biçimde kullanılarak belleğe ulaşılır.

TLB (Translation look-aside buffer)



- Eğer sayfa numarası TLB'de bulunmazsa (TLB'de eksik olarak bilinir) sayfa tablosuna gidilerek çerçeve adresi bulunur ve bu TBL'ye eklenir ki böylece bir daha ihtiyaç olduğunda bu adres hızlı bir biçimde bulunabilir.
- Eğer TLB alanı dolarsa, işletim sistemi buradaki en az kullanılan değerleri bularak bunların yerine yeni değerleri ekleyecektir.

[Bölümleme (Segmentation)]

- Bölümlemeye; mantıksal adres alanı bölümlerden oluşmaktadır.
- Adres hem bölüm numarası hem de bölüm içerisindeki adresi belirten ofset numarasından meydana gelmektedir.
 - <bölüm numarası, ofset>

[Bölümleme (Segmentation)]

- Bölüm tablosunda;
 - bölümün başlangıç yerini tutan bölüm taban ve
 - bölümün uzunluğunu yani bitiş yerini belirlemede kullanılan bölüm tavan kaydedicileri bulunmaktadır.



Bölümleme (Segmentation)

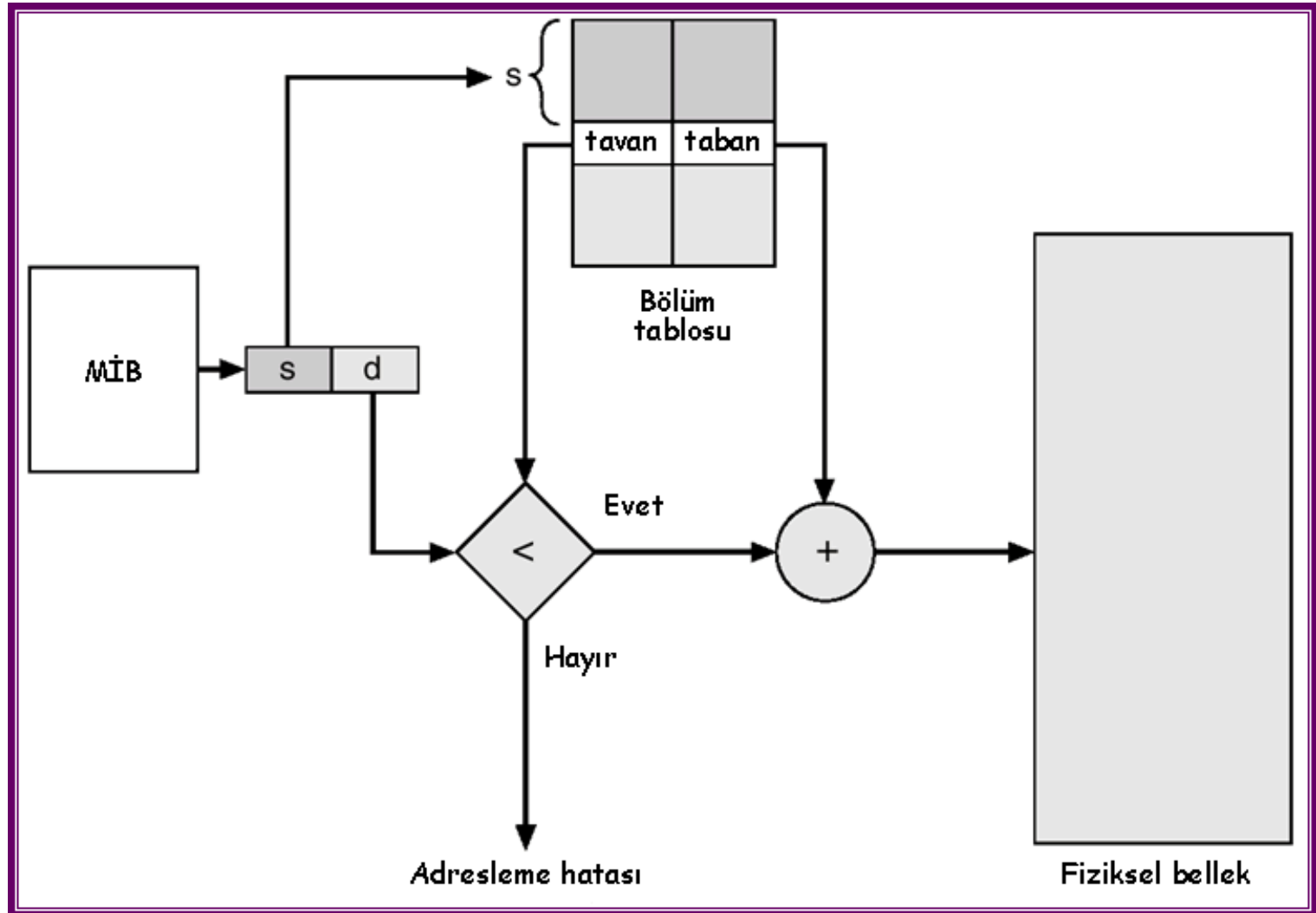


- MİB tarafından oluşturulan mantıksal adreste;
 - s: bölüm numarasını,
 - d: bölümdeki ofset adresi göstermektedir.

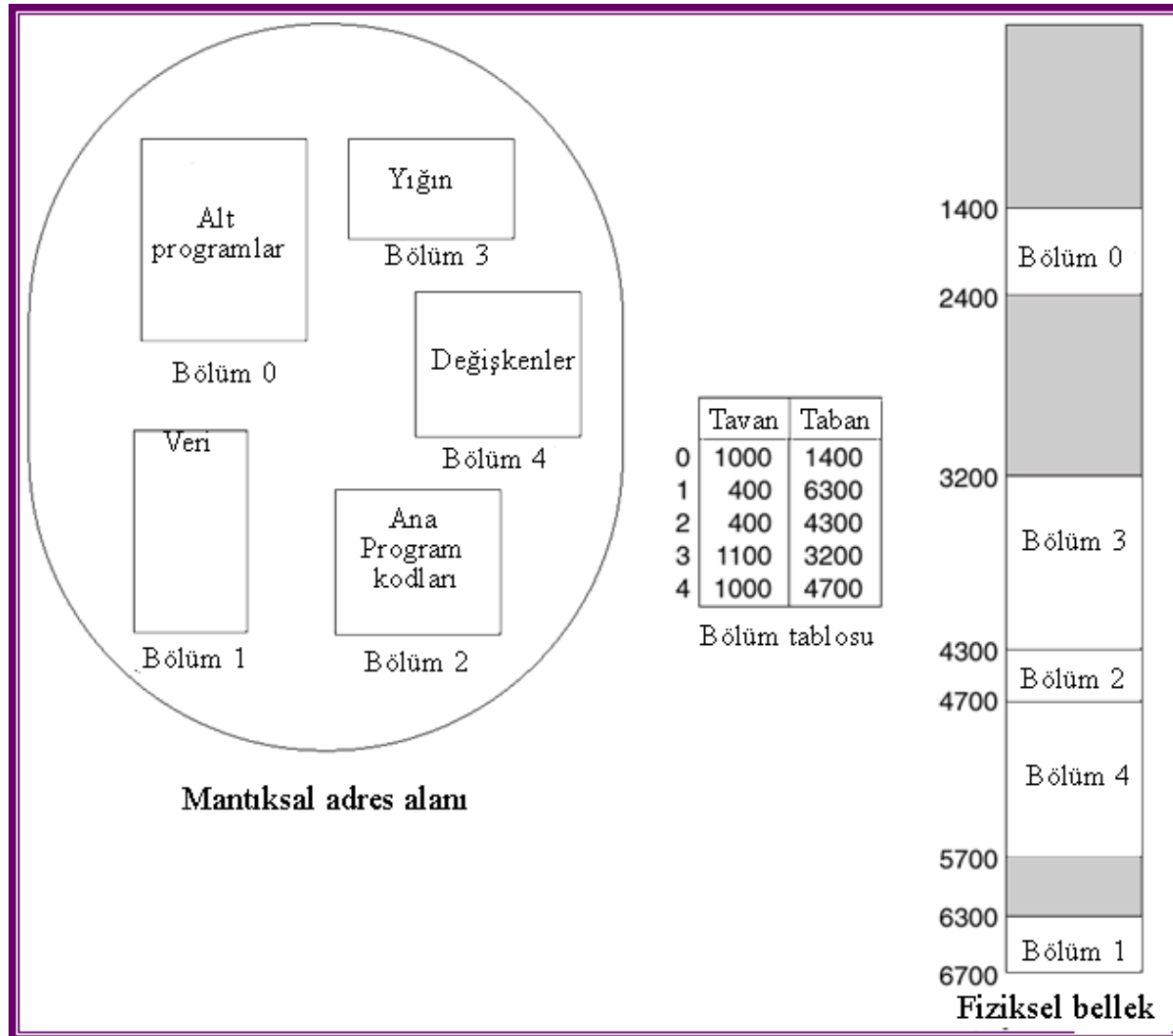
[Bölümleme (Segmentation)]

- Ofset adres, taban ve tavan kaydedicisi değerleri arasında olmalıdır.
 - Eğer değeri fazla olursa mantıksal adresin bölümü aşması nedeniyle işletim sistemi devreye girmektedir.
 - Eğer değeri bu belirtilen sınırlar arasında ise bölüm taban kaydedicisi ile bu adres toplanmakta ve bu şekilde istenilen byte'a ulaşılabilirilmektedir.

[Bölümleme]



Bölümleme örneği



Sayfalama ve bölümlleme arasındaki farklar;

- Bölümlemenin amacı adres alanının mantıksal olarak dilimlenmesidir.
- Sayfalama ise belleğin fiziksel olarak dilimlenip bir düzeyli bir bellek (sabit disk, bellek farkı gözetmeden erişim) oluşturulması amacına yöneliktir.
- Sayfalar makine donanımına bağlı olarak sabit boyuttur.
- Bölümler ise kullanıcı tarafından belirlenecek boyuttur.

Sayfalama ve bölümlleme arasındaki farklar;

- Program adresinin sayfa ve ofset numaralarına ayrılması donanımın bir işlevidir. Ofset numarasının sınırı aşması otomatik olarak sayfanın çevrilmesine sebep olur.
- Oysa bölüm ve ofset numarasında bir sınır aşma hadisesi söz konusu değildir. Bu durumda bellek erişim hatası yani erişilmemesi gereken yere erişim hatası oluşur.