

BİLGİSAYAR ORGANİZASYONU

- **Donanım**

- Bilgisayarın fiziksel bölümü
- Monitor, klavye, fare
- Entegreler, kartlar
- Kablolar

- **Yazılım:**

- Bilgisayarın mantıksal bölümü
- Programlar: Bilgisayarın gerçekleştireceği komutlar



BİLGİSAYAR DONANIMI

- Merkezi İşlem Birimi(Central Processing Unit)
- Ana Bellek (Main Memory)
- İkincil Bellek (Secondary Storage)
- Giriş Birimleri(Input Devices)
- Çıkış Birimleri(Output Devices)

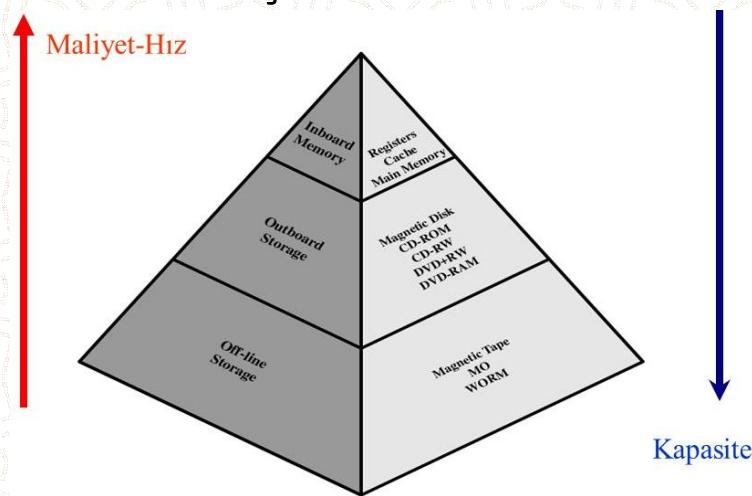


MERKEZİ İŞLEM BİRİMİ

- Bilgisayarın beynidir ☺
- İki bölümden oluşur:
 - Kontrol Birimi(Control Unit - CU)
 - İşlem akışını kontrol eder, komutları yerine getirir
 - Aritmetik-Lojik Birimi(Arithmetic Logic Unit - ALU)
 - Toplama, çıkarma, karar verme vs. işlerini yapar



BELLEK HİYERARŞİSİ



ANA BELLEK

- Rasgele Erişimli Bellek (**Random Access Memory**)
- Herbiri 1 byte'lık hücrelerden oluşur.
- Her bellek hücresinin adresi vardır.
- Bu adreslerde veri veya komut saklanır.
- Saklanan bilgiler 0 ve 1'lerden oluşur.



ANA BELLEK

- İkili sistem kullanıldığı için bellekte saklama ve adresleme de 2'nin kuvvetlerine göredir.
 - 1 kilobyte = $1024 \text{ byte}(2^{10})$
 - 1 megabyte = $1,048,576 \text{ byte}(2^{20})$
 - 1 gigabyte = $1,073,741,824 \text{ byte}(2^{30})$
 - 1 terabyte = $1,099,511,627,776 \text{ byte}(2^{40})$
- Bilgisayar kapatıldığında bilgiler silinir (**volatile**).



İKİNCİL BELLEK

- Bilgiler kalıcıdır.
- Rasgele veya sıralı erişim olabilir.
- Disk sürücülerı
 - Harddisk, floppy disk ..
- Optik sürücüler
 - CD, DVD ..



BELLEK BİRİMLERİ

- RAM(Random Access Memory)
 - Rasgele erişim mümkün
 - Okuma/yazma yapılabilir
- ROM(Read Only Memory)
 - Rasgele erişim mümkün
 - Sadece okuma yapılabilir
 - Genel olarak ana kartta bilgisayar ilk açıldığında gerekli komutları saklar.



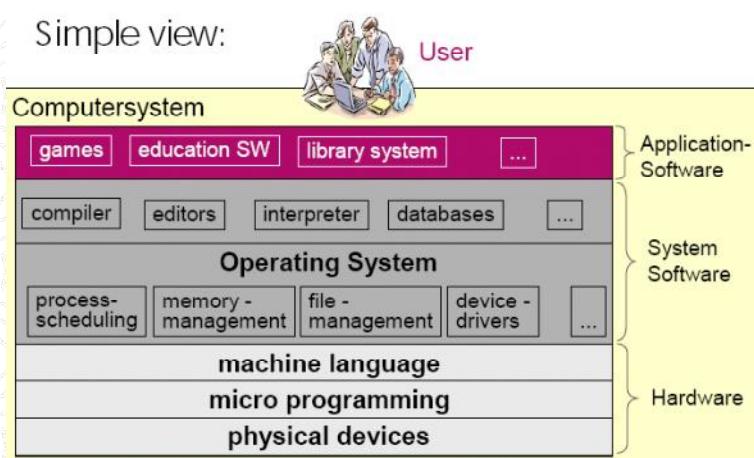
Giriş/Çıkış Birimleri

- Dışardan bilgisayara veri aktarımı
 - Klavye, fare, tarayıcı..
- Bilgisayardan dışarıya bilgi aktarımı
 - Monitör, yazıcı ..



Bilgisayar Organizasyonu

Simple view:



YAZILIM TÜRLERİ

- Sistem Yazılımları
 - İşletim Sistemleri, Derleyiciler ..
- Uygulama Yazılımları
 - Yazım araçları, istatistik paketleri, çizim araçları, eğitim paketleri, oyunlar vs.



PROGRAMLAMA DİLLERİ

- 1. Nesil: Makine kodu(Machine code)
- 2. Nesil: Assembly diller
- 3. Nesil: Yöntemsel(procedural) diller
- 4. Nesil: Uygulamaya özel diller
- 5. Nesil : Kısıtlı diller



1. NESİL : MAKİNA DİLİ

- Komutlar 1'ler ve 0'larla ifade edilir.
- **Örnek** : İki sayıyı toplama için makine kodu:

000000	00001	00010	00110	00000	100000
--------	-------	-------	-------	-------	--------



2. Nesil : Assembly Diller

- Makine kodunu anlaması zor, hata yapması kolay
- Çözüm: 1950'lerde assembly diller
- Alçak seviyeli(low level) diller
- İkili sayı yerine anlaşılır kısaltmalar
- **Örnek** : ADD AX, BX



2. Nesil : Assembly Diller

- Bilgisayar sadece makine kodunu anlar
- Assembly program, makine koduna (object code) çevrilmeli
- Çevirme işlemini **asembler** yapar



3. NESİL : YÖNTEMSEL DİLLER

- Assembly'de de kodlaması zor
- Farklı mimariler farklı assembly komutları kullanıyor
- **Çözüm**
 - 1950'lerde yöntemsel diller
 - İngilizce komutlar
- İleri seviyeli(high level) programlama dilleri:
 - **Fortran, Algol, Pascal, C, Java..**



İleri seviye Programlama Dillerinin Tarihçesi

- 1950'ler Grace Hopper : **Cobol**
- 1954, Backus & IBM grubu : **Fortran** –bilimsel programlama-
- 1968, N. Wirth: **Pascal**
- 1970, B. Kernighan, D. Ritchie : **C** (UNIX OS yazılımı için)



NESNEYE DAYALI DİLLER

- Yazılımlar karmaşıklaştıkça daha efektif kodlama ihtiyacı
- Nesneye dayalı programlama ile modüler ve yeniden kullanılabilir kodlar
- İş yapan, birbirleriyle haberleşen nesneler(objeler) üzerine kurulu



Nesneye Dayalı Dillerin Tarihçesi

- 1967, Dahl&Nygaard(NCC) : **Simula**
- 1980, Goldberg(Xerox) : **Smalltalk-80**
- 1988, B. Stroustrup : **C++**
- 1995, J. Gosling(SUN) : **JAVA**
- 1995, OMG : **UML**(Unified Modelling Language)



Interpreter - Compiler

- Compiler (Derleyici): High-level bir dille program yazdığımız kaynak kodun makine koda dönüştürülmesini sağlar
- Interpreter (Yorumlayıcı): Program bir seferde değil satır değil satır derlenir. Bir satır çalıştırıldıktan sonra, bir sonraki satır çalıştırılır
- Pekçok programlama dili hem derleyici hem yorumlayıcı ile çalışabilir
- Derleyici ile çalışan diller:
[C](#), [JAVA](#), [C++](#)..
- Yorumlayıcı ile çalışan diller:
[Basic](#), [Lisp](#), [MATLAB](#), [SmallTalk](#) ..



4. NESİL : UYGULAMA ÖZEL DİLLER

- Rapor üreticiler
 - Quest, Oracle Reports, RPG II ..
 - Veritabanı Sorgulama
 - SQL, Informix-4GL, FOCUS ..
 - Veri İşleme, Analiz ve Raporlama
 - MATLAB, Mathematica, PL/SQL..
 - ...



5. Nesil : Kısıtlı(Constraint) Diller

- Yapay Zeka uygulamaları için geliştirilen diller
 - Prolog, Mercury, LISP, ..

- Görüsel(visuel) diller: Kutu, daire vs. görsel ifadelerle program yazımı
 - Simulink, LabVIEW, Analytica, Flow..

