


- ▶ Donanımsal Yazılım (Firmware)
  - ▶ Basic Input Output System (BIOS) : Legacy, UEFI
- ▶ Sistem Yazılımları (System Software)
  - ▶ Bootloaders : BootMgr (Windows), BootX (Apple), Grub (Linux)
  - ▶ Operating Systems : Windows, Mac OS, Linux, Android, iOS
  - ▶ Translators : Assembler, Compiler, Interpreter

- ▶ Yardımcı Yazılımlar (Utility Software)
  - ▶ Task Manager, Network Manager
- ▶ Uygulama Yazılımları (Application Software)
  - ▶ Word, PowerPoint, Excel
  - ▶ Edge, Chrome, Firefox
  - ▶ Adobe Photoshop, GIMP

- ▶ **Temel Giriş Çıkış Sistemi** (**Basic Input Output System** - BIOS), bilgisayarı açtıktan sonra sistemdeki donanımları ve sistemin başlatılması ile ilgili ayarları denetleyen yazılımdır.
- ▶ BIOS'un sistemin başlatılması ile ilgili kısmı **bootstrap** olarak adlandırılır. BIOS, genel amaçlı bilgisayar sistemlerinde (örneğin masaüstü, laptop vb.) anakart üzerindeki E2PROM belleklerde yer alır.
- ▶ E2PROM belleklerde yer alan ve sistem açılışından sorumlu olan BIOS tarzı programlar **firmware** olarak adlandırılır.

- ▶ BIOS görevini tamamlamasının akabinde sistem kontrolünü, disk (HDD veya SSD)'in bir bölümünde (**partition**) bulunan **bootloader** programına devreder.
- ▶ **Bootloader** programının görevi, disk (HDD veya SSD)'de kurulu olan işletim sistemini RAM belleğe yüklemektir.
- ▶ Dolayısı ile **BIOS**, sistemin tüm yönleri ile başlatılmasından; **bootloader** ise işletim sistemi çekirdeği (**kernel**)'nin ana belleğe yüklenip çalıştırılmasından sorumludur.

- ▶ Bilgisayar açılır (Power On )
- ▶ CPU, E2PROM'da yer alan BIOS'u (**firmware**) çalıştırır.
- ▶ BIOS'da **bootstrap** ve diğer fonksiyonlar işletilir.
- ▶ BIOS, işletim sistemi kurulumu esnasında yapılan disk bölümlleme (**disk partition**) sonucu oluşan bölümlleme tablosu (**partition table**)'nu bulur.

- ▶ BIOS, bölümlene tablosunun içeriğini **bootloader**'a aktarır.
- ▶ Bootloader, bölümlene tablosunun içeriğinde birden fazla işletim sistemi tesbit ederse kullanıcıdan hangi işletim sistemini başlatmak istediğini seçmesini ister.
- ▶ Bootloader tesbit edilen/seçilen işletim sistemini RAM'e yükler ve bir müddet sonra oturum açma (**log on**) ekranı görülür.
- ▶ Daha önceden oluşturulmuş bir kullanıcı hesabı (**user account**) ile oturum açılır.

- ▶ İşletim sistemi, bir bilgisayar sisteminde kullanıcı ile etkileşim kurarak, bilgisayardaki donanım ve yazılım kaynaklarının bu kaynakları talep edenler arasında paylaşılmasını sağlayan bir sistem yazılımıdır.
- ▶ Bilgisayar donanımı ile kullanıcı arasında bir aracı görevi görür. Bilgisayar donanımının yöneticisi olarak da düşünülebilir.
- ▶ Uygulama yazılımlarının çalışması için gerekli ortamı oluşturur ve donanım birimlerinden uygulamalardan gelen talepler doğrultusunda faydalanılmasını sağlar.

## 1 Donanım (Hardware) :

Temel işlem/hesaplama gereksinimlerini karşılayan kaynaklardır.

Örneğin CPU, RAM, ROM, SSD, HDD, Ekran Kartı vb.

## 2 İşletim Sistemi (Operating System) :

Donanımın çeşitli uygulamalar ve kullanıcılar tarafından kullanımının koordinasyonu ve kontrolünden sorumludur.

Örneğin Windows 10, Linux Dağıtımları, Mac OS X, Android, iOS vb.



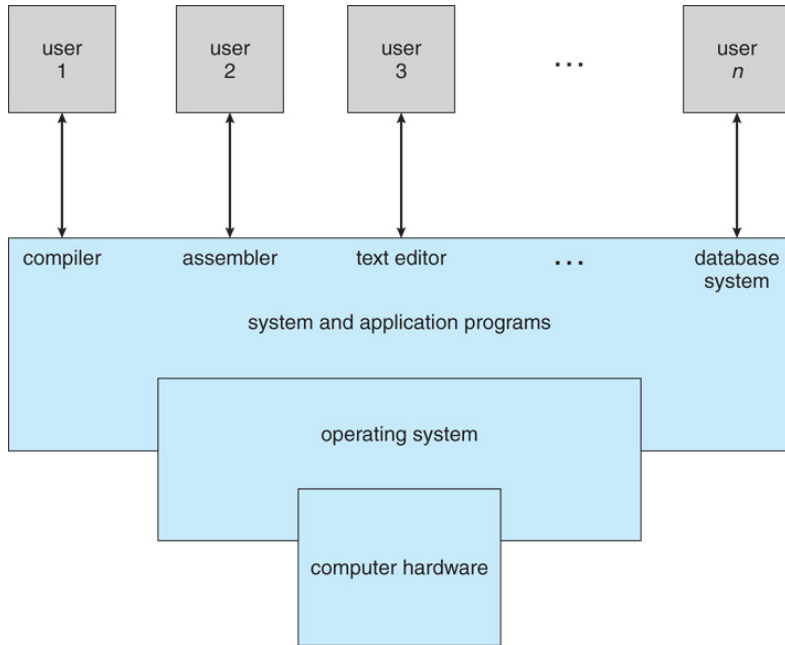
## 3 Sistem ve Uygulama Programları (System and Application Programs) :

Kullanıcıların yapmak istediği işler için gerekli kaynakların kullanım prosedürünü tanımlar.

Örneğin MS Word, IE, Chrome, Acrobat Reader, Skype, Notepad vb.

## 4 Kullanıcılar (Users) :

İnsanlar, makineler veya diğer bilgisayarlar.



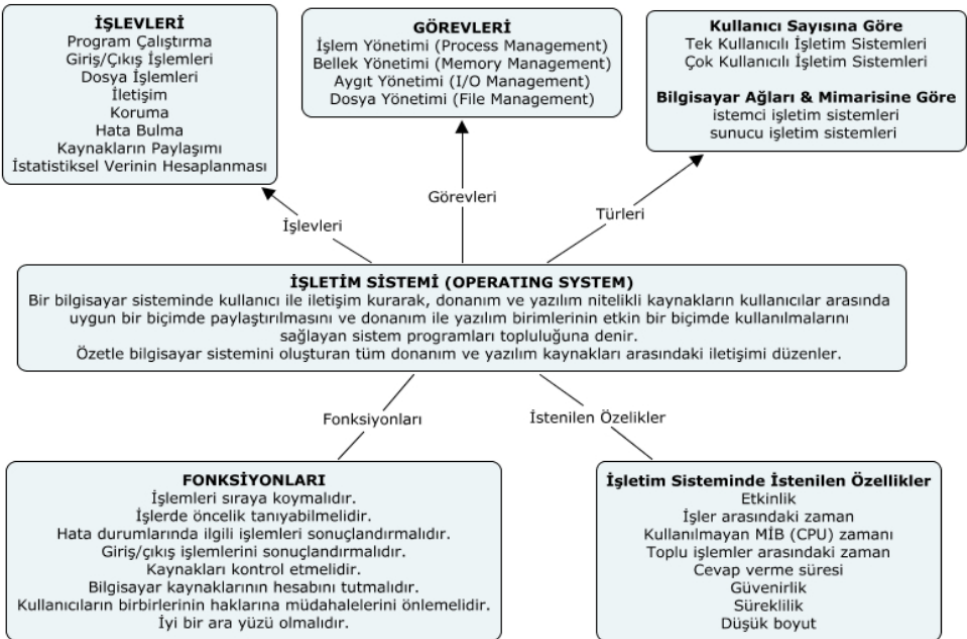
- ▶ İşletim sistemi, bilgisayardaki yazılım, donanım ve bunların işleyeceği verilerin uygun ve verimli bir şekilde kullanımını sağlayan bir platformdur.
- ▶ İşletim sistemi, bilgisayardaki bütün birimlerin etkili bir iletişim kurmasını sağlayan ortamı oluşturur.
- ▶ İşletim sisteminin görevi, çok sayıda birimden (yazılım) gelen kaynak taleplerini (donanım) birbirleri ile çakışmadan karşılamaktır.

- ▶ İşletim sistemi bu gelen taleplere binaen kaynakların talep edenlere etkili ve verimli bir şekilde tahsisinden sorumludur.
- ▶ İşletim sistemi dizaynında **kullanım kolaylığı (kullanışlılık)**, **yüksek performans (verimlilik)** ve **kaynakların özel kullanımı** önemli parametrelerdir.
- ▶ **Kullanışlılık**, bilgisayar kullanıcısına sağlanan kolaylığı ve kullanıcının donanım birimleriyle kolay etkileşim sağlamasını ifade eder.

- ▶ **Verimlilik**, bilgisayar donanımlarının en uygun, tercih edilir ve etkin bir şekilde yüksek performans sağlayarak kullanımını ifade eder.
- ▶ **Kaynakların özel kullanımı**, gelen kaynak taleplerine binaen ilgili kaynakların izole edilmesini ifade eder.
- ▶ İşletim sistemi, bir işlemin (aktif çalışan bir program) diğer bir işleme ait kaynaklara olan müdahalesine izin vermez.

- ▶ Ayrıca işlemlere ait bilgilerin silinmesine izin vermeyen bir koruma mekanizması işlevi görür.
- ▶ Böylece işlemlerin birbirlerini veya işletim sistemini modifiye etmesi engellenmiş olur.
- ▶ Dolayısı ile her işletim sisteminin, kaynaklara ulaşımı sağlamak ve bu kaynakları talep eden işlemleri yönetmek için kullandığı stratejiyi belirleyen bir güvenlik politikası vardır.

- ▶ İşletim sistemi bir kaynak tahsis edicidir.
  - ▶ Bütün kaynakları yönetir.
  - ▶ Çakışan istekler arasında etkili kaynak kullanımı ve verimliliği de göz önüne alarak karar verir.
  
- ▶ İşletim sistemi bir kontrol programıdır.
  - ▶ Hataları ve bilgisayarın bilinçsizce kullanımını önlemek amacıyla bütün programların çalışmasını kontrol eder.





## Categories of Operating Systems

Category	Operating System Name
<b>Desktop</b>	<ul style="list-style-type: none"><li>• DOS</li><li>• Early Windows versions (Windows 3.x, Windows 95, Windows NT Workstation, Windows 98, Windows 2000 Professional, Windows Millennium Edition, Windows XP)</li><li>• Windows Vista</li><li>• Mac OS X</li><li>• UNIX</li><li>• Linux</li></ul>
<b>Server</b>	<ul style="list-style-type: none"><li>• Early Windows Server versions (Windows NT Server, Windows 2000 Server, Windows Server 2003)</li><li>• Windows Server 2008</li><li>• UNIX</li><li>• Linux</li><li>• Solaris</li><li>• NetWare</li></ul>
<b>Embedded</b>	<ul style="list-style-type: none"><li>• Windows Embedded CE</li><li>• Windows Mobile</li><li>• Palm OS</li><li>• iPhone OS</li><li>• BlackBerry</li><li>• Embedded Linux</li><li>• Symbian OS</li></ul>

## Categories of Operating Systems

Category	Operating System Name
<b>Desktop</b>	<ul style="list-style-type: none"><li>• DOS</li><li>• Early Windows versions (Windows 3.x, Windows 95, Windows NT Workstation, Windows 98, Windows 2000 Professional, Windows Millennium Edition, Windows XP)</li><li>• Windows Vista</li><li>• Mac OS X</li><li>• UNIX</li><li>• Linux</li></ul>
<b>Server</b>	<ul style="list-style-type: none"><li>• Early Windows Server versions (Windows NT Server, Windows 2000 Server, Windows Server 2003)</li><li>• Windows Server 2008</li><li>• UNIX</li><li>• Linux</li><li>• Solaris</li><li>• NetWare</li></ul>
<b>Embedded</b>	<ul style="list-style-type: none"><li>• Windows Embedded CE</li><li>• Windows Mobile</li><li>• Palm OS</li><li>• iPhone OS</li><li>• BlackBerry</li><li>• Embedded Linux</li><li>• Symbian OS</li></ul>

## Categories of Operating Systems

Category	Operating System Name
<b>Desktop</b>	<ul style="list-style-type: none"><li>• DOS</li><li>• Early Windows versions (Windows 3.x, Windows 95, Windows NT Workstation, Windows 98, Windows 2000 Professional, Windows Millennium Edition, Windows XP)</li><li>• Windows Vista</li><li>• Mac OS X</li><li>• UNIX</li><li>• <b>Linux</b> <b>Ders Kapsamı</b></li></ul>
<b>Server</b>	<ul style="list-style-type: none"><li>• Early Windows Server versions (Windows NT Server, Windows 2000 Server, Windows Server 2003)</li><li>• Windows Server 2008</li><li>• UNIX</li><li>• Linux</li><li>• Solaris</li><li>• NetWare</li></ul>
<b>Embedded</b>	<ul style="list-style-type: none"><li>• Windows Embedded CE</li><li>• Windows Mobile</li><li>• Palm OS</li><li>• iPhone OS</li><li>• BlackBerry</li><li>• Embedded Linux</li><li>• Symbian OS</li></ul>

- ▶ İşletim sistemi terminolojisinde çalışmakta olan bir program (yani RAM belleğe yüklenmiş olan bir program) **işlem** (**process**) olarak adlandırılır.
- ▶ Program, **pasif** bir varlıktır fakat işlem (process), **aktif** bir varlıktır. Tüm işlemler (yani çalışan tüm programlar) **işletim sistemi çekirdeği** (**kernel**) tarafından yönetilir.
- ▶ Bir process başlatıldığı zaman işletim sistemi bu process için **işlem kontrol bloğu** (**process control block**) adı altında bir veri yapısı tanımlar.

- Bir işlem kontrol bloğu, ilgili işlem hakkında işletim sistemi tarafından takip edilmesi gereken bütün bilgiyi ihtiva eder.

Process ID
State
Pointer
Priority
Program counter
CPU registers
I/O information
Accounting information
etc....

- ▶ **Process ID** kısmı, işletim sistemi tarafından işlem (process) için oluşturulmuş spesifik bir işlem numarasını ifade eder.
- ▶ **State** kısmı, işlem (process) için oluşturulmuş durum diyagramındaki ilgili durumu ifade eder. Örneğin oluşturuldu, çalışıyor, bekliyor, bloke edildi, sonlandırıldı gibi process için mevcut durum bilgisini ihtiva eder.
- ▶ **Pointer** kısmı, mevcut işlemi oluşturan ana işlemin hafızadaki adresini gösteren pointer değişkenini ifade eder.

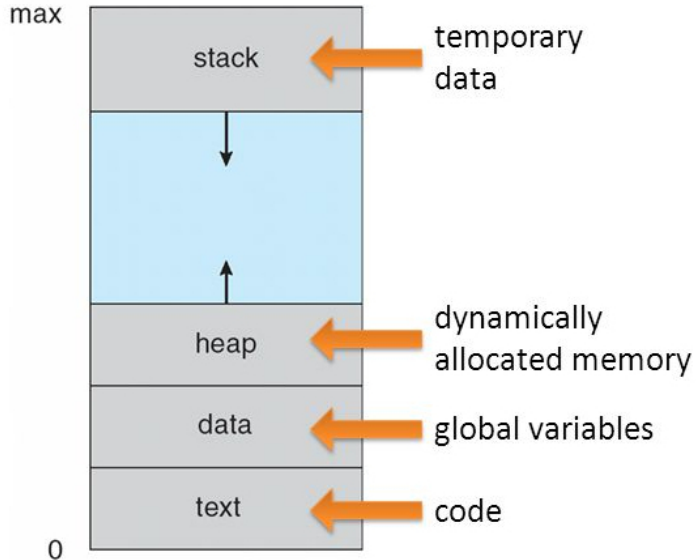
- ▶ **Priority** kısmı, işlemin herhangi bir ayrıcalığa sahip olup olmadığı bilgisini ihtiva eder. Ayrıca CPU zaman planlaması ile ilgili pointer bilgisi de burada tutulur.
- ▶ CPU'nun çok sayıda bekleyen işlemler arasında etkili ve verimli bir şekilde tahsisi **CPU zaman planlaması (CPU scheduling)** olarak adlandırılır.
- ▶ **Program counter** kısmı, ilgili işlem için bir sonraki çalıştırılacak komutun hafızadaki adresini gösteren program sayacını ifade eder. Program sayacı (program counter) bir pointer değişkenidir.

- ▶ **CPU registers** kısmı, işlemin çalışması ile ilgili verinin tutulması gereken CPU kaydedicileri (CPU registers) bilgisini tutan kısımdır.
- ▶ İlgili CPU kaydedicileri (CPU registers) ve program sayacı (program counter) içeriği, işlemler CPU tarafından sırayla çalıştırılırken ve işlemler arası değişim olurken kaydedilmeli ve tekrar yüklenmelidir.
- ▶ Bu durum bilgisayar bilimi terminolojisinde içerik değiştirme (**context switch**) olarak adlandırılır.



- ▶ **I/O information** kısmı, ilgili process için tahsis edilmiş giriş-çıkış aygıtlarının (I/O devices) listesini ihtiva eder.
- ▶ **Accounting information** kısmı, işlemi çalıştırmak için o ana kadarki CPU kullanım zamanı ve işlemin çalışması ile ilgili extra bilgileri ihtiva eden kısımdır.

# İşlemin Ana Hafızadaki Yerleşim Planı



Kaynak

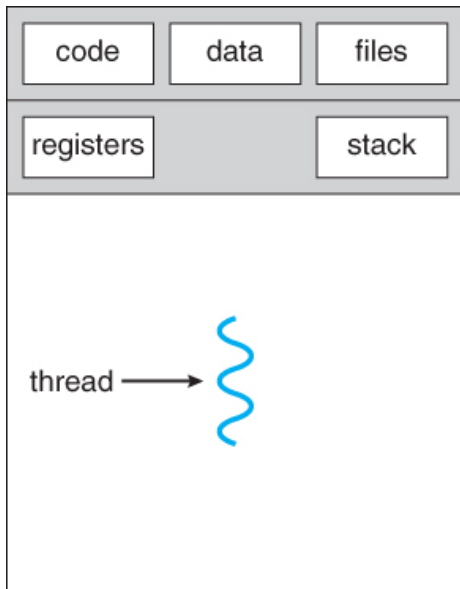
- ▶ **Text** kısmı, program başlatıldığı zaman akışkan olmayan (**non-volatile**) hafızadan (**HDD** veya **SSD**) okunan **derlenmiş** program kodunu ihtiva eder.
- ▶ **Data** kısmı, program kodunda tanımlanmış olan statik ve global değişkenleri ihtiva eder.
- ▶ **Heap** kısmı, dinamik hafıza tahsisi (**dynamic memory allocation**) için kullanılan kısımdır ve **calloc**, **malloc**, **free**, **new**, **delete** gibi yapılara olan çağrılarını ihtiva eder.

- ▶ **Stack** kısmı, programda tanımlanan lokal değişkenler için kullanılır. **Yığın** olarak da adlandırılır.
- ▶ Lokal değişkenler tanımlanır tanımlanmaz stack kısımda yerini alır. Örneğin bir **fonksiyon çağırısı** (**function call**) yapıldığında bu gerçekleşir.
- ▶ Lokal değişkenlerin tanımlı olduğu **kapsam** (**scope**) sonlandığında bu değişkenler stack kısmından silinir.

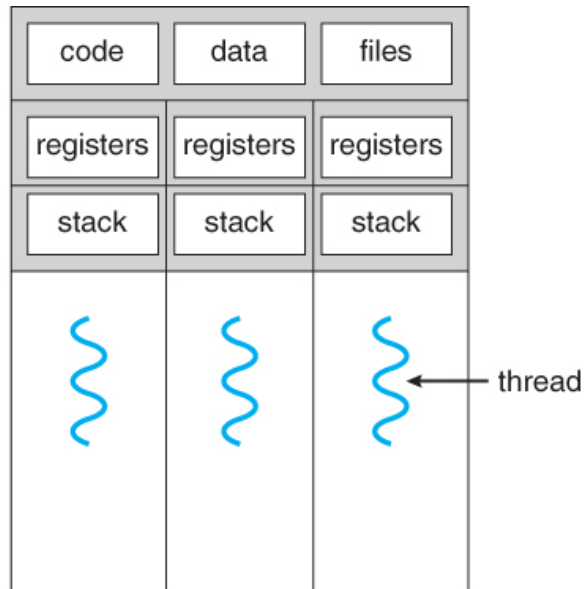
- ▶ Bu kısım aynı zamanda **fonksiyon dönüş değeri** (**function return value**)'nin de tutulduğu kısımdır.
- ▶ Stack yönetiminin detayları kullanılan programlama diline göre farklılık gösterebilir.
- ▶ Burada dikkat edilmesi gereken bir diğer nokta da **stack** ve **heap** genişleme yönlerinin birbirlerinin zıddı olmasıdır (şekilde ok yönlerine dikkat ediniz).

- ▶ İşlem (process) için kullanılabilir boş alan, stack ve heap genişlemesi neticesinde daralacaktır. Bu daralma ana hafızada program için tahsis edilen alanda hiç boş yer kalmayıncaya kadar devam eder.
- ▶ Eğer ana hafızada hiç boş alan kalmazsa yani stack ve heap kısımları birbirlerine sınır olacak şekilde genişlerlerse, ya **yığın taşması (stack overflow)** oluşur yada **calloc, malloc** veya **new** gibi yapılara yapılan çağrılar yetersiz hafıza nedeniyle başarısız olur.

- ▶ Bir **işlem** (**process**)'de bir veya birden fazla **iş parçacığı** (**thread**) bulunabilir. İş parçacığı bir işlemin içerisindeki bağımsız komut akışını ifade eden bir kavramdır.
- ▶ Her bir thread birkaç bileşenden oluşur:
  - ▶ thread numarası (thread ID)
  - ▶ yığın (stack)
  - ▶ kaydediciler (registers)
  - ▶ program sayacı (program counter)



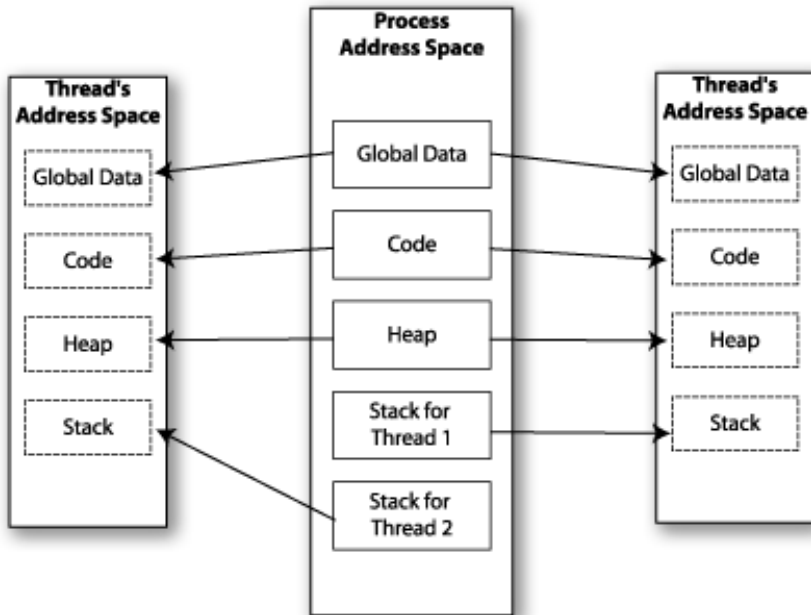
single-threaded process



multithreaded process

Kaynak



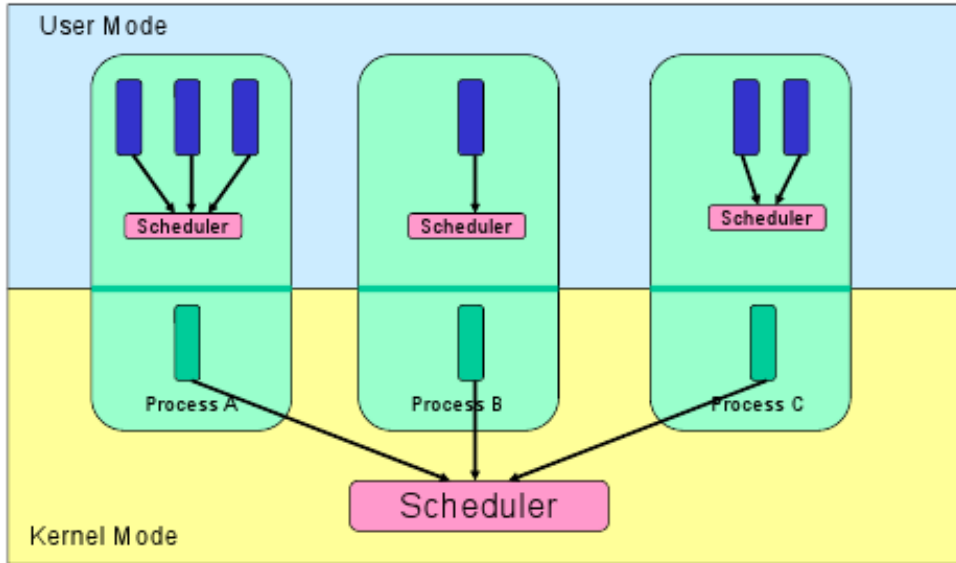


- ▶ İşlem (process) ve iş parçacığı (thread) arasındaki temel fark şudur:
- ▶ Her bir işlemin RAM'de kendine özel kapladığı bir adres uzayı (address space) vardır ve her bir işlem sadece kendi adres uzayını (hafıza alanını) görebilir ve değiştirebilir.
- ▶ Fakat iş parçacıkları ortak adres uzayını kullanırlar (yığın ve kaydediciler kısımları hariç).

- 1 Donanım Seviyesinde İş Parçacıkları  
(Hardware Level Threads)
- 2 Yazılım Seviyesinde İş Parçacıkları  
(Software Level Threads)

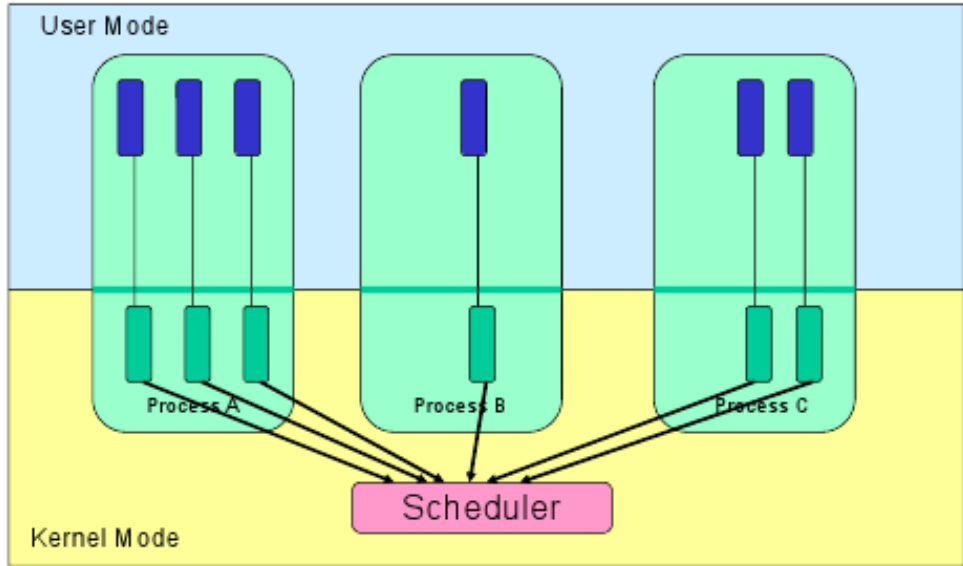
- 1 Kullanıcı Seviyesinde İş Parçacıkları  
(User Level Threads)
- 2 Çekirdek Seviyesinde İş Parçacıkları  
(Kernel Level Threads)

# İş Parçacığı Uygulamaları : Kullanıcı Seviyesi



Kaynak

# İş Parçacığı Uygulamaları : Çekirdek Seviyesi



Kaynak

