

İşletim Sistemleri

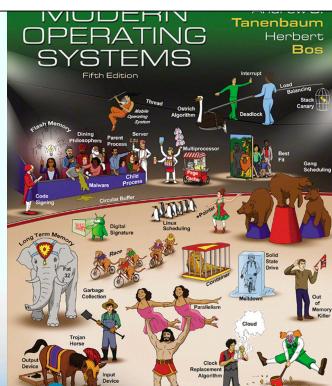
Giriş

Emir ÖZTÜRK

İsletim Sistemleri

Giriş

- Bilgisayarın "parçaları"
 - CPU
 - RAM
 - Disk
 - I/O
 - Parçaların yönetimi

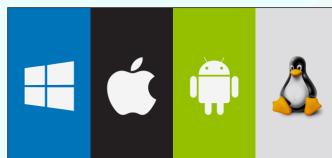


Bir bilgisayarın çalıştırılabilmesi için belirli bir donanıma sahip olması gerekmektedir. Temelde işlemlerin yapılması için cpu, bu işlemlerin verisinin tutulması için ram, verilerin kalıcı bir şekilde bilgisayar kapandığında kaybedilmeden saklanması için disk ve bilgisayar ile haberleşme için gerekli giriş çıkış donanımı bulunması gerekmektedir. Bu donanımın yönetimini sağlamak işletim sisteminin ilk görevidir.

İşletim Sistemleri

Çeşitlilik

- Yalnızca pc için değil
 - Gömülü cihazlar
 - Telefon - tablet
 - Server'lar



İşletim sistemi kavramı yalnızca kişisel bilgisayarlar için geçerli değildir. Mevcut kullanılan bir çok cihaz işletim sistemine sahiptir. Gömülü cihazlar, mobil telefonlar, internetteki verilerin bulunduğu sunucular gibi bir çok donanımda bir işletim sistemi bulunmaktadır.

İşletim Sistemleri

Etkileşim katmanları

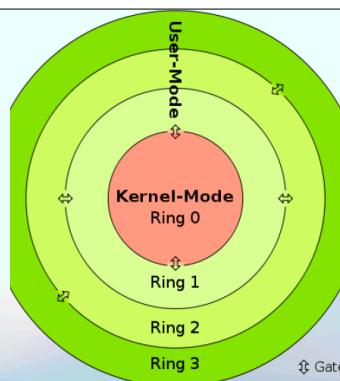
- Shell
- Text tabanlı
- GUI
- Gooey



İşletim Sistemleri

İşletim sistemi modları

- Kernel Mode
- User Mode
- Keskin bir sınır yok



İşletim Sistemleri

Bakım yapılabılırlik

- Windows
- 50 milyon satır
- Linux
- 20 milyon satır



Bilgisayar ile etkileşim için bir çok yol bulunmaktadır. Tarihsel olarak bakıldığından bu iletişim kartları vasıtasıyla yapılmırken günümüzde işletim sistemlerinin üzerinde çalışan bir grafik kullanıcı arayüzü bulunmakta ve bu arayüz ile bilgisayar etkileşimi sağlanmaktadır. Grafik kullanıcı arayızlarından önce bulunan “shell” adı verilen iletişim katmanı ise istenirse işletim sistemleri içerisinde çalıştırılmıştır. Shell ile komut yazma yolu ile etkileşime geçilirken bir pencere yöneticisinin kullanılması durumunda kullanıcının girdi bekleyen bir arayüz bulunmaktadır.

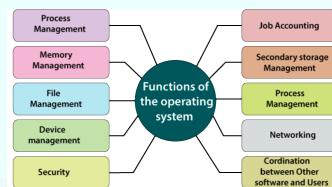
İşletim sistemlerinin çalışma modları bulunmaktadır. Bu modlar çekirdek modu ve kullanıcı modu olarak adlandırılabilirler. Çekirdek modu işletim sisteminin tüm kaynaklara erişiminin sağlandığı durumlarda, sürücülere erişimde ve ayrıcalıklı işlemlerin çalıştırılmasında görev alırken kullanıcı modu ise uygulama programlarının çalıştığı ve işletim sistemlerinden sadece belirli kaynakların tahsis edilebildiği bölümdür. İşletim sistemlerinde kullanıcı ve çekirdek modları arasındaki sınır her zaman çok keskin olmayıpabilir.

İşletim sistemleri farklı bir çok donanımda farklı işleri yapabilecek durumda olmalıdır. Bu da işletim sistemlerinin karmaşıklığını oldukça artırmaktadır. Bu da işletim sistemlerinin değişiklik ya da bakım durumlarının çok zor olduğu anlamına gelmektedir.

İşletim Sistemleri

Amaçlar

- Donanımın yönetimi
- Uygulama geliştiricilere bir kaynak seti oluşturmak
- Asıl müşteri uygulama geliştiricileri
- Son kullanıcı tarayıcı tabanlı bile çalışabiliyor, OS farketmiyor



İşletim Sistemleri

Birimlerin mimarisi

- Bir diskin arayüzü için kılavuz 450 sayfa
- Maliyet
- Geriye yönelik uyumluluk
- Uygulama geliştiricileri için her seferinde kodlanması imkansız
- Bunun yerine "sürücü" kavramı
- Sürcü bile primitif
- Dosya sistemleri



İşletim sisteminin görevi yalnızca donanımı yönetmek değildir. Bu donanımı yöneterek etkin bir şekilde kullanıcının işini gerçekleştirmesi gerekmektedir. Ayrıca işletim sistemleri bu kaynakların istenebildiği ve kullanılabileceği bir arayüzü kullanıcılar sunmalıdır. Burada bahsedilen arayüz daha önceki slaytta bahsedilen GUI değil, bir uygulama programlama arayüzü (API) olarak kabul edilebilir. İşletim sistemlerinin asıl hedefi bu açıdan uygulama geliştiricileridir. Bunun nedeni son kullanıcının donanıma erişim için herhangi bir istekte bulunmaması fakat kullandığı uygulamanın bu yönetimi sağlamasıdır. Örneğin fotoğraf çekip bunu bir platformda paylaşan bir uygulama kameraya, işlemciye, ağ erişimine ve resmi kaydetmek için disk erişimine ihtiyaç duyar. Bahsedilen tüm bu kaynakların uygulamaya tahsis işlem sisteminin görevidir. Son kullanıcı ise yalnızca uygulama ile haberleşmektedir.

Temelde her sürücü elektronik olarak yönetilmeli ve bu oldukça düşük seviyeli bir kodlama ile yapılmalıdır. Uygulama geliştiricilerinin tüm donanımın detaylarına hakim olma gibi bir ihtimali bulunmamaktadır. Ayrıca uygulama geliştiricilerinin her sürücü için ayrı bir arayüz geliştirmesi olanağı da bulunmamaktadır. Uygulama geliştiricilerinin dışında işletim sistemlerinin bile tüm donanım sürücülerini içermeye şansı bulunmaz. Bu sebeple işletim sistemlerinin birimler ile haberleşmesi için o birimin bir sürücüsünün olması gerekmektedir. İşletim sistemlerine tanıtılan bu sürücüler birim ile haberleşmeyi sağlayan bir arayüz sunar ve işletim sistemlerinin isteklerini karşılamak için birim ile haberleşirler.

Örneğin bir diske yazma ve okumanın düşük seviye detayları ile sürücü ilgilenirken işletim sistemi sektörler ve bloklar düzeyinde çalışır. Uygulama geliştiricisi ise işletim sisteminin sunduğu dosya sistemleri ile haberleşme işlemi gerçekleştirirler. Bu modüler yapı sayesinde her katman standartlaştırılabilmektedir.

İşletim Sistemleri

Kaynak yönetimi

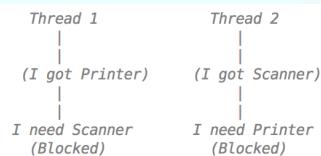
- Önemli bir diğer görev
- Cihaz soyutlamaları için kaynakları uygulamalara paylaştırmak gereklidir
- İşlemci, ram, diskler vb girdi / çıktı bekler
- Ayrıca birden fazla işlemin tek kaynağı kullanmak isteme durumunun çözümü



İşletim Sistemleri

Kaynak paylaşımı örneği

- Örneğin iki işlemin aynı anda yazıcıyı kullanmak istediği düşününelim
- Her biri yazıcıya gönderilmesi gereken veriyi paralel gönderirse ne olur?
- Çözüm?
- Buffer



İşletim sistemlerinin bir görevi de birden fazla işlemin çalıştırılması durumunda bu işlemler arasında kaynak paylaşımıdır. Özellikle işlemcinin işlemler tarafından ortak kullanıldığı senaryolarda işlemciyi kullanmayan işlemin durdurulup yerine işlemciyi kullanacak işlemin getirilmesi gerekmektedir. Bu sırada işlemin sahip olduğu verinin bellekten diske aktarımı da sağlanmalıdır. Bunun dışında veri güvenliği, hata tespiti dosya yönetimi gibi işlemler de işletim sistemi görevlerindendir.

İşletim Sistemleri

Çoklu kullanıcılar

- Birden fazla kullanıcının aynı anda sistemi kullanabilme gerekliliği.
- Kaynakları birbirine karışmaması gerekiyor.
- Kaynakları etkin yönetimi gereklidir.
- Ayrıca zaman ve yer açısından paylaşım planlamalarının yapılması gereklidir.

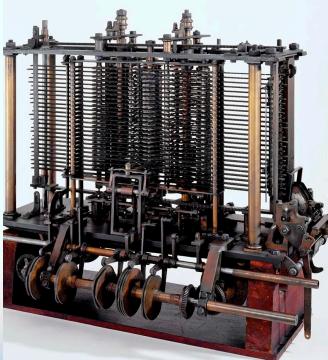


Çok kullanıcıya sahip sistemlerde kaynakların kullanıcılar arası paylaşımı problemi bulunmaktadır. Her kullanıcı aynı anda sistemi kullanmadığı gibi her kullanıcının ihtiyaç duyduğu kaynak miktarı da anlık olarak değişmektedir. İşlemci zamanı ve bellek alanı açısından hesaplamaların ve paylaşımın kullanıcıların bekleyicilerini karşılayacak şekilde ayarlanması gerekmektedir.

İşletim Sistemleri

İşletim sistemlerinin tarihi

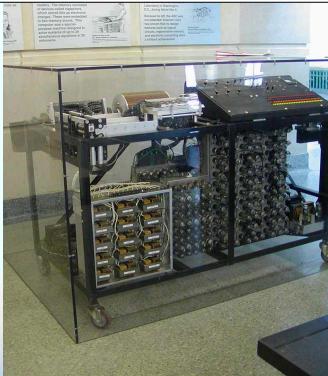
- Charles Babbage
- Analytical engine
- Keskin teknoloji eksikliği
- Saf mekanik bir sistem
- Yazılım ihtiyacı için
- Ada Lovelace



İşletim Sistemleri

İlk bilgisayar girişimleri

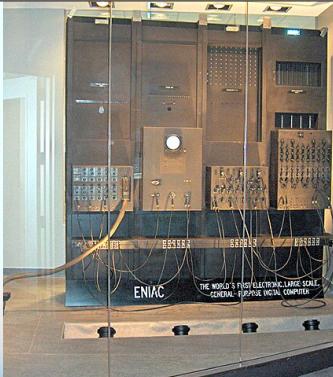
- Atanasoff Berry
- 300 elektron tüpü
- Zuse Z3
- Colossus
- Alan Turing



İşletim Sistemleri

İlk bilgisayar girişimleri

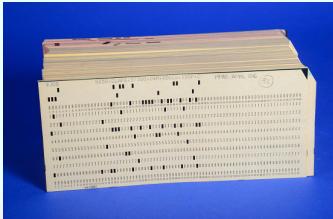
- Mark 1
- ENIAC
- Hesaplamalar çoğunlukla makine dili
- Hatta donanımsız olarak devrelerin değiştirilmesi
- Assembly henüz yok
- İşletim sistemi kavramı yok
- Problemlerin tamamı matematisel ve nümerik hesaplamalar



İşletim Sistemleri

İş kavramı

- 1950'lerde transistörün icadı
- Mainframe'ler
- İş - JOB kavramı
- Her "iş" için yazılımcı önce Assembler veya Fortran ile kodlama yapar
- Kart delme işlemi
- Kartlar operatörlerle teslim edilir
- İş sonunda çıktı kartları programcıya teslim edilir



İşletim sistemlerinden önce yapılacak işlemler job (iş) olarak adlandırılmıştır. Böyle bir durumda her iş bir kart grubunun bilgisayara verilmesi ile yönetilmektedir.

İşletim Sistemleri

Birden fazla işin çalıştırılması

- Bu işlerin otomatize edilmesi için işleri sıraya alan basit bir bilgisayarın kullanılması.
- İşlerin birinin bitmesi ile diğerini sıraya alan bir sistem
- İlk işletim sistemi

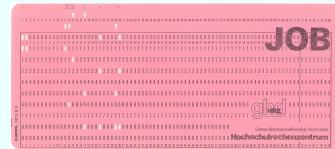


İşler tamamlandığında sıradaki işlerin alınma işleminin otomatizasyonu ilk yönetim ve bu sebeple ilk işletim sistemi prototipi olarak kabul edilebilir.

İşletim Sistemleri

Bir "iş" sırası

- \$JOB kartı
 - Maksimum çalışma süresi
 - Hesap numarası
 - Çalıştırın kişinin adı
- \$FORTRAN kartı ile compiler'in yüklenmesi
- Derlenecek program
- \$LOAD kartı
- \$RUN kartı
- \$END kartı



İşletim Sistemleri

Farklı mimariler

- 1960'larda karmaşık bilimsel problemlerinin çözümü için word-oriented bilgisayarlar
- Basit işlemler için char-oriented bilgisayarlar
- İki ürünün paralel geliştirilmesi ve takibi problem yaratmaktadır
- Mimari değiştirmek isteyenler, daha hızlı ortama geçmek isteyenler programlarını değiştirmek istemiyor.



Farklı mimarilere sahip bilgisayarlar üzerinde aynı kodun çalıştırılma isteği ile bir standartlaşma ihtiyacı duyulmuştur.

İşletim Sistemleri

360

- IBM - System/360
- Modüler
- Fiyat / performans
- Diğer büyük firmalar da bu yolu izlemeye karar veriyor.
- OS/360



İşletim Sistemleri

One ring to rule them all

- OS/360'in amacı farklı sistemlerin tamamında tam performans sağlayan bir işletim sistemi oluşturmak
- Pratikte mümkün değil
- Milyonlarca satır
- Bir çok bug
- Buna rağmen elde edilen başarı



İşletim Sistemleri

Yeni kavramlar

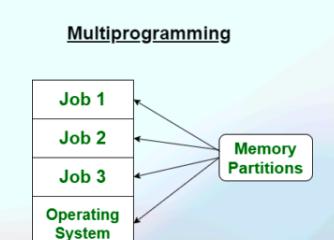
- OS/360 ile ortaya çıkan kavramlar
- Multiprogramming
 - Sistemin IO beklerken CPU'nun boşta kalmamasının amaçlanması
- Spooling
 - İşler bitikçe otomatik yeni işin okunması
 - Tepk okuma bilgisayarlarının ortadan kalkması



İşletim Sistemleri

Multiprogramming problemi

- Multiprogramming mantığında CPU paylaşımına OS karar veriyor.
- Programalar ise belirli bir aralıkta bilgisayarın tamamen kendilerine ait olmasını istiyorlar.
- Hata olduğunda sonucun alınması daha uzun süreceğinden daha fazla zaman harcıyor.

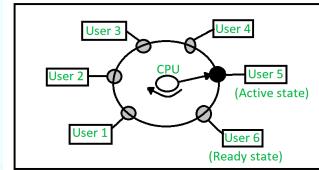


Bir işin IO cihazı veya başka bir kesme sebebi ile beklemesi durumunda CPU'nun boşta kalmaması amacı ile başka bir işe geçilebilir.

İşletim Sistemleri

Time sharing

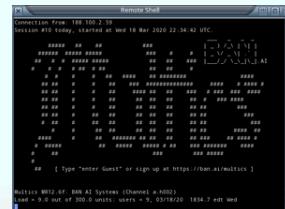
- Bunların çözümü için timesharing
- Herkes paralel olarak sisteme bağlı. İş isteyenlerin işi devreye alınıyor.
- İlk timesharing sistemi CTSS (MIT)
- Bu fikrin geliştirilmesi - MULTICS
- Çok büyük bir bilgisayar ve birden fazla kullanıcı
- Şimdiki bulut mimarisine benzer



İşletim Sistemleri

MULTICS

- PL/I compiler ile yazılıyor
- Compiler geç çalışıyor
- Çıkığı zaman da problemli
- Kütlesi sadık
- 90'ların sonuna kadar kullanım



İşletim Sistemleri

UNIX

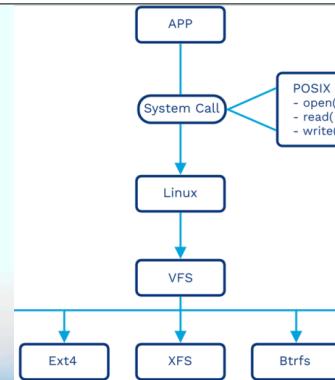
- PDP bilgisayarlar
- Ken Thompson
- PDP-7
- UNIX
- Akademide yaygın kullanım



İşletim Sistemleri

POSIX

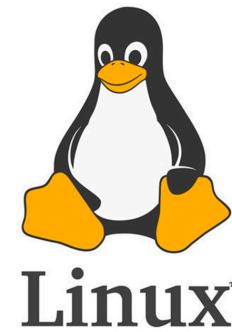
- UNIX sistemlerinin standartlaştırılması
- Sistemler için ortak bir uygulama geliştirme standardı
- POSIX
 - System-call interface



İşletim Sistemleri

Linux

- Tanenbaum
 - UNIX klonu MINIX
- Linus Torvalds
 - Linux



İşletim Sistemleri

4. nesil işletim sistemleri

- 1980 sonrası
- Kişisel bilgisayarlar yaygınlaşıyor
- Intel 8080
- Kildall'dan işletim sistemi isteniyor
- CP/M
 - 5 sene kullanılıyor



İşletim Sistemleri

DOS ve MS-DOS

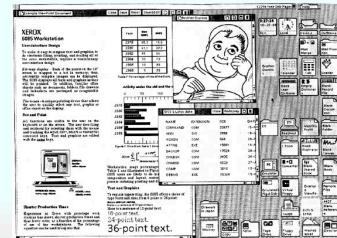
- IBM PC geliştiriliyor.
- Bill Gates'ten BASIC interpreter'i alınıyor
- İşletim sistemi istendiğinde Kildall'a yönlendiriliyor
- Kildall IBM ile görüşmüyordu.
- Bill Gates SCP'nin DOS'unu alıyor.
- DOS-BASIC paketini IBM'e satıyor
- Tim Paterson
- MSDOS



İşletim Sistemleri

GUI

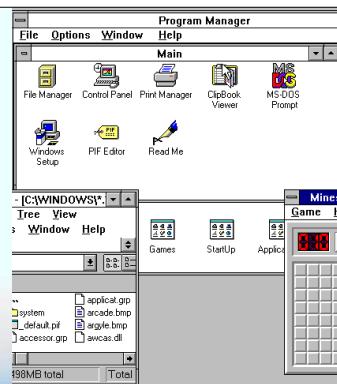
- Engelbart
- GUI
- Xerox PARC
- Steve Jobs
- Apple 2
- Microsoft
- Windows



İşletim Sistemleri

Windows

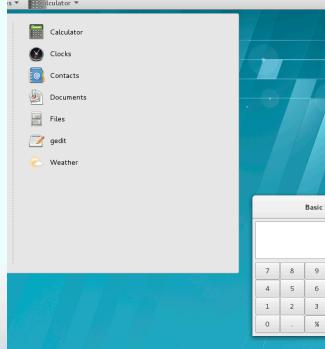
- Windows dos üzerinde çalışıyor
- Win95'ten itibaren birlleştiriliyor
- NT geliştiriliyor
- NT5 Windows 2000 olarak çalışıyor
- XP
- Windows 8 10 11



İşletim Sistemleri

GUI

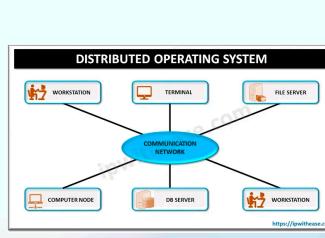
- Linux tarafı
- X window
- Gnome
- KDE



İşletim Sistemleri

Şimdi - Gelecek

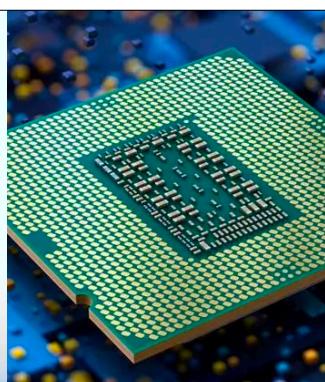
- Dağıtık işletim sistemleri
- Mobil işletim sistemleri



Donanım

İşlemci

- Bellekten alınan instruction'lar
- x64 - Arm farklı instruction setleri
- CPU register'ları
- Program counter
- Stack pointer
- Instruction pointer

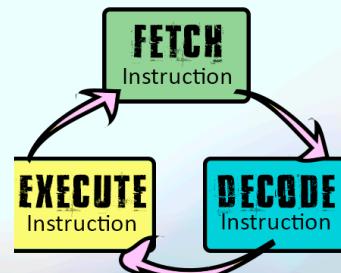


Bir program çalıştırıldığında belleğe çıkartılır ve bellekteki instruction'lar cpu üzerinde çalıştırılır. Instruction setleri işlemci mimarilerine göre değişim göstermektedir. İşlemci üzerinde temelde veriye hızlı erişim sağlayan ve ram'e göre küçük boyuta sahip cache bölümleri, instruction ve stack durumunu saklayan pointer değerlerinin saklandığı bölümler ve registerlar gibi bir çok farklı birim bulunmaktadır.

Donanım

İşlemci

- Fetch
- Decode
- Execute
- Birden fazla instruction yönetimi
- Pipeline
- Karmaşık Pipeline compiler yazanlar için sıkıntı



Donanım

İşlemci

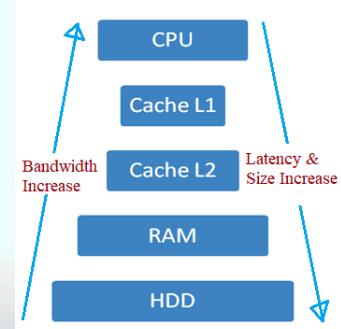
- Frekans artırmanın sınırı
- Moore kanunu
- Multithreading
- Gerçek paralelizm değil
- Çok hızlı thread-switching imkanı
- Multicore sistemler



Donanım

Bellek

- Bellek hızı büyük ve ucuz olmalı
- Üçünü sağlamak mümkün değil
- Register
- Cache
- Ana bellek
- Saklama belleği



İşlemcinin instruction elde ettiği fetch, bu instruction'ının çözümlendiği decode ve bu instruction'ın çalıştırıldığı execute olmak üzere üç işleminden oluşan bir döngü bulunmaktadır. Bu döngünün bir aşamasındayken diğer aşamaları gerçekleştiriren birimlerin boş kalmaması amaçlandığında bir instruction decode edilirken yenisinin fetch edilebileceği savunulmuş ve pipeline kavramı ortaya atılmıştır.

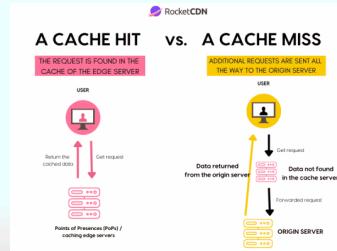
İşlemcilerin frekans artışındaki hızın kesildiği söylenebilir. İşlemlerin paralel olarak yapılması da hız artımına destek sağlamaktadır. Multithreading temelde tek bir işlemci ile gerçekleştirilebilir ve bu durumda gerçek paralelizmden bahsedilemez. Bunun yerine işlemci üzerinde her thread paralel çalışıyor izlenimi verecek hızda belirli aralıklarla değiştirilerek çalıştırılır.

Belleğin hızlı büyük ve ucuz olması istenir fakat bu üç özelliğin aynı anda sağlanması mümkün değildir. Bu sebeple daha küçük ve daha hızlı, daha büyük ve daha yavaş ve kalıcı veri saklama yapan farklı bellek formları bulunmaktadır. Örneğin register ve cache üzerinde okuma yazma işlemleri ram'e göre daha hızlı olurken boyutu ram'e göre oldukça küçüktür. Aynı şekilde ram disk'e göre çok hızlı olmasına rağmen kalıcı saklama imkanı sağlamamaktadır.

Donanım

Cache

- Cache miss
- Bellek ile haberleşme
- Etkin yönetimi gereklidir
- Bellek - disk arasında benzer bir durum mevcut
- L1 - L2 - L3
- Multicore işlemciler için L2
- Ortak
- Ayrı ayrı



Donanım

Bellek

- RAM
- ROM
- Flash Memory
- CMOS
- Virtual Memory



Donanım

Saklama belleği

- Teyp
- Disk
- Rpm
- Track - sektör
- PATA - SATA
- SSD

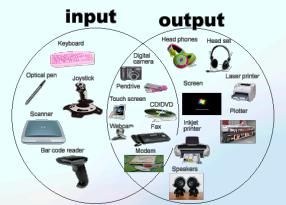


Cache içerisindeki veriye erişim hızlıdır fakat cache'in boyutu sınırlıdır. Cache içerisinde aynı anda olacak veri miktarı az olduğu için istenen bir verinin cache'de bulunmaması durumunda belleğe erişim ve bellekten cache'e veri okuma işlemi gerçekleştirilmelidir. Bu işlemin çok sık olması durumunda cache avantajı kaybedilmektedir. Aynı durum bellek ve disk arasında da (örn sanal bellek) yaşanmaktadır.

Donanım

IO

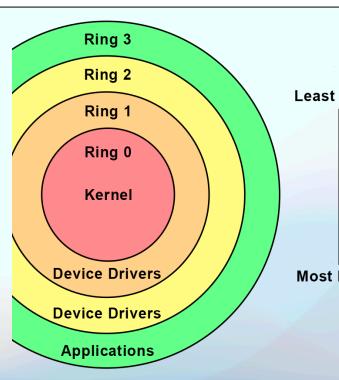
- IO cihazları cihaz ve kontrolörden oluşur.
- Kontrolörler farklı, bu yüzden farklı yazılımlara ihtiyaç duyarlar.
- Aygit sürücülerleri (Device driver)



Donanım

Device driver

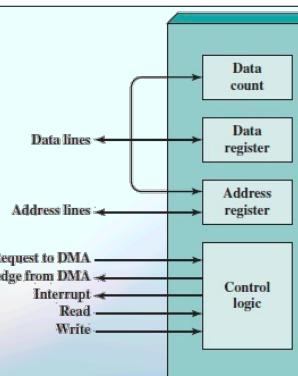
- Driver'lar işletim sisteminde 3 farklı şekilde bulunabilir.
- İşletim sisteminin içine dahil edilip edilmesi
- Driver'in yolunun işletim sistemine verilmesi ve yeniden başlatma
- Dinamik olarak driver ekleme



Donanım

IO çalışma prensibi

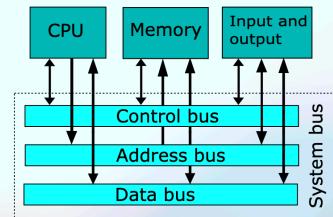
- Bir kullanıcı program sistem çağrısı yapar
- Kernel sürücüye çağrı yapar
- Driver gerekli işlemi başlatır.
- Bu noktada driver cevap bekleyebilir ya da cevap alacağı zaman interrupt isteyebilir



İşletim Sistemleri

BUS

- İşlemci ve bellek hızlandırdıça bus yetersiz kalmakta
- Bus sayısı artırılıyor
- PCIe
- DDR4
- DMI
- USB
- SATA



İşletim Sistemleri

Bilgisayarın açılması

- Bootstrapping için BIOS kullanılır.
 - UEFI
- Anakart power supply'dan cevap aldığında cpu çalışır
- PCI cihazları kontrol edilip başlatılır.
- Yeni cihazlar da dahil edilir.
- Bu noktada BIOS ve UEFI için farklı
 - Diskin başlatılması
 - MBR - GPT



İşletim sistemi türleri

Mainframe işletim sistemleri

- Batch processing
- Transaction processing
- Time sharing
- Z/OS
 - OS/390
 - OS/360
- Linux



Mainframe işletim sistemlerinde kullanıcıya etkin bir zaman vermek ilk amaç değildir. Bunun yerine amaç birden fazla işin en kısa sürede bitirilmesi ve bu sırada işlemin kendi içerisindeki çalışma zamanının da dengelenmesidir.

İşletim sistemi türleri

Server işletim sistemleri

- Büyük bilgisayarlar
- İş istasyonları
- Mainframe'ler
- Yazdırma, dosya, veritabanı veya web servisleri
- Linux
- Windows Server
- FreeBSD
- Solaris



Server işletim sistemlerinde amaç farkı servislere etkin bir zaman paylaşımı gerçekleştirmektir.

İşletim sistemi türleri

PC işletim sistemleri

- Günlük işler
- Daha basit cihazlar
- Windows
- macOS
- Linux
- FreeBSD



İşletim sistemi türleri

Mobil işletim sistemleri

- PDA
- Android
- iOS



İşletim sistemi türleri

IoT ve gömülü işletim sistemleri

- Sensör ağırlıklı
- Ağa bağlı olmayabilir
- Kullanıcı tanımlı uygulama içermez
- Olabildiğince basitleştirilmiş bir sistem
- Düşük güç
- RIOT, Embedded Linux, TinyOS

İşletim sistemi türleri

Gerçek zamanlı işletim sistemleri

- Belirli işlerin belirli kesinlikte yapılması gerekliliği durumlarda kullanılır
- Bir araç kontrol fabrikasında doğru zaman doğru proseslerin işlenmesi
- Bu örnekler hard real-time sistemler olarak adlandırılır.
- Medya oynatımı, akıllı telefonlar vb sistemler soft real-time olarak anılır.

Özellikle zaman sınırlı üretim veya sağlık işlemelerinde kritik zaman planlaması yapılması gerekmektedir. Herhangi bir parçanın işleminin aksaması durumunda tüm bandın aksaması veya sağlık sistemlerinde anlık verinin alınamaması gibi problemler ortaya çıkabilir. Medya oynatımı gibi durumlarda da realtime ihtiyacı bulunmakla beraber önceki örnekler gibi kritik değildir.

İşletim sistemi konseptleri

İşlemler

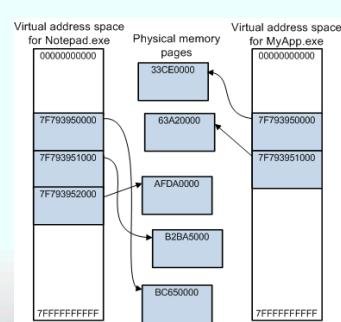
- Process
- Çalışan program
- Kendine ait adres alanı
- Process tablosu
- Process'ler arası haberleşme
- Senkronizasyon
- UID - GID

Bilgisayarda çalıştırılan her uygulama bir işlem olarak çalışır. Bu işlemlerin kendine ait adres alanı bulunmaktadır. Bu işlemlerin takibi için işletim sistemleri bir işlem tablosu saklar. Bu işlemlerin belirli durumlarda kendi aralarında haberleşmeleri gerekmektedir. Bu işlemlerin yönetimi işletim sisteminin görevidir.

İşletim sistemi konseptleri

Adres alanları

- Programın çalıştığı bellek
- Swap
- Birden fazla program bellekteyse
 - Güvenlik
- Gerçek bellekten fazla adresleme
- Gerçek bellekten fazla ihtiyaç
 - Sanal bellek



İşletim sistemi konseptleri

Dosyalar

- Blok, sektör ve sayfalar hala çok primitif
- Dizin / Klasör
- Ağacı yapısı
- Kök dizin
- Çalışma dizini (working directory)
- Dosya yetkileri
- Dosya türleri

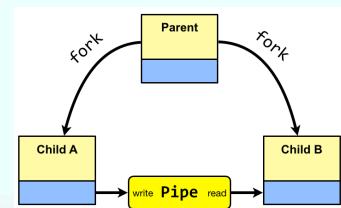


Verilerin nasıl saklanacağı ve hiyerarşik bir yapının nasıl kurulacağı da işletim sisteminin yönetimindedir. Aynı zamanda oluşturulan dosyaların erişim kontrolü ve güvenliği de işletim sisteminin üstlendiği görevlerdendir.

İşletim sistemi konseptleri

Dosyalar

- Bir saklama ortamının bağlanması
- Windows - Sürücü harfi
- UNIX - Linux
 - Mount
- Özel dosyalar
 - IO cihazlarının dosya olarak görülmesi
- Pipe
 - İki işlemi birbirine bağlar

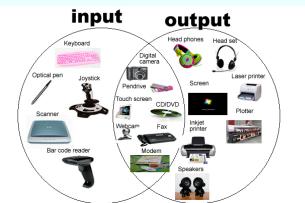


Bir cihaz işletim sistemine bağlılığında bu cihazın dosya sistemindeki yeri ve nasıl erişileceği de işletim sistemi tarafından kararlaştırılmalıdır.

İşletim sistemi konseptleri

IO

- Girdi / Çıktı
- Klavye, fare, monitör, yazıcı vb.
- Sürücüler



İşletim sistemi konseptleri

Güvenlik

- UNIX dosya güvenliği
 - rwx
 - User, group, other
- Diğer güvenlik mekanizmaları
 - TBA

drwxrwxrwx	7	rwx	111
d = Directory	6	rW-	110
r = Read	5	r-x	101
w = Write	4	r-	100
x = Execute	3	-wx	011
	2	-w-	010
	1	--x	001
	0	---	000

chmod 777

↓ ↓

rwx		rwx		rwx
Owner		Group		Others

İşletim sistemi konseptleri

Shell ve mekanizma değişikliği

- Shell işletim sistemi çekirdeğine dahil değil
- İşletim sistemlerinin temelde yaptığı şeyleri karşılar
- Belirli algoritmalar ve yöntemler donanımın değişmesi ile ortadan kalkabilir veya tekrar gündeme gelebilir

```
l-HP: ~
ls -l
de overide 4096 May 19 03:45 acadenv
de overide 4096 May 27 18:28 acadview_demo
de overide 4096 May 3 15:14 anaconda3
de overide 4096 May 31 16:49 Desktop
de overide 40966 Jun 1 13:09 Downloads
de overide 8980 Aug 8 2016 examples.deskt
de overide 45065 May 28 01:40 hs_err_pid1971
de overide 45147 Jun 1 03:24 hs_err_pid2006
de overide 4096 Mar 2 18:22 Music
de overide 4096 Dec 25 00:13 Mydata
de overide 4096 Sep 20 2016 newbin
de overide 4096 Dec 20 22:44 nltk_data
de overide 4096 May 31 20:46 Pictures
de overide 4096 Aug 8 2016 Public
de overide 4096 May 31 19:49 scripts
de overide 4096 Aug 8 2016 Templates
de overide 4096 Feb 14 11:22 test
de overide 4096 Mar 11 13:27 Videos
de overide 4096 Sep 1 2016 xdm-helper
```

Bilgisayara bağlanan her cihazın yönetimini de işletim sistemi üstlenir.

Dosya güvenliği ve bu dosyalara erişimin belirlenmesi her işletim sisteminde farklı bir şekilde ele alınmaktadır. Örneğin unix üzerinde bir dosyanın okuma yazma ve çalışma yetkileri, kullanıcı, grup ve diğer olmak üzere üç farklı grup için tanımlanabilir. Bu işlemlerin detayları ilerleyen haftalarda irdelenenecektir.

Shell işletim sistemi çekirdeğinin bir parçası olmamakla beraber işletim sisteminin temel işlevlerine erişim için bir arayüz sağlar. Aynı zamanda işletim sistemleri farklı zamanlarda farklı donanımlar üzerinde çalıştığı için her dönem ihtiyaç duyulan durumlara göre farklı işlevleri üstlenebilir veya bu işlevleri bırakabilirler. Örneğin işlem gücü ve güç sınırı bulunmazken işletim sisteminin görevi %100 kaynak kullanımıyken, mobil cihazlarda pil yönetiminin devreye girmesi ile işletim sisteminin amacı minimum güç ile işlem yapmaya dönüşebilir.

İşletim sistemi konseptleri

Sistem çağrıları

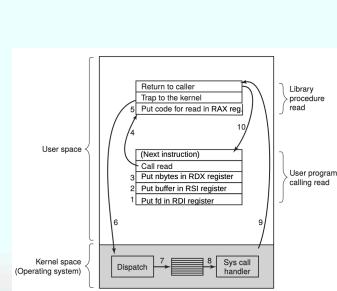
- Kullanıcı uygulamalarının kaynakları işletim sistemi üzerinden kullanılabilmesi için
- Her işletim sisteminde farklı
 - UNIX - POSIX
 - İşlemler
 - Dosyalama
 - Dizin işlemleri



İşletim sistemi konseptleri

Sistem çağrıları

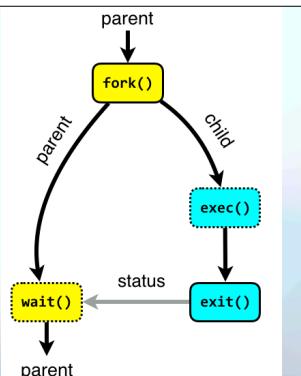
- Bir sistem çağrısının süreci
- Trap instruction
- Kernel mode'a geçiş
- Sabit bir adrese yönlendir
- Çağırılan işlem blokları
- Bu sırada başka işleme geçilebilir



İşletim sistemi konseptleri

Fork

- Bir işlem tarafından çağrırlar
- Çağırılan işlemin bir kopyası olarak oluşturulur.
- Çağırılan işleme yönlendirilir
- Ana işlem waitpid ile bekler
- İşlem bittiğinde ana işleme geri dönülür



İşletim sisteminin çekirdek moduna geçilmesi ve kaynakların kullanımı için sistem çağrıları kullanılmaktadır. Farklı işletim sistemleri bu çağrıları farklı şekillerde gerçekleştirirler. Çağrılar dosyalama işlem kontrolü, cihaz erişimi gibi farklı işler için gerçekleştirilebilirler.

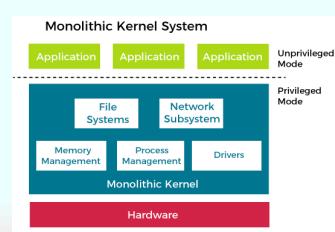
Sistem çağrısı yapıldığında trap instruction'ı çağrırlar ve çekirdek moduna geçilir. Daha sonra bu çağrıyı yapan işlem durdurulur ve çekirdek uzayı üzerinde istenilen işlem gerçekleştirip cevap işleme dönülür. Bu sırada işletim sistemi işlemciyi başka bir işleme tahsis edebilme yetkisine sahiptir.

Bir işlem başka bir işlem oluşturuyorsa bu işleme fork adı verilir. Bu durumda çağrılan işlemin bir kopyası oluşturulur ve bu işleme yönlendirilir. Bu sırada bu işlemi çağıran üst işlem iste beklemeye alınır. Alt işlemin işi bittiğinde ana işleme geri dönülür.

İşletim sistemi mimarisi

Monolitik sistemler

- Tek büyük bir çalıştırılabilir program
- Tüm prosedürler tek bir uygulama içerisinde
- Bir prosesin çöküşü tüm sistemin çöküşü
- Dışarıdan bir ekleme söz konusu değil



İşletim sistemi çekirdekleri farklı mimarilere sahip olabilirler.

İşletim sistemi mimarisi

Katmanlı sistemler

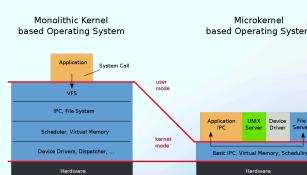
- Her katmanın farklı bir işi üstlenmesi
- Her katman için ayrı bir grup
- MULTICS

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

İşletim sistemi mimarisi

Mikro çekirdek

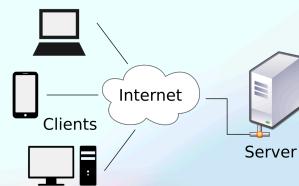
- Microkernel
- Her modülün kendi işini yapması
- Yalnızca microkernel'in kernel mode'da çalışması
- Kalan tüm uygulamaların user mode'da çalışması
- Bir bileşenin çökmesinin işletim sistemini etkilememesi
- macOS hariç desktop tarafında kullanılmıyor
- Kritik sistemler için daha uygun



İşletim sistemi mimarisi

İstemci - Sunucu modeli

- Sunucu
- İstemci
- Mesajlaşma
- LAN ve WAN üzerinden haberleşme



İşletim sistemi mimarisi

Sanal makineler

- Host
- Client
- Tip 1
- Tip 2
- Java VM
- Container

