İş Parçacığı Senkronizasyonu

- İş parçacığı senkronizasyonu (thread synchronization / serialization), iki veya daha fazla eşzamanlı olarak çalışan iş parçacığının işlem adres uzayında ortak kullandıkları kısımdaki yazılım bileşenlerini aynı anda çalıştırmalarına veya üzerlerinde aynı anda değişiklik yapmalarına engel olmak amaçlı belirli mekanizmaların işletilmesidir [Kaynak].

- İşte bu ortak kullanılan program bölümü (program segment) bilgisayar bilimi terminolojisinde kritik bölüm (critical segment) olarak adlandırılır. Bir başka ifade ile kritik bölüm, bir programın seri (serial) olarak işletilmesi gereken bölümüdür.

İş Parçacığı Senkronizasyonu

- İş parçacıklarının kritik bölüme erişimleri senkronizasyon teknikleri ile kontrol edilir. Eğer bir iş parçacığı kritik bölüme erişim durumunda ise diğer iş parçacıkları ilgili iş parçacığı işini bitirmeden kritik bölüme erişemezler.

- Eğer düzgün senkronizasyon teknikleri uygulanmazsa örneğin değişken değerlerinde tahmin edilmesi zor hatalara sebep olabilir.

- Bu durum bilgisayar bilimi terminolojisinde yarış durumu (race condition) kavramı ile ifade edilir.

İş Parçacığı Senkronizasyonu

Thread 1	Thread 2		Integer value
			0
read value		←	0
increase value			0
write back		→	1
	read value	←	1
	increase value		1
	write back	→	2

Kaynak

İş Parçacığı Senkronizasyonu

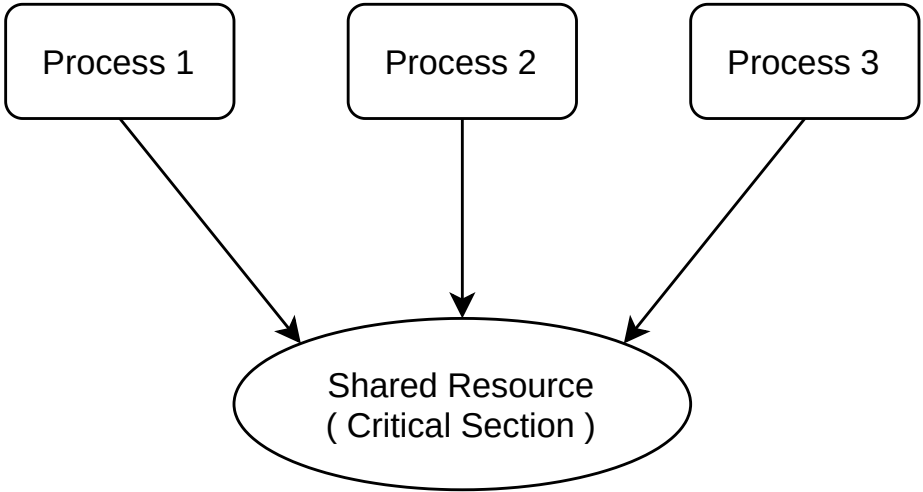
Thread 1	Thread 2		Integer value
			0
read value		←	0
	read value	←	0
increase value			0
	increase value		0
write back		→	1
	write back	→	1

Kaynak

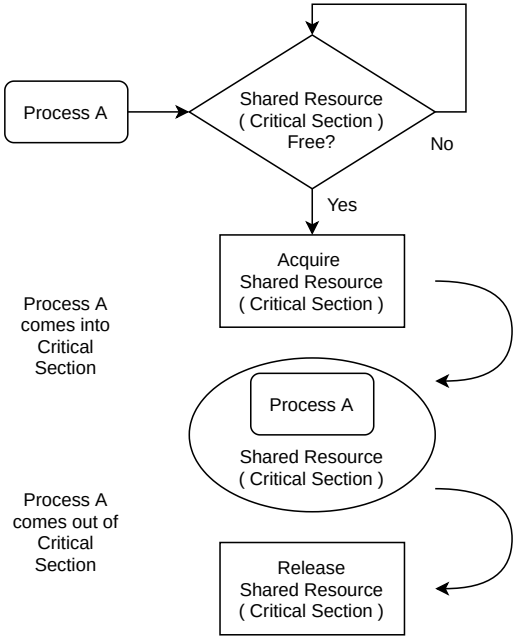
İşlem Senkronizasyonu

- Her ne kadar iş parçacıkları kadar sık olmasa da işlemler de belli kritik bölümlere erişim talebinde bulunabilirler.
- Bu durumda iş parçacıklarında kullanılan senkronizasyon teknikleri işlemler için de kullanılmalıdır.
- Bunun sebebi adres uzayı erişim farkı dışında çalışma prosedürü olarak işlemler ile iş parçacıkları arasında pek bir fark olmamasıdır.

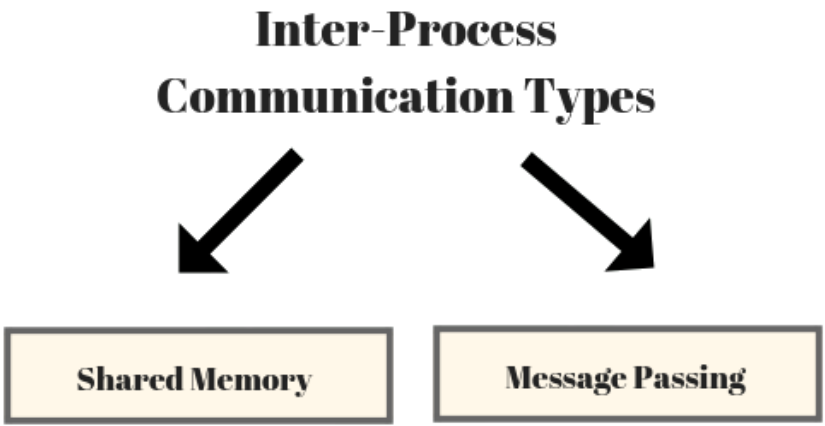
İşlem Senkronizasyonu



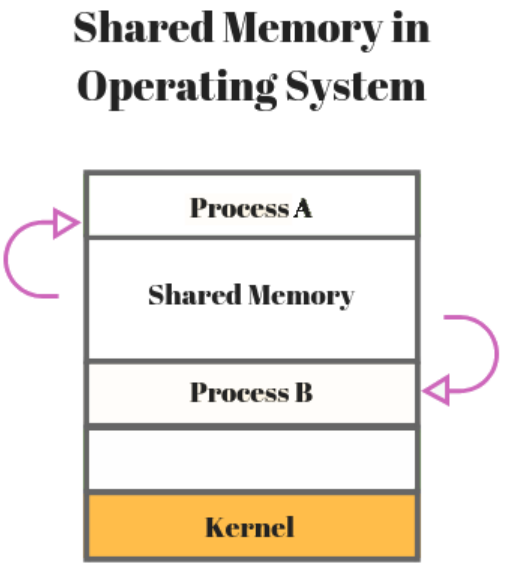
Kaynak


Process A
Shared Resource
(Critical Section)
Free?
No
Yes
Acquire
Shared Resource
(Critical Section)
Process A
comes into
Critical
Section
Process A
Shared Resource
(Critical Section)
Process A
comes out of
Critical
Section
Release
Shared Resource
(Critical Section)

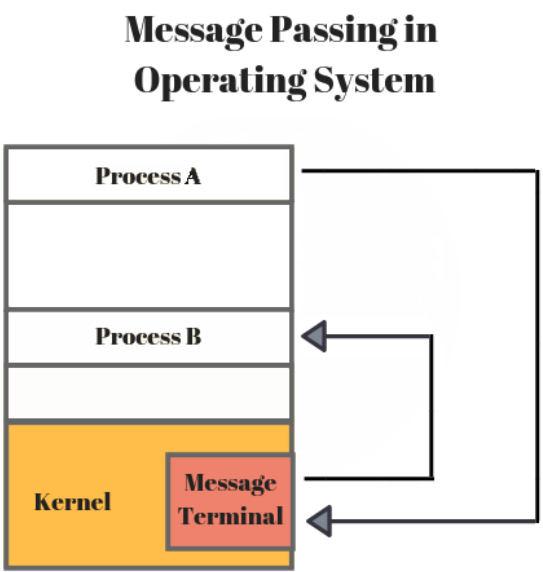
Kaynak


Inter-Process
Communication Types
Shared Memory
Message Passing

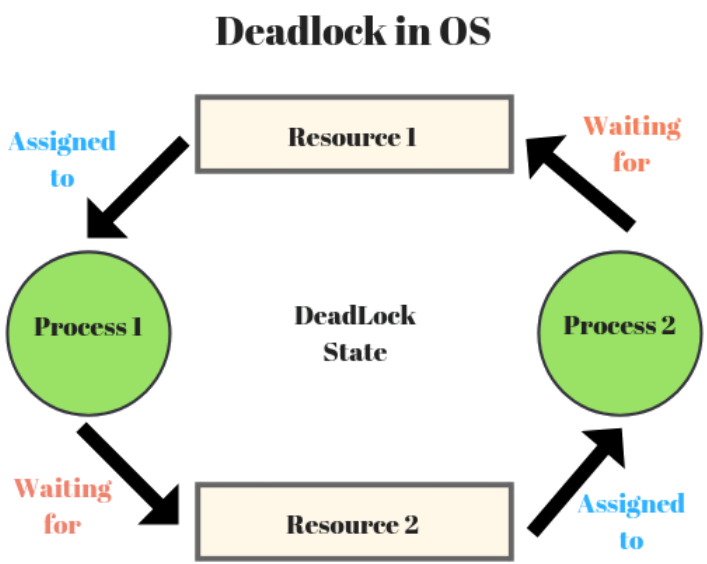
Kaynak


Shared Memory in
Operating System
Process A
Shared Memory
Process B
Kernel

Kaynak


Message Passing in
Operating System
Process A
Process B
Kernel
Message
Terminal

Kaynak


Deadlock in OS
Resource 1
Assigned to
Waiting for
Process 1
DeadLock State
Process 2
Waiting for
Resource 2
Assigned to

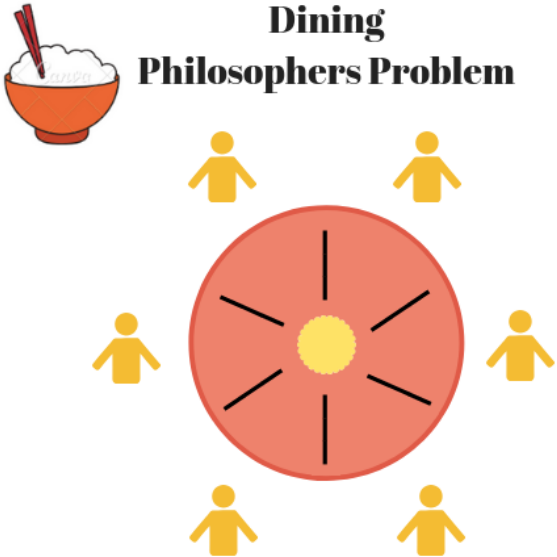
Kaynak

Deadlock Conditions in Operating System

Conditions

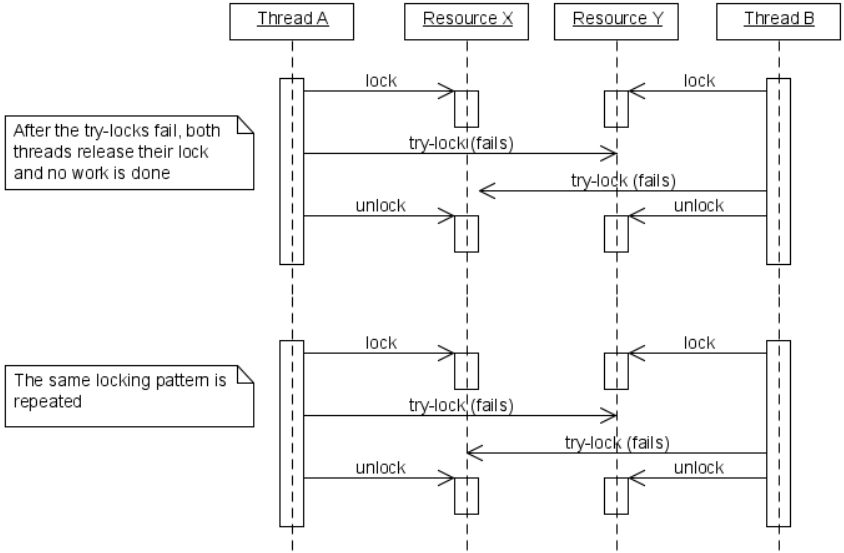
- Mutual Exclusion
- Hold and Wait
- No Preemption
- Circular Wait

Kaynak



Kaynak

Kilitlenme Durumları : Livelock

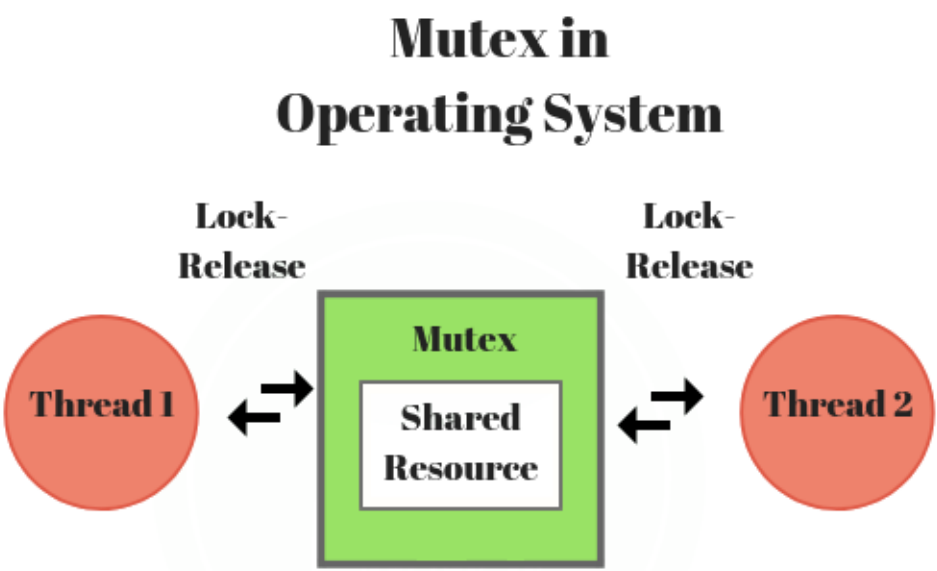


Kaynak

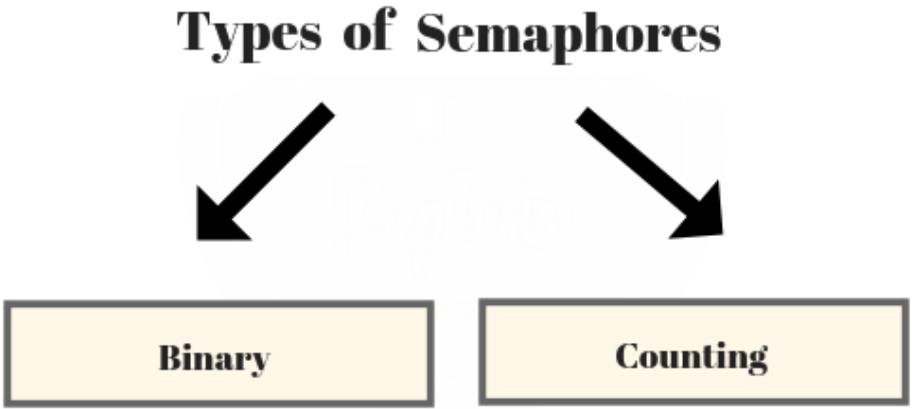
Senkronizasyon Mekanizmaları

1. Mutex (for Threads)

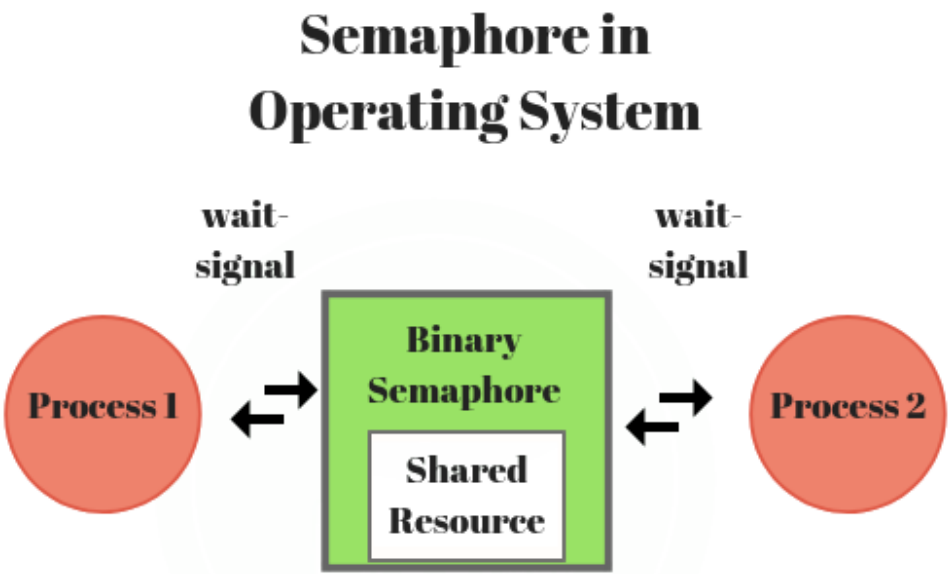
2. Semaphores (for Processes)


Mutex in
Operating System
Lock-
Release
Lock-
Release
Mutex
Shared
Resource
Thread 1
Thread 2

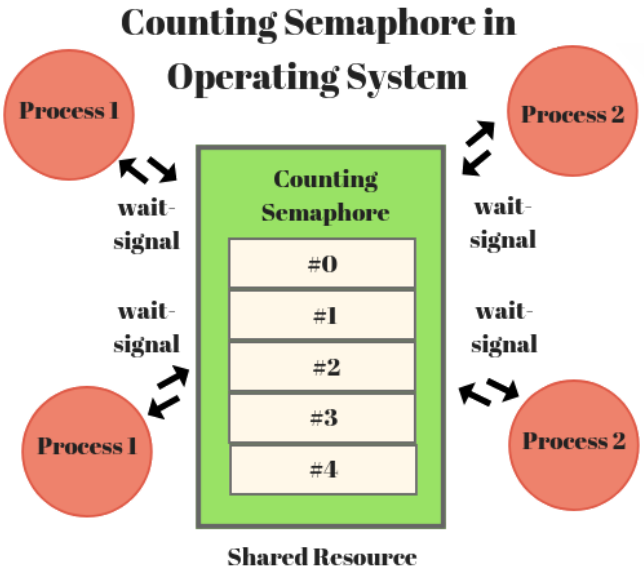
Kaynak



Kaynak


Semaphore in
Operating System
wait-
signal
wait-
signal
Process 1
Binary
Semaphore
Shared
Resource
Process 2

Kaynak


Counting Semaphore in Operating System
Process 1
Process 2
wait-signal
wait-signal
Counting Semaphore
#0
#1
#2
#3
#4
wait-signal
wait-signal
Process 1
Process 2
Shared Resource

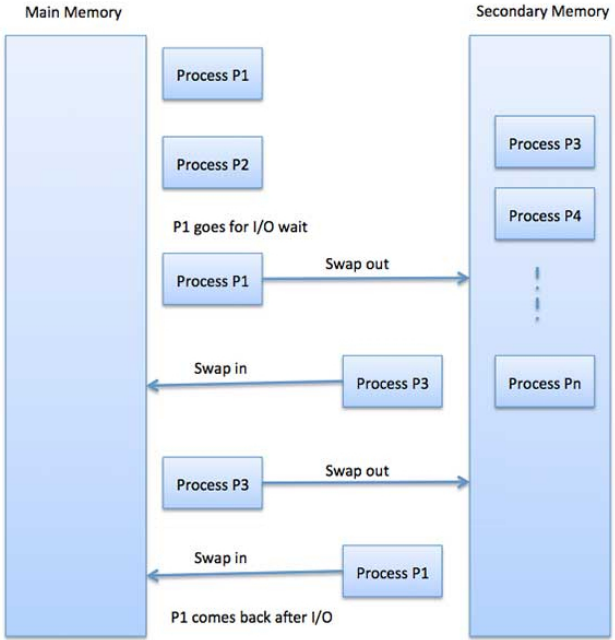
Kaynak

Sanal Hafıza ve Karşılıklı Yer Değiştirme

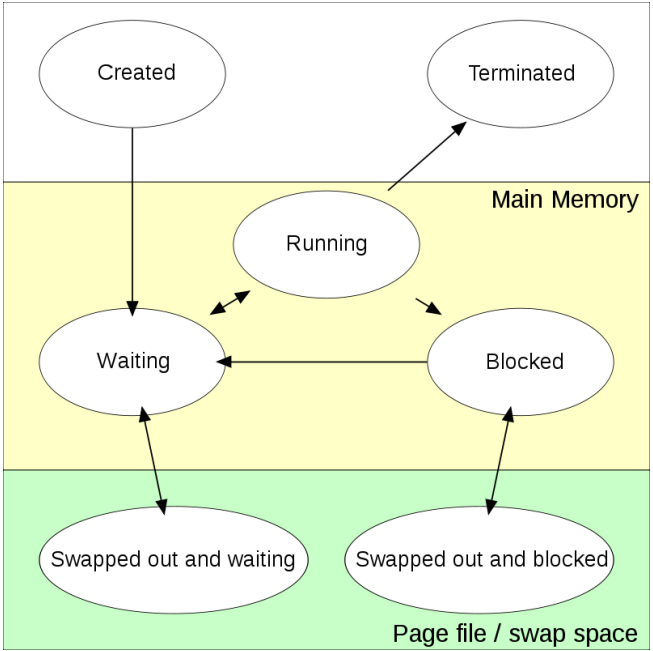
- Çok sayıda işlemin aynı anda RAM de yer işgal etmesi sınırlı RAM kapasitesine sahip bilgisayar sistemlerinde performans düşüşüne sebep olacaktır.

- Bu durumda performansı artırmak için sanal hafıza (virtual memory) olarak adlandırılan hafıza yönetim tekniğinden yararlanılır.

- Bu teknik sayesinde işlemler sabit disk alanını da sistem ana hafızası gibi kullanabilmekte ve sistem ana hafızası ile sabit disk arasında karşılıklı yer değiştirme yani bir başka ifade ile takas (swap) mümkün olmaktadır.

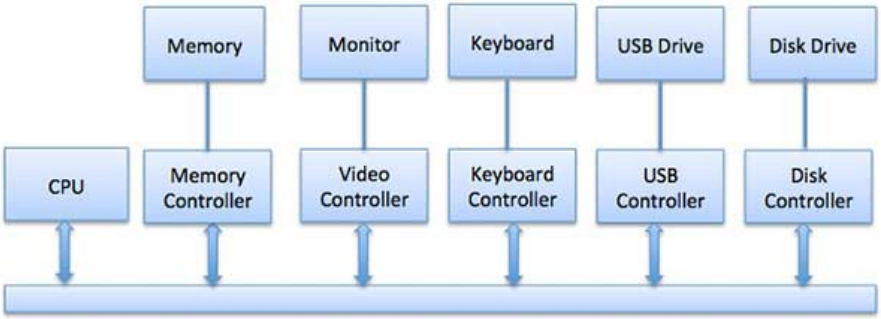

Main Memory
Secondary Memory
Process P1
Process P2
Process P3
Process P4
P1 goes for I/O wait
Process P1
Swap out
Swap in
Process P3
Process Pn
Process P3
Swap out
Swap in
Process P1
P1 comes back after I/O

Kaynak


Created
Terminated
Main Memory
Running
Waiting
Blocked
Swapped out and waiting
Swapped out and blocked
Page file / swap space

Kaynak

- CPU ve aygıt denetleyicileri (device controllers), sistem iletişim yolu (system bus) üzerinden sistem ana hafızasına (main memory) erişim sağlarlar.



Kaynak

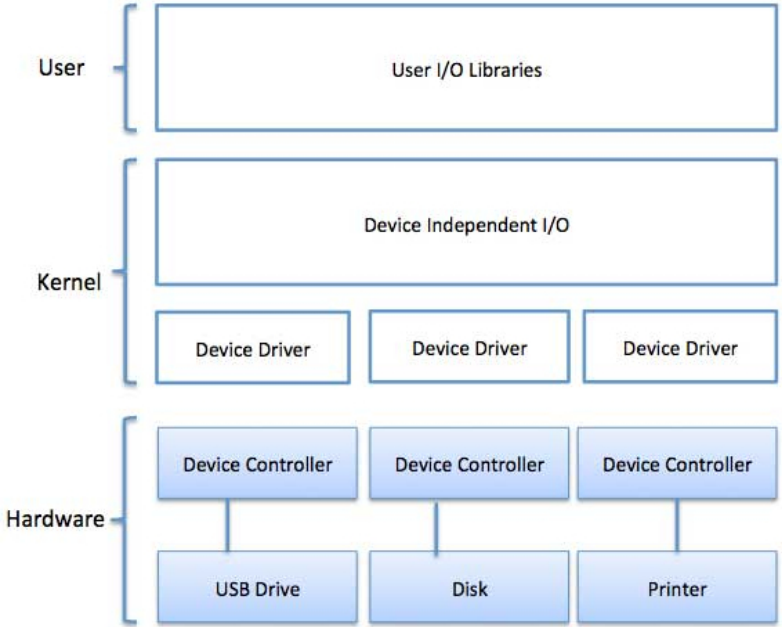
Bilgisayarın Sistem Organizasyonu

- CPU ve giriş/çıkış aygıtları (I/O devices) eşzamanlı olarak çalışabilirler. Herbir aygıt, aygıt denetleyicisi (device controller) vasıtası ile kontrol edilir.
- Aygıt denetleyicisi, aygıt ile işletim sistemi arasında köprü vazifesi görür.
- İşletim sistemi, aygıt denetleyicisi (device controller) ile aygıt sürücüsü (device driver) vasıtasıyla iletişim kurar.
- Aygıt sürücüsü her bir aygıt için özel olarak geliştirilmiş olan bir yazılımdır.

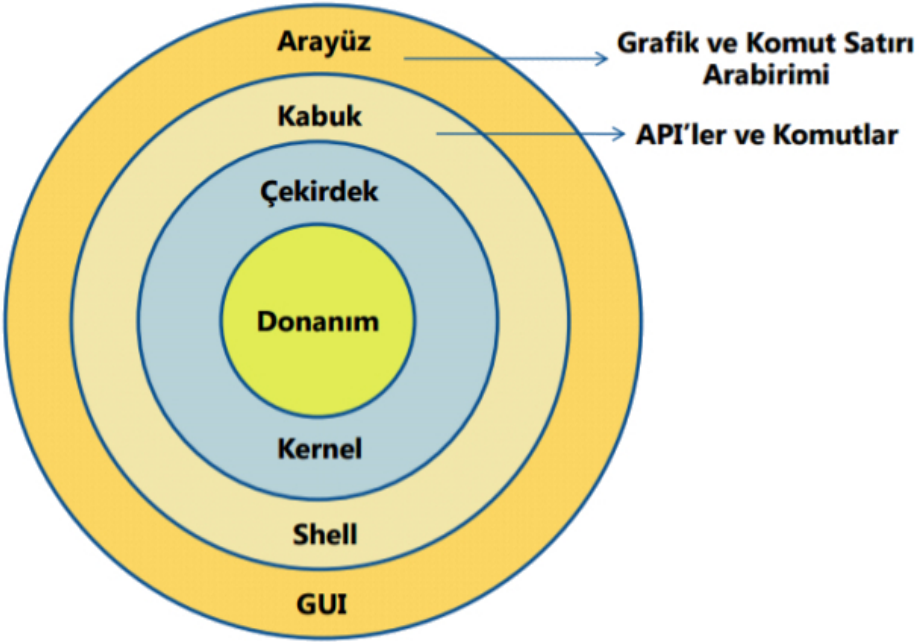
Bilgisayarın Sistem Organizasyonu

- Aygıt sürücüsü, aygıt denetleyicisi ile işletim sistemi arasındaki bir yazılım arayüzüdür.

- İşletim sistemi, sistem yazılımları ve diğer uygulama programlarının bir donanım aygıtı ile, o donanım aygıtı hakkında detaylı bilgiye sahip olmadan aygıt sürücüsü vasıtasıyla iletişim kurabilmesini sağlar.


User
User I/O Libraries
Kernel
Device Independent I/O
Device Driver
Device Driver
Device Driver
Hardware
Device Controller
Device Controller
Device Controller
USB Drive
Disk
Printer

Kaynak


Arayüz
Grafik ve Komut Satırı Arabirimi
Kabuk
API'ler ve Komutlar
Çekirdek
Donanım
Kernel
Shell
GUI

Kaynak

İşletim Sistemi Çekirdeği (Kernel)

- İşletim sisteminin her ne kadar evrensel kabul görmüş bir tanımı olmasa da satın aldığınız bir bilgisayarla beraber sunulan sistem yazılımının tamamı olarak da ifade edilebilir.

- İşte bu bilgisayar ile beraber sunulan işletim sistemi yazılımında, bilgisayar aktifken daima çalışan program, çekirdek (kernel) olarak adlandırılır.

- Çekirdek (kernel), işletim sisteminin en temel ve olmazsa olmaz parçasıdır ve işletim sisteminin esasını teşkil eder.

İşletim Sistemi Çekirdeği (Kernel)

- En temel görevi, yazılım ile donanım birimlerinin haberleşmesini sağlamaktır.
- Ayrıca tüm bilgisayarın ve bilgisayardaki yazılım uygulamalarının düzgün çalışabilmesi için gerekli hizmetleri sağlar.
- Bu hizmetlerden bazıları şunlardır :

İşletim Sistemi Çekirdeği (Kernel)

- Program Çalıştırma (Program Execution)
- Programlara Bellek Tahsisi (Memory Allocation)
- İşlem Yönetimi (Process Management)
- Hafıza Yönetimi (Memory Management)
- Görev Planlama (Task Scheduling)

İşletim Sistemi Çekirdeği (Kernel)

- Dosya Yönetimi (File Management)
- Giriş/Çıkış Aygıtları Yönetimi (I/O Device Management)
- Sistem çağrılarına (System Calls) cevap vermek
- Bütün donanım birimlerine erişim
- Bütün yazılım birimlerini çalıştırmak

İşletim Sistemi Çekirdeği (Kernel)

- Çekirdek (kernel) ile beraber pek çok uygulamanın birleştirilerek bir paket halinde sunulması, işletim sistemi (operating system) olarak adlandırılır.
- İşletim sisteminde çekirdek dışındaki programlar, ya işletim sistemiyle beraber sunulan sistem yazılımlarıdır ya da uygulama yazılımlarıdır.
- Çekirdek, uygulama yazılımları ile donanımlar arasındaki bağlantıyı sağlar.
- Donanımlara, kullanıcıların ve uygulama yazılımlarının doğrudan erişimlerini sınırlandırır ve düzenler.

İşletim Sistemi Çekirdeği (Kernel)

- İşletim sistemi çekirdeğinin üç tipi vardır [Kaynak] :

1. Monolitik Çekirdekler (Monolithic Kernel)
2. Mikro Çekirdekler (Micro Kernel)
3. Hibrit Çekirdekler (Hybrid Kernel)

İşletim Sistemi Çekirdeği : Monolithic Kernel

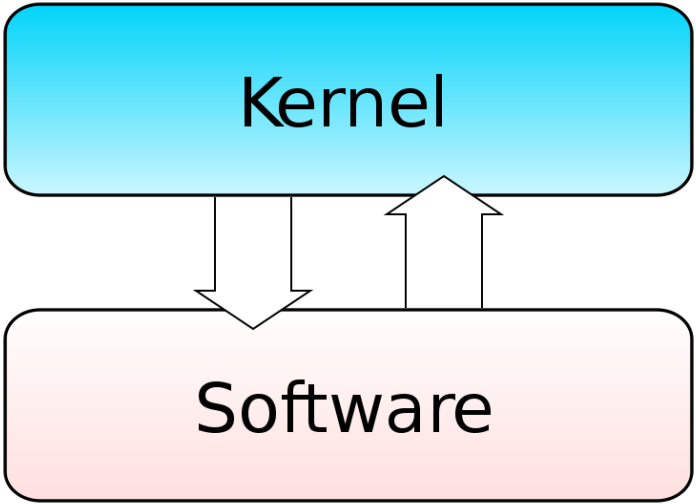
- Monolitik çekirdekler (monolithic kernel), 1970–1990 yılları arasında kullanılan ilk çekirdeklerdir.
- Monolitik modelde tüm yazılımlar (görev yöneticisi, dosya yöneticisi, hafıza yöneticisi vb.) ve aygıt sürücüleri işletim sisteminin çekirdeğinde yer almaktadır.
- Monolitik çekirdekler, boyut olarak büyük olmalarına karşın her tür fonksiyonu aynı adres uzayında içerdikleri için hızlıdırlar ve tasarımları kolaydır.

İşletim Sistemi Çekirdeği : Monolithic Kernel

- Hız ve tasarım avantajlarına karşın monolitik çekirdeklerin dezavantajları da vardır. Örneğin bir aygıt sürücüsündeki hata tüm sistemin çökmesine (crash) sebep olabilir.

- Dolayısı ile monolitik çekirdeklerin çökme tehlikesine karşı güven vermeyen (crash insecure) bir yapısı vardır.

- Bunun dışında bakımları zordur ve sistem kararlılığı (stability) konusunda sıkıntılar yaşanabilir.



Kaynak

İşletim Sistemi Çekirdeği : Monolithic Kernel

- Avantajları
  - Performansları yüksektir.
  - Tasarımları kolaydır.
- Dezavantajları
  - Sistem kararlılıkları göreceli olarak düşüktür.
  - Çökme güvenlikleri göreceli olarak düşüktür.

İşletim Sistemi Çekirdeği : Monolithic Kernel

- Monolitik Çekirdek Örnekleri [Kaynak 1 , Kaynak 2]
  - Geleneksel Unix Çekirdekleri
  - MS-DOS, Windows 9x Serisi
  - Mac OS (Sürüm 8.6 ve öncesi)
  - Linux, FreeBSD, OpenBSD, Solaris

İşletim Sistemi Çekirdeği : Micro Kernel

- Monolitik çekirdeklerin boyutlarının çok büyük olması, sistem kararlılığı (stability) ve çökmeye karşı güvende olma (crash secure) konusundaki endişeler vb. sebeplerle modüler yapıda olan mikro çekirdekler (micro kernel) geliştirilmiştir.

- Mikro çekirdeklerin modüler yapısından dolayı aygıt sürücüleri ve benzeri hizmetler çekirdeğin bir parçası olmaktan çıkmıştır. Bu durum sistem kararlılığını ve çökme güvenliğini artırmıştır.

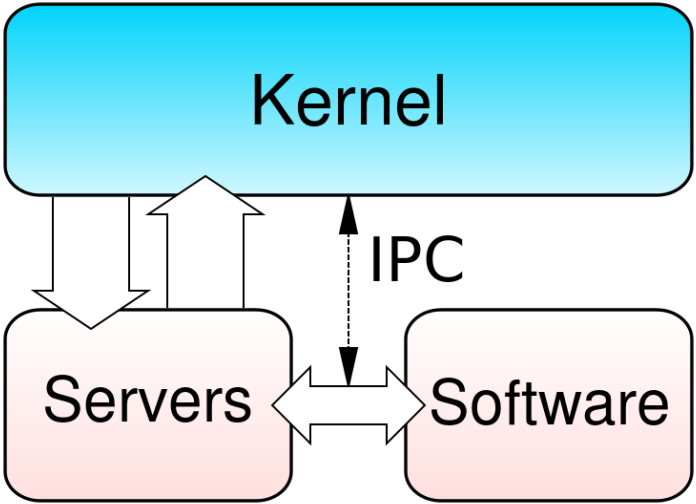
- Mikro çekirdekler, sadece en önemli işletim sistemi fonksiyonlarını içerdikleri için oldukça küçük boyutta olmaktadır.

İşletim Sistemi Çekirdeği : Micro Kernel

- Mikro çekirdekte pek çok bileşen modül şeklinde olduğundan, örneğin, bir aygıt sürücüsünde sorun oluşsa ilgili aygıt sürücüsü modül yeniden başlatılarak sorun giderilebilir.

- Kararlılık ve küçük kod boyutu avantajlarına rağmen mikro çekirdek sistemlerin dezavantajları da vardır.

- Örneğin çekirdek ile aygıt sürücüleri ve benzeri fonksiyonların farklı adres uzayını kullanması kernel ve driver veya diğer process'ler arasındaki iletişimde ekstra yük getirir ve bu durum performans düşüşüne sebep olur.



Kaynak

İşletim Sistemi Çekirdeği : Micro Kernel

- Avantajları
  - Sistem kararlılıkları yüksektir.
  - Çökme güvenlikleri yüksektir.
- Dezavantajları
  - Performansları göreceli olarak düşüktür.
  - Tasarımları, ekstra mekanizmaların (context switch vb.) işletilmesi gerektiğinden zordur.

İşletim Sistemi Çekirdeği : Micro Kernel

- Mikro Çekirdek Örnekleri [Kaynak 1 , Kaynak 2]
  - GNU OS
  - QNX
  - Minix
  - Mach

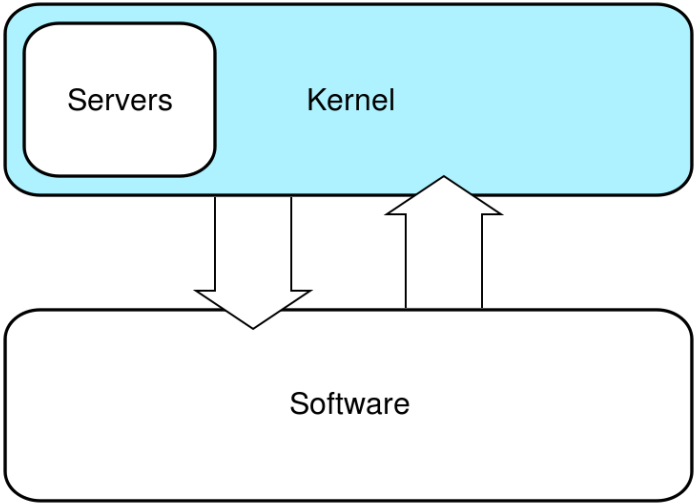
İşletim Sistemi Çekirdeği : Hybrid Kernel

- Performans ve sistem kararlılığının bilgisayar sistemlerinde oldukça önem arz etmesi monolitik ve mikro çekirdeklerin avantajlarına sahip olan hibrit çekirdeklerin (hybrid kernels) geliştirilmesini kaçınılmaz kılmıştır.

- Hibrit çekirdekler modüler yaklaşımla tasarlanmış monolitik çelirdekler olarak da düşünülebilir.

- İdealde her iki çekirdeğinin avantajlarına sahip olup dezavantajlarına sahip olmaması beklense de bu durum tasarımın ne kadar düzgün yapıldığına göre değişiklik gösterir.

İşletim Sistemi Çekirdeği : Hybrid Kernel



Kaynak

İşletim Sistemi Çekirdeği : Hybrid Kernel

- Hibrit Çekirdek Örnekleri [Kaynak 1 , Kaynak 2]
  - NT kernel (Windows NT, 2000, XP, Vista, 7, 8, and 10)
  - XNU (Mac OS X and iOS kernel)
  - DragonFly BSD