

İşletim Sistemleri

Binnur Kurt
binnur.kurt@ieee.org

İstanbul Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü



Copyright © 2005

Version 0.0.1

About the Lecturer



- BSc
İTÜ, Computer Engineering Department, 1995
- MSc
İTÜ, Computer Engineering Department, 1997
- Areas of Interest
 - Digital Image and Video Analysis and Processing
 - Real-Time Computer Vision Systems
 - Multimedia: Indexing and Retrieval
 - Software Engineering
 - OO Analysis and Design

2

Önemli Bilgiler

□ Dersin

- Gün ve Saati
 - 18:30-21:30 Cuma
- Adresi
 - <http://www.cs.itu.edu.tr/~kurt/Courses/os>
- E-posta
 - kurt@ce.itu.edu.tr

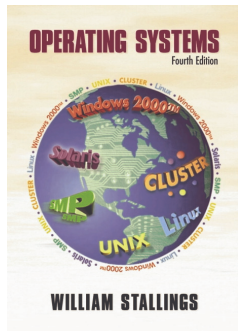
3

Notlandırma

- 3 Ödev (30%)
- Yıllık Sınavı (30%)
- Final Sınavı (40%)

4

Kaynakça



5

*Tell me and I forget.
Show me and I remember.
Let me do and I understand.*

—Chinese Proverb

6

İçerik

1. Giriş.
2. Prosesler ve Proses Kontrolü.
3. İplikler
4. Prosesler Arası İletişim
5. Ölümcül Kilitlenme
6. İş Sıralama
7. Bellek Yönetimi
8. Giriş/Çıkış Yönetimi
9. Dosya Sistemi

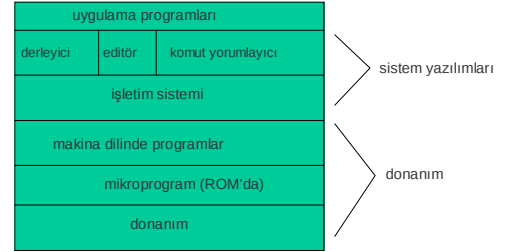
1

GİRİŞ

İşletim Sistemi

- donanımı kullanılabilir yapan yazılım
 - bilgisayar kaynaklarını:
 - denetler,
 - paylaşır
- üzerinde program geliştirme ve çalışma ortamı
- çekirdek (kernel) = işletim sistemi

Bilgisayar Sistemi



İşletim Sistemi

- güncel işletim sistemleri doğrudan donanıma erişmeyi engeller
 - kullanıcı modu × çekirdek modu
- donanımın doğrudan kullanımının zorluklarını gizler
- kullanıcı ve donanım arasında arayüz
 - sistem çağrılar

Sistem Çağrıları

- kullanıcı programların
 - işletim sistemi ile etkileşimi ve
 - işletim sisteminden iş isteği için
- her sistem çağrısına karşılık kütüphane rutini
- kullanıcı program kütüphane rutinini kullanır

İşletim Sisteminin Temel Görevleri

- kaynak paylaşımı
- görüntü makina sağlanması

Giriş 1

Kaynak Paylaşımı

- kullanıcılar arasında paylaşım
- güvenlik
 - kullanıcıları birbirinden yalıtır
- paylaşılan kaynaklar:
 - işlemci
 - bellek
 - G / Ç birimleri
 - veriler

Giriş 1

Kaynak Paylaşımı

- amaçlar:
 - kaynakların kullanım oranını yükseltmek (utilization)
 - bilgisayar sisteminin kullanılabilirliğini arttırmak (availability)

Giriş 1

Kaynak Paylaşımı

- verdiği hizmetler:
 - kullanıcı arayüzünün tanımlanması
 - sistem çağrıları
 - çok kullanıcı sistemlerde donanımın paylaşılması ve kullanımın düzenlenmesi
 - kaynaklar için yarış önlemek
 - birbirini dışlayan kullanım
 - kullanıcıların veri paylaşımını sağlamak (paylaşılan bellek bölgeleri)
 - kaynak paylaşımının sıralanması (scheduling)
 - G/Ç işlemlerinin düzenlenmesi
 - hata durumlarından geri dönüş

Giriş 1

Kaynak Paylaşımı

- örnek:
 - yazıcı paylaşamaz; bir kullanıcının işi bitince diğeri kullanılabilir
 - ekranda paylaşım mümkün

Giriş 1

Görüntü Makina Sağlanması

- donanımın kullanılabilir hale getirilmesi
- kullanıcı tek başına kullanıyormuş gibi
 - kaynak paylaşımı kullanıcıya şeffaf
- görüntü makinanın özellikleri fiziksel makinadan farklı olabilir:
 - G/Ç
 - bellek
 - dosya sistemi
 - koruma ve hata kotarma
 - program etkileşimi
 - program denetimi

Giriş 1

Görüntü Makina Sağlanması

► G/Ç

- donanımına yakın programlama gerekir
- işletim sistemi kullanımı kolaylaştırır
 - aygıt sürücüler
- örnek: diskten / disketten okuma

Giriş 1

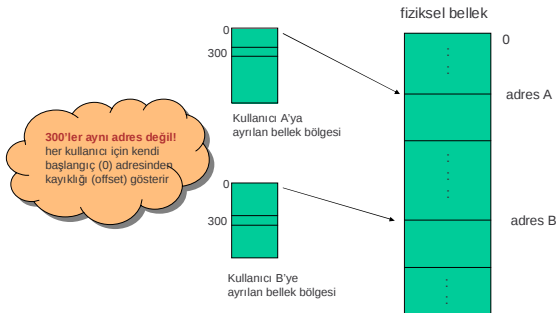
Görüntü Makina Sağlanması

► Bellek

- fiziksel bellekten farklı kapasitede görüntü makina
 - disk de kullanılarak daha büyük
 - kullanıcılar arasında paylaştırılarak daha küçük
 - her kullanıcı kendine ayrılan bellek alanını görür

Giriş 1

Görüntü Makina Sağlanması



Giriş 1

Görüntü Makina Sağlanması

► Dosya sistemi

- program ve verilerin uzun vadeli saklanması için
- disk üzerinde
- bilgilere erişimde fiziksel adresler yerine simgeler kullanımı
 - isimlendirme
 - UNIX işletim sisteminde herşey dosya

Giriş 1

Görüntü Makina Sağlanması

► Koruma ve hata kotarma

- çok kullanıcıli sistemlerde kullanıcıların birbirlerinin hatalarından etkilenmemesi

Giriş 1

Görüntü Makina Sağlanması

► Program etkileşimi

- çalışma anında programların etkileşmesi
 - örneğin birinin ürettiği çıkış diğerine giriş verisi olabilir

Giriş 1

Görüntü Makina Sağlanması

- Program denetimi
 - kullanıcıya yüksek düzeyli bir komut kümesi
 - kabuk (shell) komutları
 - kabuk: komut yorumlayıcı
 - kabuk işletim sistemi içinde değil
 - ama sistem çağrılarını yoğun kullanır

İşletim Sistemi Türleri

- Anaçatı işletim sistemleri (mainframe)
- Sunucu (server) işletim sistemleri
- Çok işlemcili işletim sistemleri
- Kişisel bilgisayar işletim sistemleri
- Gerçek zamanlı (real-time) işletim sistemleri
- Gömülü (embedded) işletim sistemleri
- Akıllı-kart (smart card) işletim sistemleri

Anaçatı İşletim Sistemleri

- yoğun G/Ç işlemi gerektiren çok sayıda görev çalıştırmaya yönelik
- üç temel hizmet:
 - batch modda çalışma
 - etkileşimsiz, rutin işler
 - örneğin bir sigorta şirketindeki sigorta tazminatı isteklerinin işlenmesi
 - birim-iş (transaction) işleme
 - çok sayıda küçük birimler halinde gelen isteklere yanıt
 - örneğin havayollarında rezervasyon sistemi
 - zaman paylaşımı çalışma
 - birden fazla uzaktan bağlı kullanıcının sistemde iş çalıştırması
 - örnek: veri tabanı sorgulaması
- Örnek: OS/390

Sunucu İşletim Sistemleri

- sunucular üzerinde çalışır
 - büyük kaynak kapasiteli kişisel bilgisayarlar
 - iş istasyonları
 - anaçatı sistemler
- bilgisayar ağı üzerinden çok sayıda kullanıcıya hizmet
 - donanım ve yazılım paylaşım
 - örneğin: yazıcı hizmeti, dosya paylaşım, web erişimi
- örnek: UNIX, Windows 2000

Çok İşlemcili İşletim Sistemleri

- birden fazla işlemcili bilgisayar sistemleri
- işlem gücünü artırma
- işlemcilerin bağlantı türüne göre:
 - paralel sistemler
 - birbirine bağlı, birden fazla bilgisayardan oluşan sistemler
 - çok işlemcili sistemler
- özel işletim sistemi gerek
 - temelde sunucu işletim sistemlerine benzer tasarım hedefleri
 - işlemciler arası bağlaşım ve iletişim için ek özellikler

Kişisel Bilgisayar İşletim Sistemleri

- kullanıcıya etkin ve kolay kullanılabilir bir arayüz sunma amaçlı
- genellikle ofis uygulamalarına yönelik
- örnek:
 - Windows 98, 2000, XP
 - Macintosh
 - Linux

Gerçek Zamanlı İşletim Sistemleri

- ▶ zaman kısıtları önem kazanır
- ▶ endüstriyel kontrol sistemleri
 - toplanan verilerin sisteme verilerek bir yanıt üretilmesi (geri-besleme)
- ▶ iki tip:
 - katı-gerçek-zamanlı (hard real-time)
 - zaman kısıtlarına uyulması zorunlu
 - örneğin: araba üretim bandındaki üretim robotları
 - gevşek-gerçek-zamanlı (soft-real-time)
 - bazı zaman kısıtlarına uyulmaması mümkün
 - örneğin: çoğulortam sistemleri
- ▶ örnek: VxWorks ve QNX

Gömülü İşletim Sistemleri

- ▶ avuç-ıçi bilgisayarlar ve gömülü sistemler
- ▶ kısıtlı işlevler
- ▶ özel amaçlı
- ▶ örneğin: TV, mikrodalga fırın, cep telefonları, ...
- ▶ bazı sistemlerde boyut, bellek ve güç harcama kısıtları var
- ▶ örnek: PalmOS, Windows CE

Akıllı-Kart İşletim Sistemleri

- ▶ en küçük işletim sistemi türü
- ▶ kredi kartı boyutlarında, üzerinde işlemci olan kartlar üzerinde
- ▶ çok sıkı işlemci ve bellek kısıtları var
- ▶ bazıları tek işleve yönelik (örneğin elektronik ödemeler)
- ▶ bazıları birden fazla işlev içerebilir
- ▶ çoğunlukla özel firmalar tarafından geliştirilen özel sistemler
- ▶ bazıları JAVA tabanlı (JVM var)
 - küçük JAVA programları (applet) yüklenip çalıştırılır
 - bazı kartlar birden fazla program (applet) çalıştırabilir
 - çoklu-programlama, iş sıralama ve kaynak yönetimi ve koruması

Temel İşletim Sistemi Yapıları

- ▶ Monolitik
- ▶ Katmanlı
- ▶ Sanal Makinalar
- ▶ Dış-çekirdek (exo-kernel)
- ▶ Sunucu-İstemci Modeli
- ▶ Modüller

Monolitik İşletim Sistemleri

- ▶ genel bir yapı yok
- ▶ işlevlerin tamamı işletim sistemi içinde
- ▶ işlevleri gerçekleyen tüm prosedürler
 - aynı seviyede
 - birbirleri ile etkileşimli çalışabilir
- ▶ büyük

Modüler Çekirdekli İşletim Sistemleri

- ▶ çekirdek minimal
- ▶ servisler gerektiğinde çalışma anında modül olarak çekirdeğe eklenir
 - örneğin aygıt sürücüler
- ▶ küçük çekirdek yapısı
- ▶ daha yavaş
- ▶ örnek: LINUX

Katmanlı Yapılı İşletim Sistemleri

- ▶ işletim sistemi katmanlı
 - hiyerarşik
- ▶ örnek: THE işletim sistemi

5	operatör
4	kullanıcı programları
3	G/Ç yönetimi
2	operatör-proses iletişimi
1	bellek ve tambur yönetimi
0	işlemci paylaşırma ve çoklu-programlama

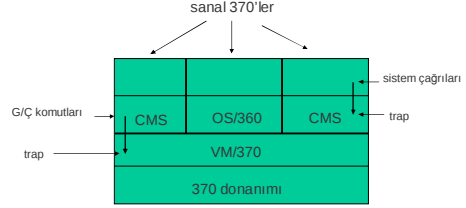
- katman 0 işlemciyi prosesler arası paylaştırır (iş sıralama)
- katman 1 bellek yönetimini yapar (bellek ve tambur arası)
- ...

Her katman altındaki işlemleriyle ilgilenmez.
Örnek: 2. katmandaki işlemler için prosesin bellek veya tamburda olması önemli değil.

Sanal Makina

- ▶ VM/370

- VM donanım üzerinde koçar
- çoklu programlama yapar
- birden fazla sanal makina sunar
- sanal makinaların her biri donanımın birebir kopyası
- her sanal makina farklı işletim sistemi olabilir



Dış-Çekirdek (Exo-Kernel)

- ▶ MIT'de geliştirilmiş
- ▶ sanal makina benzeri
 - sistemin bir kopyasını sunar
 - fark: her sanal makina kaynakların birer alt kümesini tahsis eder
 - dönüşüm gerekmez; her makina ayrılan kaynakların başı-sonu belli
- ▶ dış çekirdek var
 - görevi: sanal makinaların kendilerine ayrılan kaynaklar dışına çıkmamasını kontrol eder
- ▶ her sanal makina farklı işletim sistemi olabilir

Sunucu-İstemci Modeli

- ▶ çekirdek minimal (mikro-çekirdek)
- ▶ işletim sisteminin çoğu kullanıcı modunda
- ▶ sunumcular ve istemci prosesler var
 - örneğin dosya okuma işlemi
 - istemci proses sunucudan ister
 - sunucu işlemi yürütür
 - yanıtı istemciye verir
- ▶ çekirdek sunucu ve istemciler arası iletişimi yönetir

Sunucu-İstemci Modeli

- ▶ sunucular kullanıcı modunda
 - dosya sunucusu
 - proses sunucusu
 - terminal sunucusu
 - bellek sunucusu
- ▶ işletim sistemi alt birimlerden oluştuğundan:
 - yönetimi kolay
 - bir birimdeki hata tüm sistemi çökertmez (birimler donanıma doğrudan ulaşamaz)
 - gerçekleştirilmede sorunlar: özellikle G/Ç aygıtlarının yönetiminin tamamen kullanıcı düzeyinde yapılması mümkün değil
- ▶ dağıtık sistemlerde kullanılmaya çok elverişli yapı

Sunucu-İstemci Modeli

