

# İşletim Sistemlerine Giriş

İşletim Sistemleri ve Donanım

# İşletim Sistemi ?

Yazılım olmadan bir bilgisayar METAL yığınıdır.

Yazılım bilgiyi saklayabilir, işleyebilir ve daha önceden saklanmış bilgiyi kullanabilir.

## YAZILIM İKİYE AYRILIR:

**1. Sistem Programları :** Bilgisayarın kendisine ait olan işleri yerine getirir.

**2. Uygulama Programları:** Kullanıcının problemlerini çözer.

# İşletim Sistemi ?

En temel ***SİSTEM PROGRAMI İŞLETİM SİSTEMİ*** dir.

Bilgisayarın tüm kaynaklarını kontrol eder ve uygulama programlarının yazılabilmesi için gerekli olan temeli oluşturur.

Uygulama programları işletim sistemi üzerine yazılır.

# Bilgisayarın içerisinde neler vardır?

Bir ya da daha fazla *işlemci, bellek, monitör, diskler, ağ aygıtları, yazıcılar, cd rom, dvd rom ve daha birçok Giriş/Çıkış (I/O) aygıtı* vardır.

## **PROBLEM :**

Bilgisayarda çalışacak programın tüm bu bileşenlerin nasıl çalıştığını ve bu bileşenler ile nasıl haberleşeceğini bilmesi gereklidir.

# ÇÖZÜM

Bu problemi ortadan kaldırmak için donanım ile programcılar arasına karmaşıklığı azaltan ve programcıya yardım eden bir katman konulması tasarlanmıştır.

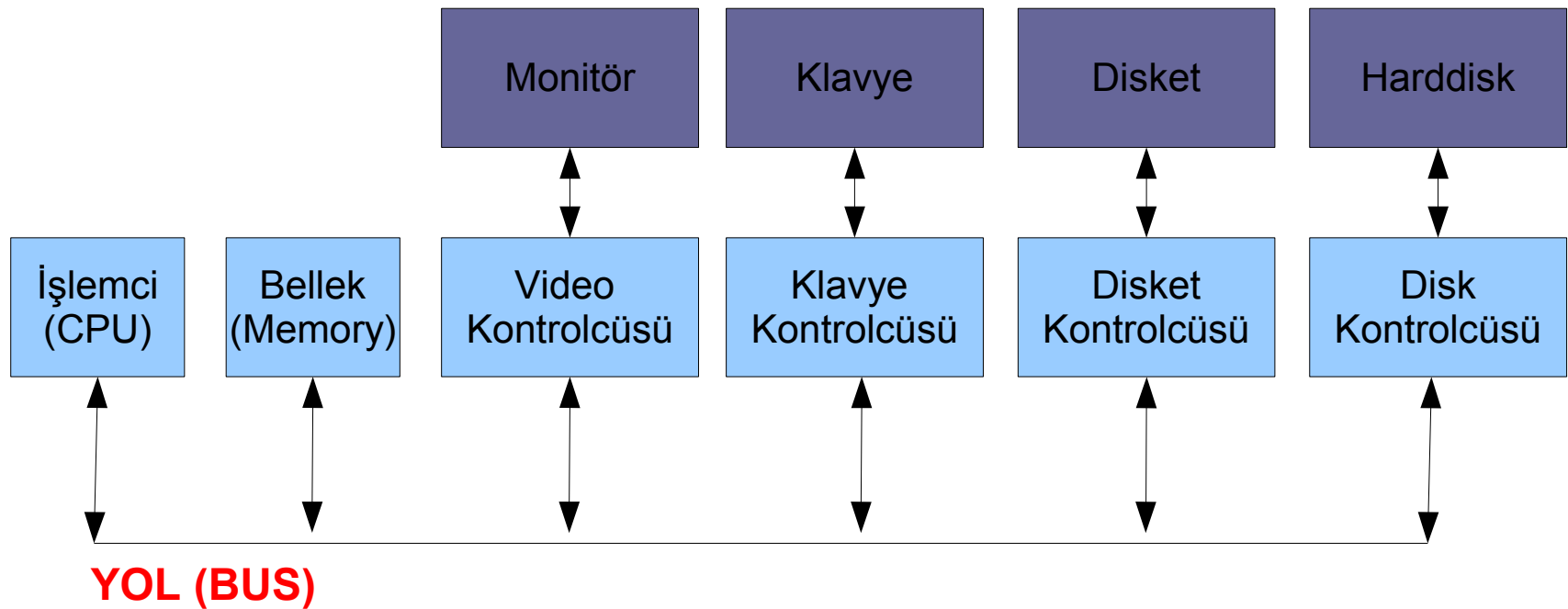
Bu katmanda bulunan yazılama  
**İŞLETİM SİSTEMİ** denilir.

# BİLGİSAYAR SİSTEMİ



# BİLGİSAYARIN DONANIMI

Bir işletim sistemi üzerinde çalıştığı donanıma tamamen bağımlıdır.



# İŞLEMCİLER (CPU)

Bilgisayarın beynidir.

Bellekten komutları (instruction) alır (fetch),  
komutun anlamını çözer (decode) ve  
çalıştırır (execute).



# İŞLEMCİLER (CPU)

Her işlemcinin çalıştırabileceği komutlar sınırlıdır ve o işlemciye özeldir.

Bu nedenle Pentium işlemci için yazılan kod, SPARC işlemci üzerinde çalışmaz.

# İŞLEMCİLER (CPU)

İşlemci üzerinde YAZMAÇ (REGISTER) adı verilen saklama birimleri bulunur. Bu birimler son derece hızlıdır.

İşlemcinin çalıştırdığı komutlar genelde yazmaçlar üzerinde işlem yapar.

# ÖZEL YAZMAÇLAR(REGISTERS)

**Program Sayacı (Program Counter-PC):**  
getirilecek olan bir sonraki komutun bellek adresini tutar.

**Yığın İşaretçisi (Stack Pointer-SP):**  
Bellekte bulunan yığın alanının en üst adres bilgisini tutar.

**Program Durum Kelimesi(Program State Word-PSW):**

Çeşitli durum bilgilerinin tutar. Komutların işlenmesi sonucunda oluşan çeşitli olayların bilgisini tutar. Ayrıca kip(mode) bilgisini tutarlar.

*Kullanıcı Kipi(User Mode) , Çekirdek Kipi (Kernel Mode)*

# CPU larda İki Kip Bulunur.

- \*Kullanıcı Kipi (User Mode)

- \*Çekirdek Kipi (Kernel Mode)

PSW deki 1 bit bunu kontrol eder.

CPU çekirdek kipinde tüm komutlarını kullanabilir, kullanıcı kipinde belirli bir komut alt kümesi kullanabilir.

Bu kipler donanıma direkt erişimi denetim altında tutmak için kullanılır.

# CPU larda İki Kip Bulunur.

İşletim sistemi Çekirdek Kipinde çalışır.

Kullanıcı programları kullanıcı kipinde çalışır. Eğer kullanıcı Giriş/Çıkış işlemi, bellek ile ilgili işlemler, ... yapacak ise bu işlemleri işletim sistemi üzerinden yapar.

Bu şekilde sistem denetim altında korunmuş olur.

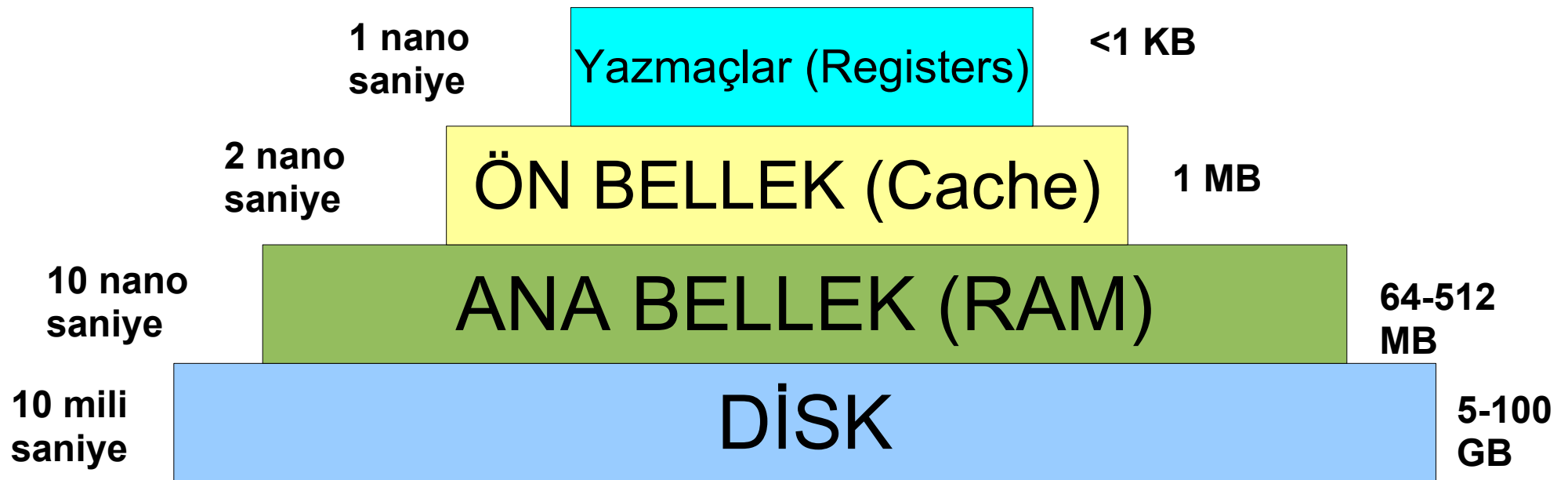
# BELLEK (MEMORY)

Bellek CPU hızına yakın hızlara sahip depolama donanımlarıdır.

CPU bilgileri bellekten alır ve belleğe yükler.

Bellek hızları CPU hızına oranla yavaş olduğu için katmanlar şeklinde tasarlanırlar.

# BELLEK (MEMORY)



# BELLEK (MEMORY)

Bilgi önbellekte aranır orada varsa yazmaçlara alınarak kullanılır, yoksa bellekte aranır.

Bilgi bellekte de bulunmuyorsa diskten önce belleğe daha sonra ön belleğe ve oradan da işlemciye getirilerek kullanılır.



# BELLEK (MEMORY)

Bir program derlenip, bağlandıktan sonra fiziksel bellekte nereye yerleştirileceğini, programın hangi adresten başlayacağını bilemez.

Bu yüzden derleyici ve bağlayıcı programın 0 adresinden başlatırlar.

# BELLEK (MEMORY)

Örneğin, bir komut 10000 adresindeki veriyi yüklemek istesin. Program ve tüm verisi bellekte 50000 adresinden itibaren yerleştirilmiş olsun. İlk komut çalıştırıldığında komut 10000 numaralı adrese erişmeye çalışacaktır.

Bu adres doğru değildir. Erişimlesi gereken gerçek adres 60000 dir.

# BELLEK (MEMORY)

Bu problem için iki yazmaç kullanılır. **Taban yazmacı (base register)**, ve **sınır yazmacı (limit register)**. Program belleğe atıldığında ve çalışmaya başladığında, taban yazmacı programın başlangıç adresini, sınır yazmacı ise programın tüm verisi ile birlikte bittiği adresi tutar.

Program tarafından kullanılan adrese **sanal adres (virtual address)**, belleğin kullandığı adrese ise **fiziksel adres (physical address)** denilir. Bu işlemleri ve çevrimi kontrol eden bellek yönetim birimi **MMU (memory management unit)** dir.

# Giriş/Çıkış (I/O)

Bir I/O aygıtı, aygıtın kendisi ve bir kontrolcü(controller) olmak üzere iki parçadan oluşur.

Kontrolcü aygıtın kendi üzerinde yer alan bir çiptir. Aygıtın çalışmasını kontrol eder. İşletim sistemi yapmak istediği komutları kontrolcüye gönderir, sonuçları da kontrolcünden alır.

Kontrolcü ile konuşan, komut gönderen ve cevaplar alan yazılımı *aygıt sürücüsü (device driver)* denilir.

# Giriş/Çıkış (I/O)

Bu aygıtın kullanılması için işletim sistemine eklenmesi gereklidir.

1.yöntem : işletim sisteminin çekirdeğine aygıt sürücüsü eklenir ve yeni çekirdek oluşturulur.

2.yöntem: İşletim sisteminin sistem dosya sistemine eklenir ve sürücünün nerede bulunacağı söylenir. Sistem yeniden başlatılır. İşletim sistemi ihtiyaç halinde yükler.

3.yöntem: İşletim sistemi sürücüyü çalışırken kabul eder, yükler ve kullanır. (USB)

# Giriş/Çıkış (I/O)

Kullanıcı I/O için bir istek yapar. Bu isteği işletim sisteminin sunmuş olduğu özel prosedürleri kullanarak yapar.

İşletim sistemi gelen isteği uygun aygıt sürücüsünün ilgili prosedürüne gönderir.

I/O bitince sürücü veriyi ya da sonuç ne ise belirtilen yere koyar.

Veri okunurken CPU bekler. Bu işleme yoğun bekleme(busy waiting) denilir.

# Giriş/Çıkış (I/O)

ikinci yöntem, sürücü başlatılır ve sürücüye işini bitirdiğinde bir sinyal-kesme(interrupt) göndermesi söylenir.

Sürücü komutu alır kendisi çalışmaya başlar ve işletim sistemine geri döner. İşletim sistemi istekte bulunan programı beklemeye alır ve başka birini çalıştırır.

Kontrolcü işini bitirdiğinde bir bilgilendirme sinyali -kesme (interrupt) gönderir. Bloklanan programa geri dönülür.

# Giriş/Çıkış (I/O)

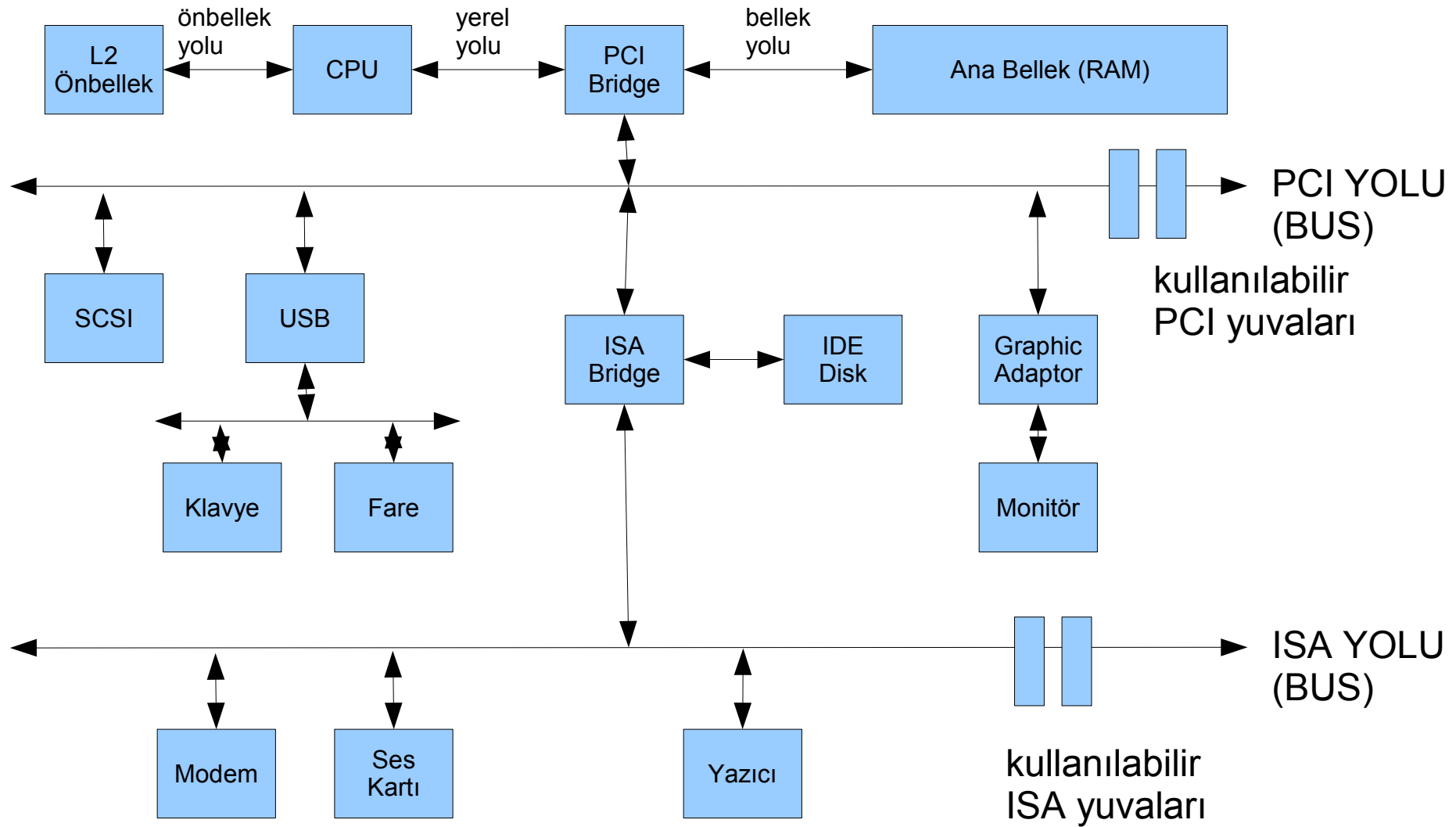
üçüncü yöntem, özel DMA (Direct Memory Access) çipleri kullanmaktır.

Bu çip CPU kullanılmadan kontrolcü ile bellek arasındaki bir akışını yönetir.

CPU, DMA ya hangi aygıttan ne kadar veri alacağını ve nereye koyacağını söyler. DMA işlemleri kendisi yaparken CPU başka işler yapar. DMA işini yaptığında bir kesme oluşturur ve CPU yu bilgilendirir.



# YOLLAR (BUSES)



# YOLLAR (BUSES)

Bilgisayardaki aygıt ve donanımlar arasındaki hız farkı nedeniyle farklı yollar vardır.

ISA – Industry Standard Architecture IBM

PCI- Peripheral Component Interconnect

# İşletim Sistemlerine Giriş

## İşletim Sistemleri ve Donanım