

İşletim Sistemleri

Binnur Kurt
binnur.kurt@ieee.org

İstanbul Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü



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

Önemli Bilgiler

❑ Dersin

➤ Gün ve Saati

- 18:30-21:30 Cuma

➤ Adresi

- <http://www.cs.itu.edu.tr/~kurt/Courses/os>

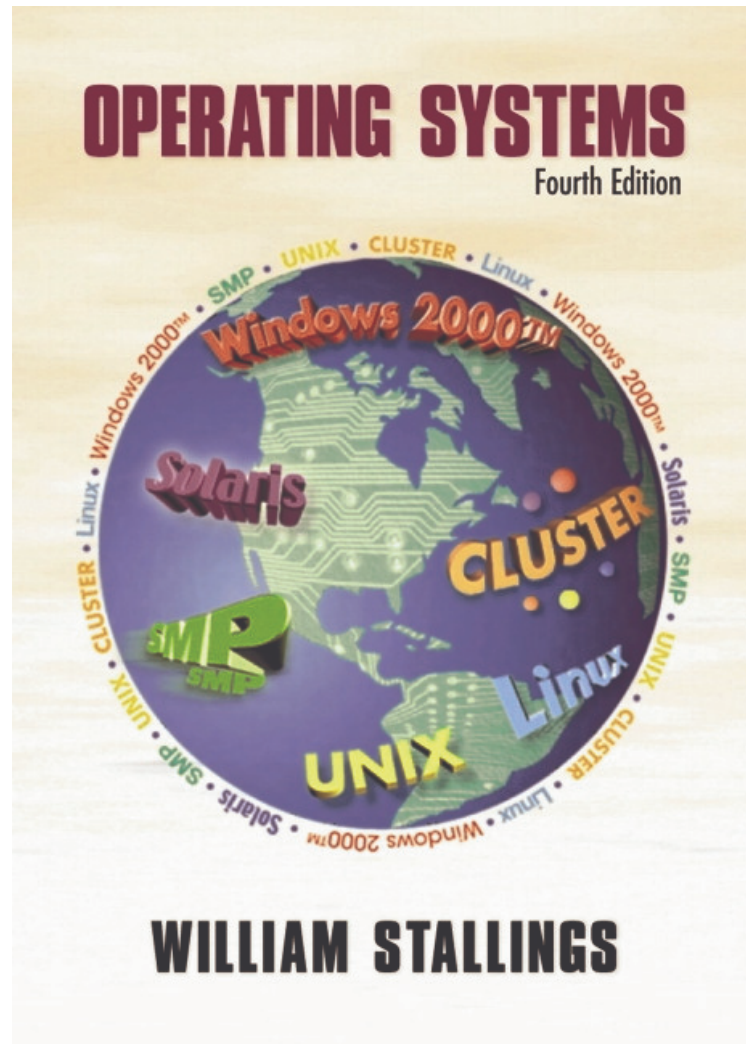
➤ E-posta

- kurt@ce.itu.edu.tr

Notlandırma

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

Kaynakça



Tell me and I forget.

Show me and I remember.

Let me do and I understand.

—Chinese Proverb

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

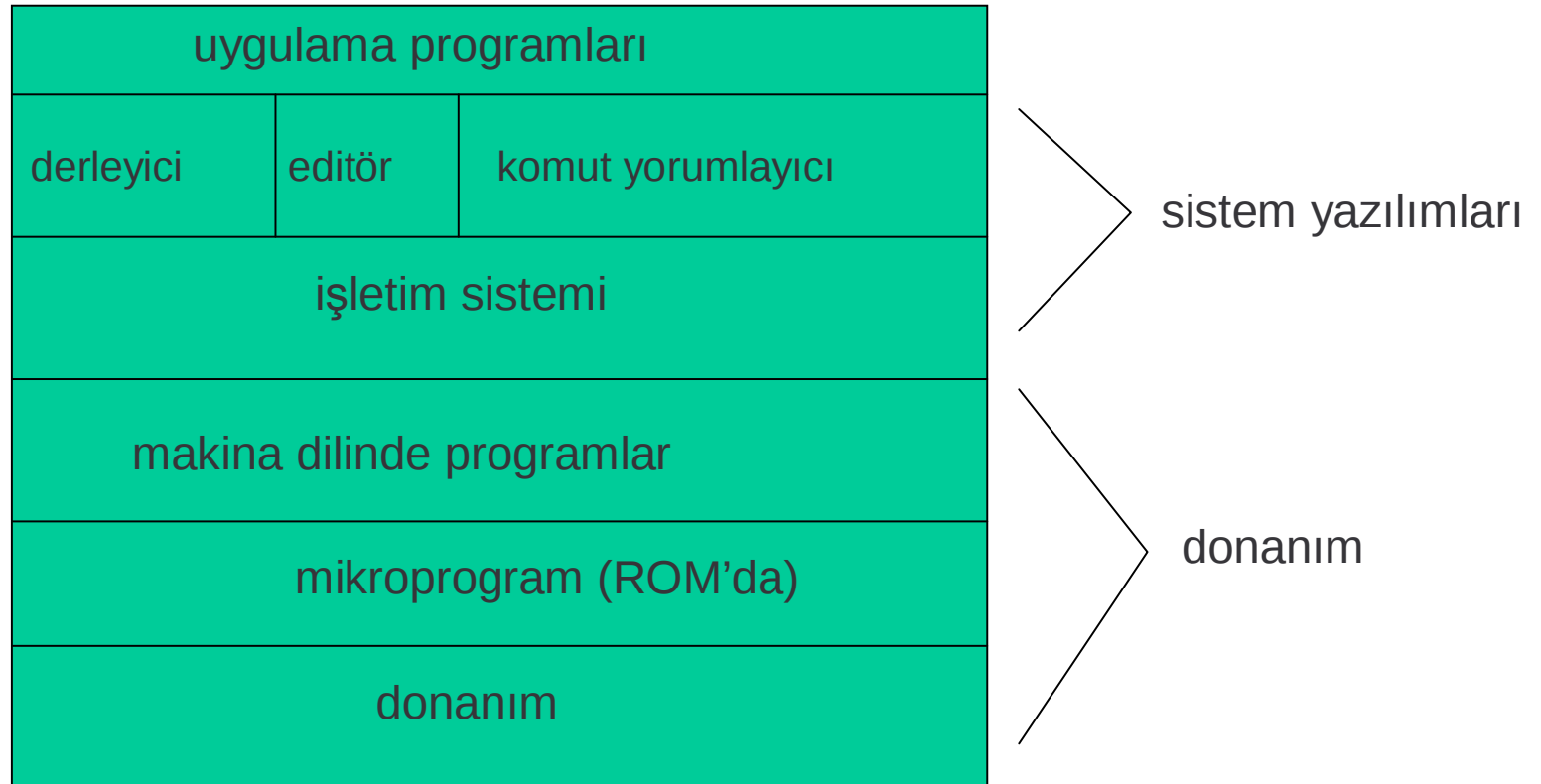
1

GİRİŞ

İşletim Sistemi

- ▶ donanımı kullanılabilir yapan yazılım
 - bilgisayar kaynaklarını:
 - denetler,
 - paylaştırır
- ▶ üzerinde program geliştirme ve çalıştırma 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 çağrıları

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

Kaynak Paylaşımı

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

Kaynak Paylaşımı

► amaçlar:

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

Kaynak Paylaşımı

- ▶ verdiği hizmetler:
 - kullanıcı arayüzünün tanımlanması
 - sistem çağrıları
 - çok kullanıcıli 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

Kaynak Paylaşımı

► örnek:

- yazıcı paylaşılabilir; bir kullanıcının işi bitince diğeri kullanabilir
- ekranda paylaşım mümkün

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

Görüntü Makina Sağlanması

► G/Ç

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

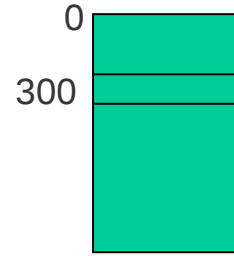
Görüntü Makina Sağlanması

► Bellek

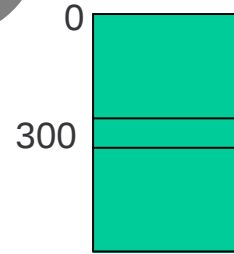
- fiziksel bellekten farklı kapasitede görüntü makina
 - disk de kullanılarak daha büyük
 - kullanıcılar arasında paylaştırılarak daha küçük
 - her kullanıcı kendine ayrılan bellek alanını görür

Görüntü Makina Sağlanması

300'ler aynı adres değil!
her kullanıcı için kendi
başlangıç (0) adresinden
kayıklığı (offset) gösterir

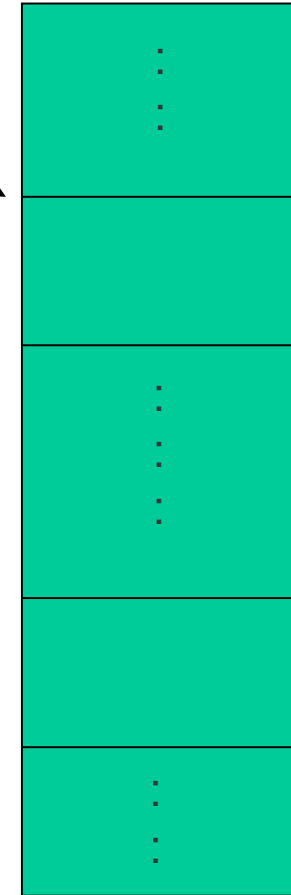


Kullanıcı A'ya
ayrılan bellek bölgesi



Kullanıcı B'ye
ayrılan bellek bölgesi

fiziksel bellek



0

adres A

adres B

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

Görüntü Makina Sağlanması

- Koruma ve hata kotarma
 - çok kullanıcıli sistemlerde kullanıcıların birbirlerinin hatalarından etkilenmemesi

Görüntü Makina Sağlanması

► Program etkileşimi

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

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 paylaşımli 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 paylaşırma
 - örneğin: yazıcı hizmeti, dosya paylaşırma, web erişimi
- ▶ örnek: UNIX, Windows 2000

Çok İşlemcili İşletim Sistemleri

- ▶ birden fazla işlemcili bilgisayar sistemleri
- ▶ işlem gücünü arttırma
- ▶ 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 kullanılır 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-ıç i 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
- ▶ Modüler

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

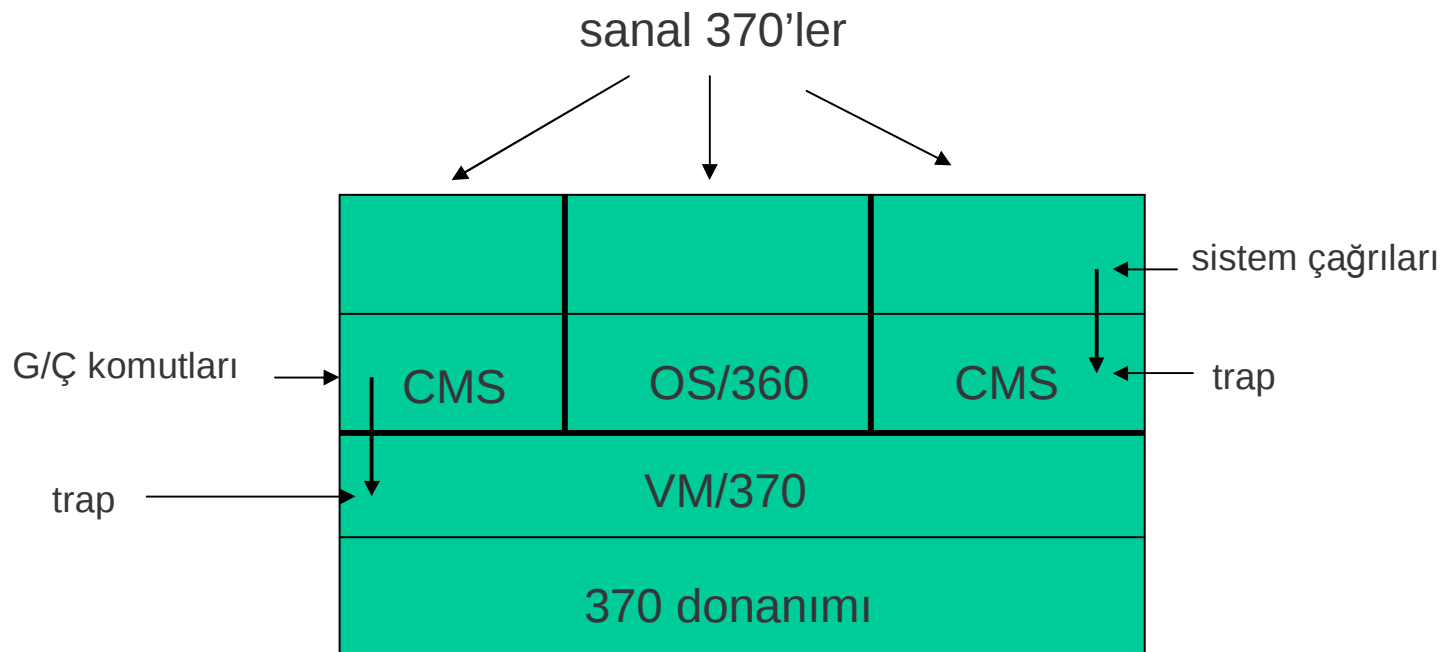
- katman 0 işlemciyi prosesler arası paylaşır (iş sıralama)
- katman 1 bellek yönetimini yapar (bellek ve tambur arası)
- ...

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

Sanal Makina

► VM/370

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



Dış-Çekirdek (Exo-Kernel)

- ▶ MIT’de geliştirilmiş
- ▶ sanal makina benzeri
 - sistemin bir kopyasını sunar
 - fark: her sanal makinaya kaynakların birer alt kümesini tahsis eder
 - dönüşüm gerekmez; her makinaya ayrılan kaynakların başı-sonu belli
- ▶ dış çekirdek var
 - görevi: sanal makinaların kendilerine ayrılan kaynaklar dışına çıkmamasını kontrol eder
- ▶ her sanal makinada 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 sisemi alt birimlerden oluştuğundan:
 - yönetimi kolay
 - bir birimdeki hata tüm sistemi çökertmez (birimler donanıma doğrudan ulaşamaz)
 - gerçeklemede sorunlar: özellikle G/Ç aygıtlarının yönetiminin tamamen kullanıcı düzeyinde yapılması mümkün değil
- ▶ dağıtık sistemlerde kullanılmaya çok elverişli yapı

Sunucu-İstemci Modeli

Giriş 1

