

Bölüm 9: Sanal Bellek





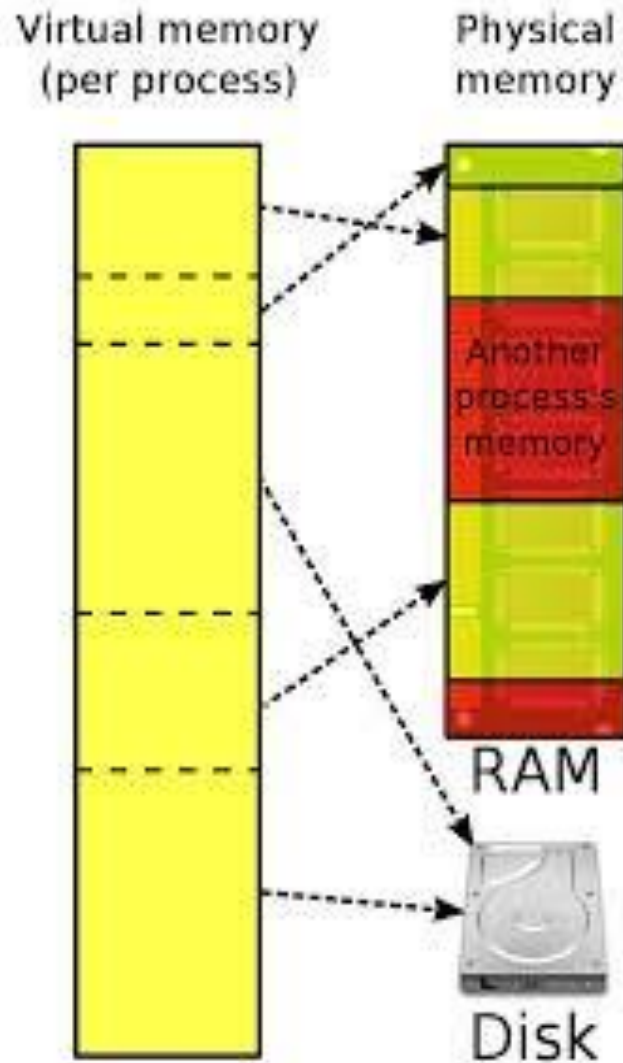
Background (Arka Plan)

- **Sanal Bellek** – Kullanıcı mantıksal hafızanın, fiziksel hafızadan ayrılması.
 - Programın bir parçasının hafızada olması bu ayrılmanın uygulanması için gereklidir.
 - Bu nedenle mantıksal adres alanı fiziksel adres alanından daha fazla yere sahip olabilir.
 - Daha verimli işlem üretimine imkan sağlar.
- Sanal Bellek şu yollar aracılığıyla uygulanır:
 - İsteğe bağlı sayfalama
 - Bölümlendirme isteği





Fiziksel hafızadan daha fazla yere sahip olan sanal hafıza





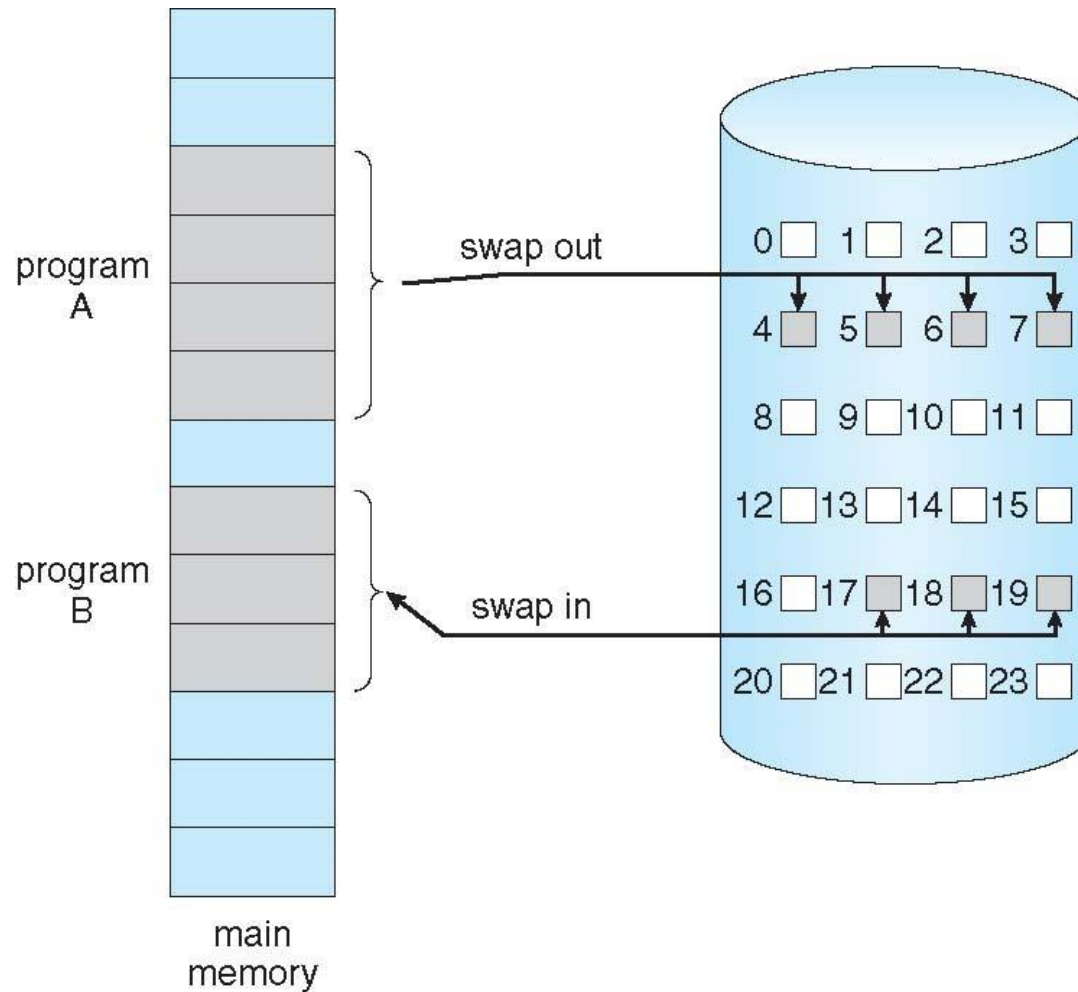
İsteğe Bağlı Sayfalama (Demand Paging)

- Sayfa sadece ihtiyaç olduğunda belleğe getirilir.
 - I/O ihtiyacı daha azdır.
 - Bellek kullanımı daha azdır.
 - Cevap verme süresi daha hızlıdır.
 - Çok kullanıcı desteği.
- **Lazy swapper (tembel yer değiştirme)** – Sayfaya ihtiyaç olana kadar o sayfayı belleğe getirmez.
 - Sayfaları idare eden yer değiştirici aynı zamanda bir sayfalayıcıdır.





Sayfalı belleğin bitişik disk alanına transferi





Bazı sayfalar ana bellekte değilken sayfa tablosu

0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

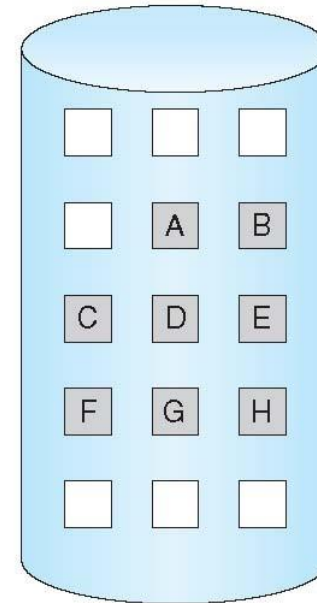
logical memory

valid-invalid bit	
frame	bit
0	4 v
1	i
2	6 v
3	i
4	i
5	9 v
6	i
7	i

page table

0	
1	
2	
3	
4	A
5	
6	C
7	
8	
9	F
10	
11	
12	
13	
14	
15	

physical memory





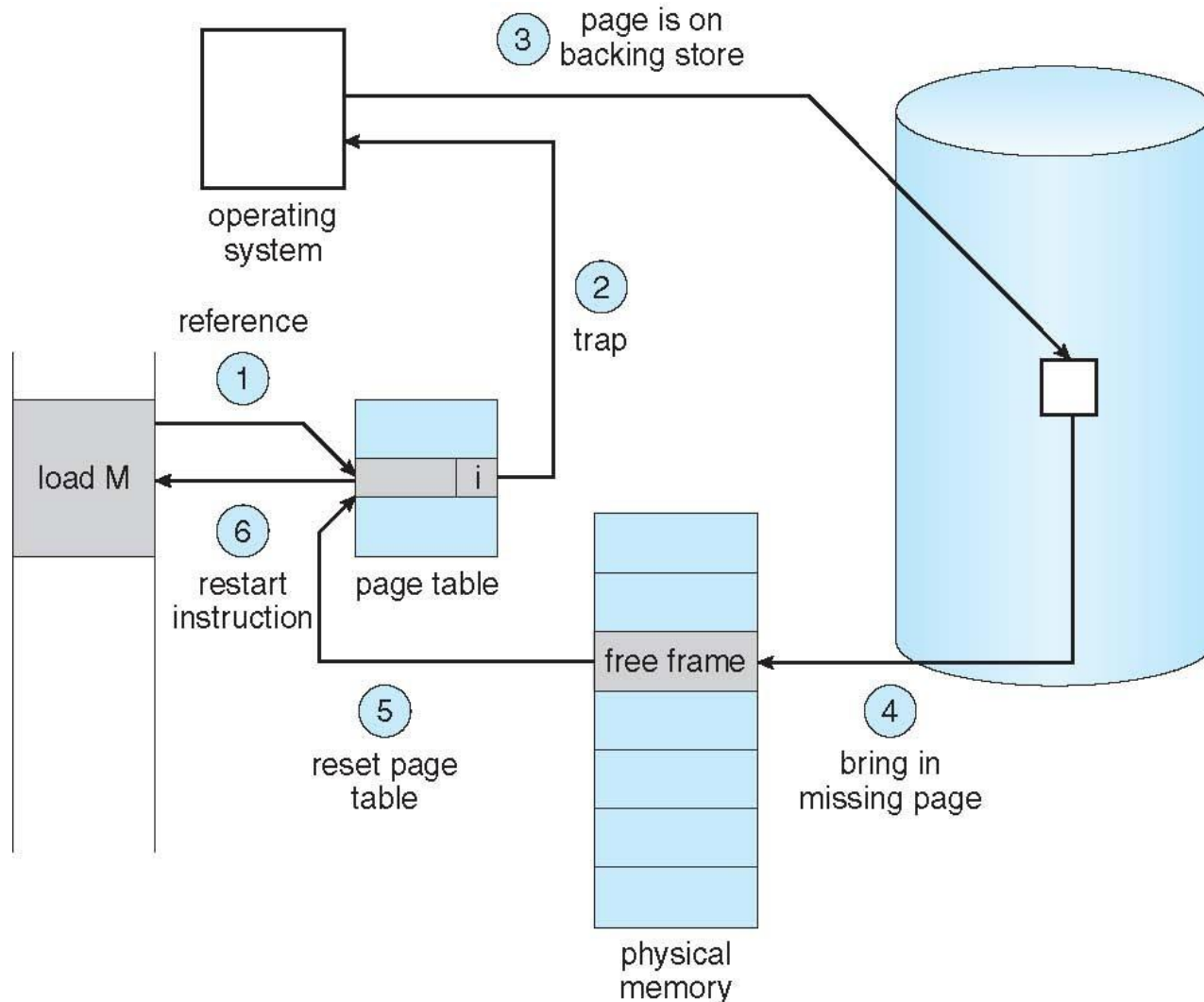
Page Fault (Sayfalama hatası)

- İşlem, hafızaya getirilmemiş bir sayfaya ulaşmaya çalıştığında ne olur?
- Geçersiz olarak işaretlenmiş bir sayfaya ulaşmaya çalışmak sayfalama hatasına sebep olur.
- Sayfalama donanımı geçersiz bit 'in ayarlandığını fark ettiğinde onu yakalayıp (trap) işletim sistemine götürür.
- Bu yakalama (trap) işletim sisteminin istenen sayfayı hafızaya götürmesini sağlar.





Sayfalama hatası durumunda yapılanlar





Boş alan olmadığında ne olur?

- Page replacement (Sayfa yer değişimi) – Bellekte yer tutan fakat o anda kullanımda olmayan başka sayfa bulur ve o sayfayı kaldırır.
 - algoritma
 - performans – Sayfa hatalarını (Page Faults) minimuma indirecek bir algoritma ister.





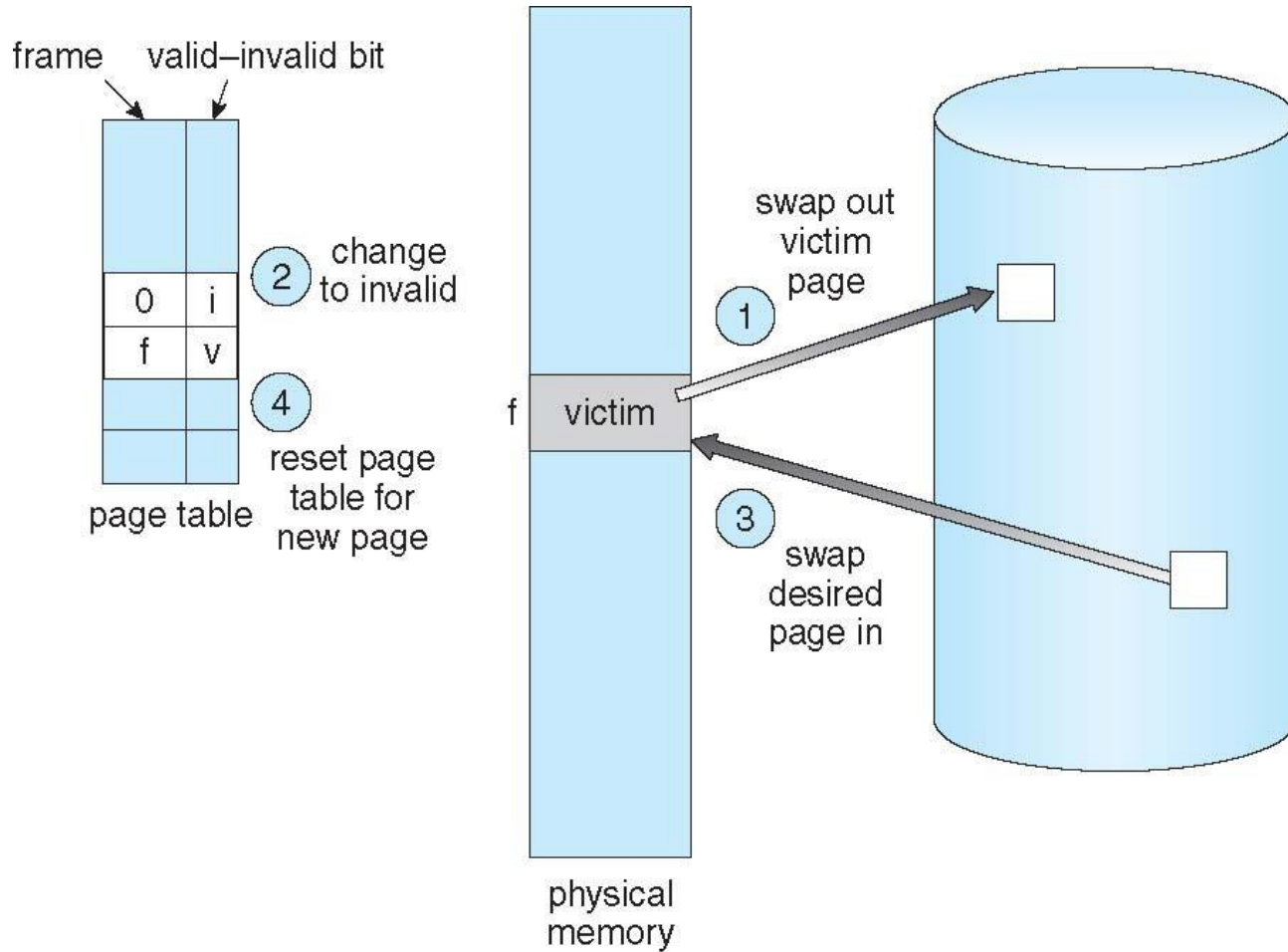
Basic Page Replacement (Temel sayfa yer değişimi)

1. Diskte istenen sayfanın yerini bulur.
2. Boş bir alan(frame) bulur:
 - Boş alan bulunduğunda oraya yerleşir.
 - Eğer boş alan bulamaz ise, sayfa yer değişimi algoritmasını kullanarak bir Victim frame (Kurban alan) seçer
3. İstenilen sayfayı yeni boş alana getirir sayfa ve alan tablolarını günceller.
4. İşlemi yeniden başlatır.



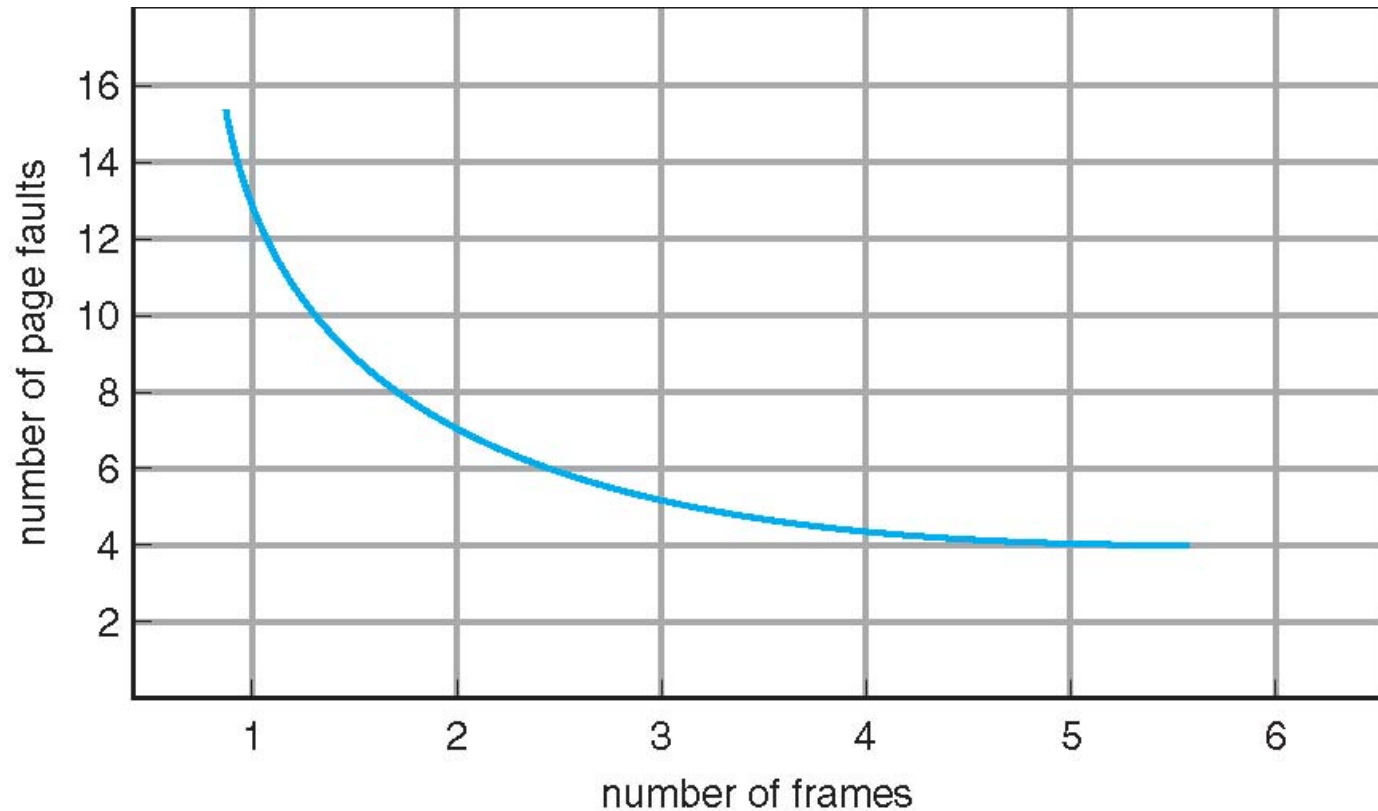


Sayfa yer deęiřimi





Sayfalama hatalarına karşın alan sayısı grafiği





FIFO Page Replacement (FIFO sayfa yer değişimi)

reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2			2	2	4	4	4	0			0	0			7	7	7
	0	0	0			3	3	3	2	2	2			1	1			1	0	0
		1	1			1	0	0	0	3	3			3	2			2	2	1

page frames





Optimal Algorithm (İdeal Algoritma)

- En uzun zaman aralığında kullanılmayan sayfa ile yer değişimi yapılır?
- Bunu nasıl bilebiliriz?
- Algoritmanızın ne kadar iyi işlediğini ölçmek için kullanılır.





Optimal Page Replacement (İdeal sayfa yer değişimi)

reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2		2		2		2		2						7		
	0	0	0		0		4		0		0						0		
		1	1		3		3		3		1						1		

page frames





Least Recently Used (LRU) Algorithm (Yakın Geçmişte En az kullanılan Algoritması)

- İdeal algoritma, bir sayfanın gelecekte kullanılacağı süreyi kullanır.
- Yakın geçmişi yakın gelecek zamana bir tahmin aracı olarak kullanırsak, yakın geçmişte en az kullanılan sayfa gelecekte de kullanılmayacaktır, o halde bu sayfa kurban seçilebilir.





LRU Page Replacement (LRU sayfa yer deđiřimi)

reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2		2		4	4	4	0		1		1		1
	0	0	0		0		0	0	3	3		3		0		0
		1	1		3		3	2	2	2		2		2		7

page frames





Tips for Avoiding Page Faults (Sayfalama hatalarından kaçınma yolları)

■ Program yapısı

- `Int[128,128] data;`
- Her bir satır bir sayfada depolanır.
- Program 1

```
for (j = 0; j < 128; j++)  
    for (i = 0; i < 128; i++)  
        data[i,j] = 0;
```

128 x 128 = 16,384 page faults (sayfalama hatası)

- Program 2

```
for (i = 0; i < 128; i++)  
    for (j = 0; j < 128; j++)  
        data[i,j] = 0;
```

128 page faults (sayfalama hatası)





Thrashing (Boşa Çalışma)

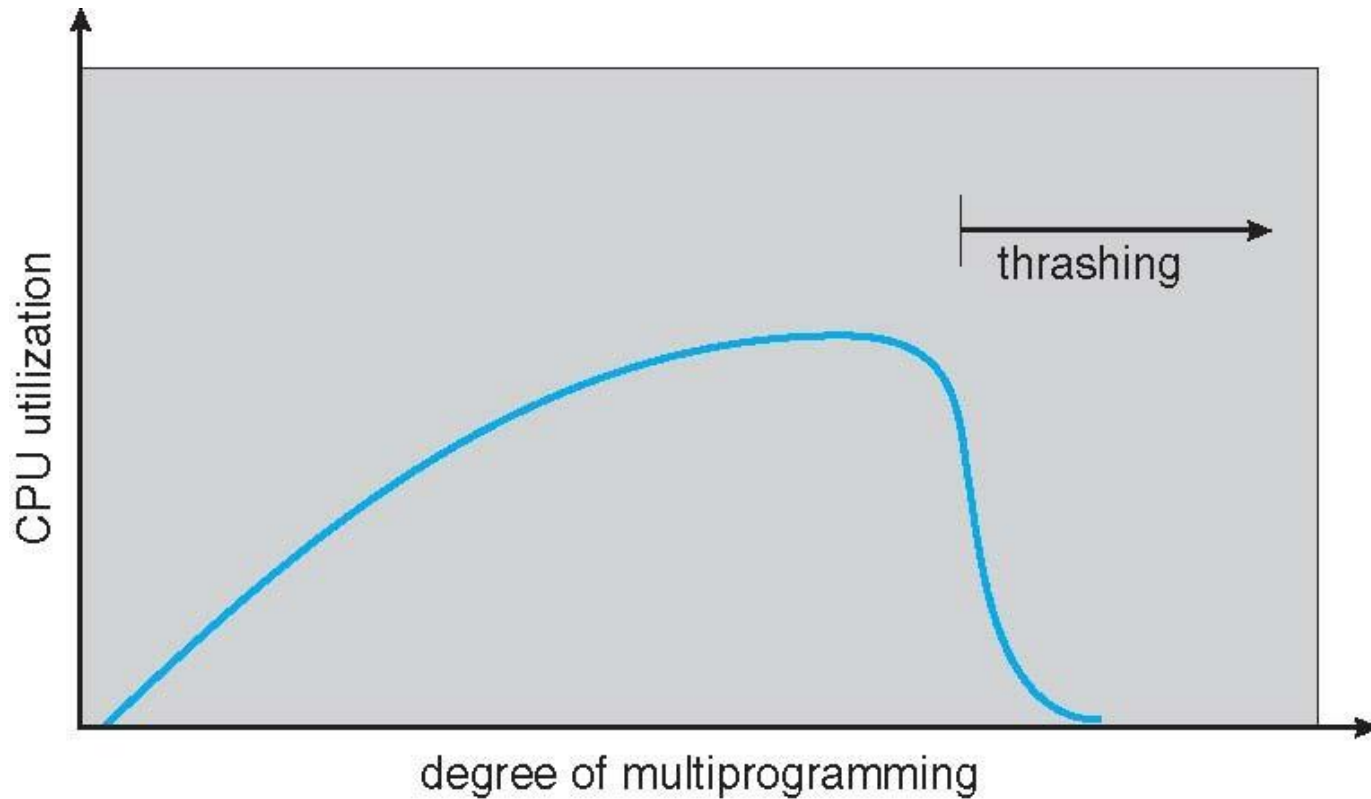
- Eğer işlemin yeteri kadar sayfası yoksa, sayfalama hatası çok yüksek olur. Bu durumda :
 - Düşük CPU kullanımına yol açar.
 - Bir işlem sayfalamaya yürütmeden daha fazla zaman harcıyorsa o işlem boşa çalışıyordur.

- **Thrashing** \equiv Bir işlemin sayfaları durmadan getirilip – götürülmeyle meşgul olması durumudur.





Thrashing (Cont.) (Boşa çalışma devam)





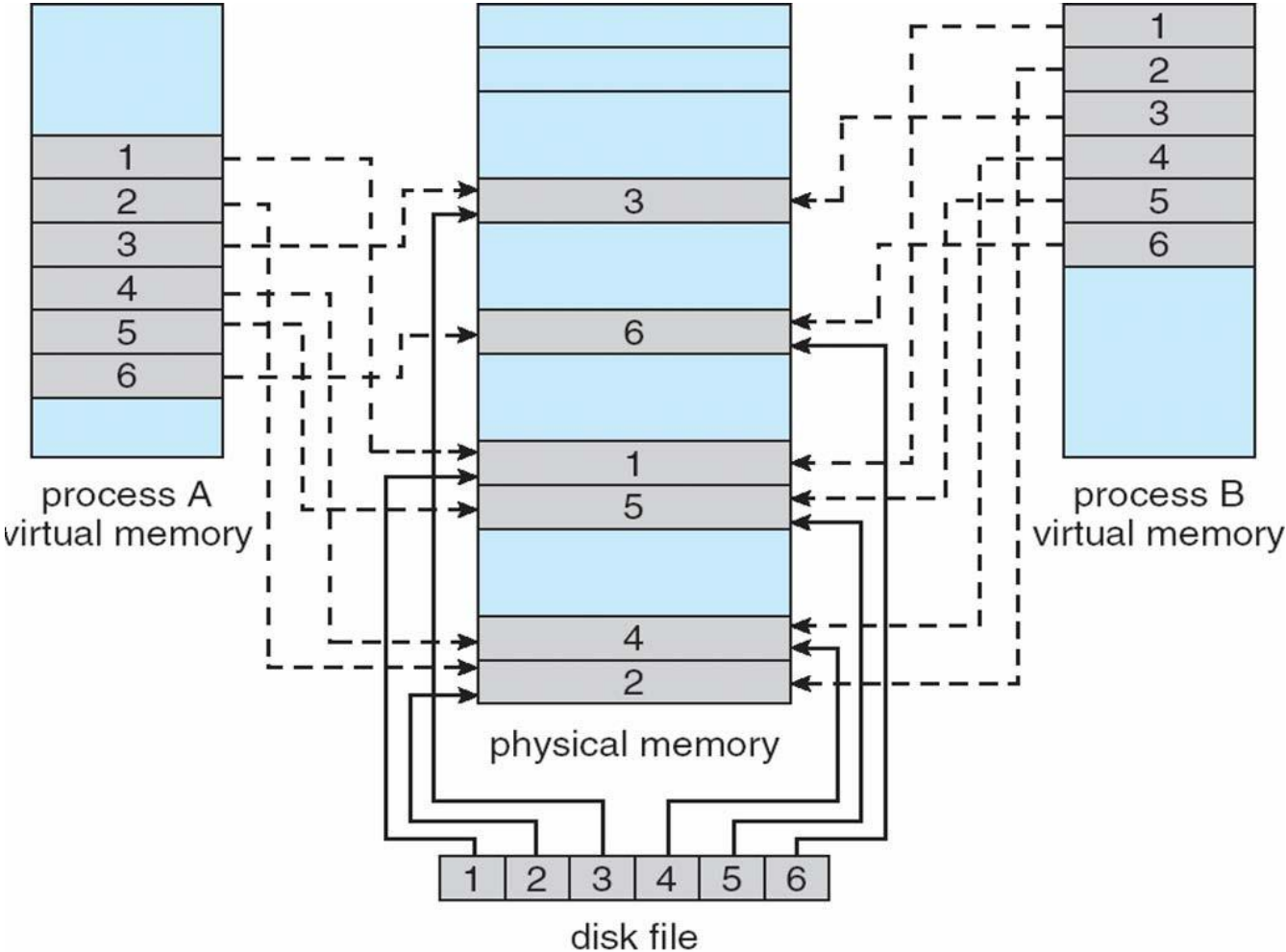
Memory-Mapped Files (Eşleştirilmiş-Bellek Dosyaları)

- Eşleştirilmiş bellek I/O dosyası bir disk bloğunu bellekteki bir sayfayla eşleştirerek I/O dosyasının rutin bellek ulaşımı gibi işlenmesini sağlar.
- Dosyaya giriş çıkış işlemleri `read()` `write()` sistem çağrıları yerine bellek üzerinden (aracılığıyla) işleyerek dosya ulaşımını kolaylaştırır.
- Aynı zamanda bellekteki sayfaların paylaşımına izin vererek bazı işlemlerin aynı dosyayı eşleştirmesine olanak sağlar.





Memory Mapped Files (Eşleştirilmiş – Bellek Dosyaları)





Memory-Mapped Shared Memory in Windows (Windows'ta Elleştirilmiş – Bellek Paylaşımlı Hafıza)

