

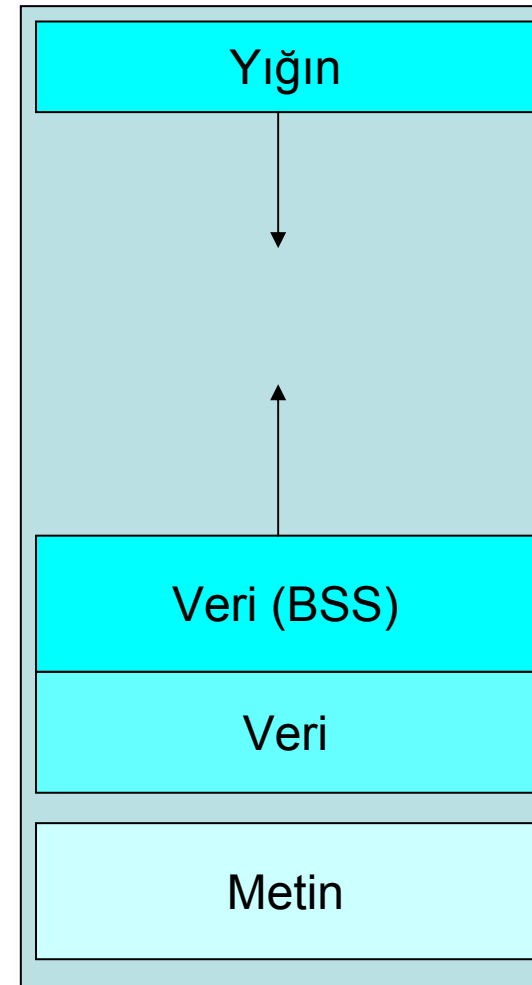
# Bilgisayar İşletim Sistemleri

## Uygulama 9

Unix'de Bellek Yönetimi

# Temel Kavramlar

- Metin: Değişmeyen boyutta program kodu
- Veri: İlk değerleri belirlenmiş boyutu değişmeyen değişkenler
- BSS: İlk değeri belli olmayan değişkenler
- Yığın



# Temel Kavramlar

- Bazı ortak alanları (segmanları) paylaşmak mümkün
- Bazı donanımlar ise program ve veri belleğini ayırmıştır
  - Veri paylaşımı daha kolay
- Bellek haritalı dosyalar (memory mapped files)
  - Ortak kütüphane dosyaları

# Unix'de Bellek Yönetimi

- System V ve BSD' den (Berkeley Software Distribution) beri pek değişmedi
- Yer değiştirmeye (swapping) dayalı algoritması oldukça güçlü; buna göre bir proses
  - Ya bellekte
  - Yada diskte bulunabilir

# Yer Değiştirme

- Çekirdek, bellekte boş alan kalmadığını fark ettiğinde bu işlemi şu koşullar altında başlatır:
  - Fork sistem çağrısı sonucu bir proses yaratılacaktır
  - brk sistem çağrısı sonucu veri segmentinin büyütülmesi gerekmektedir
  - Yığın çok büyümüştür ve ona ayrılan yeri doldurmuştur

# Yer Değiştirme

- Bunlara ek olarak diskte çok uzun zamandır bekleyen bir prosesi tekrar belleğe almak için de yer değiştirme işlemi başlatılır

# Kim Diske, Kim Belleğe?

- İlk önce askıda olan proseslere bakılır
- Birden fazla varsa “öncelik+bekleme süresi” yüksek olan seçilir
- Eğer böyle bir proses yoksa, halihazırda çalışan bir proses belli bir kurala göre seçilir

# Kim Nereye?

- Belli aralıklarla sistem, diskte olanlar arasında çalışmaya hazır bir proses var mı diye bakar
- Varsa, en uzun zamandır bekleyen seçilir ve yer değiştirmenin kolay mı zor mu olacağını belirler (easy & hard swap)
  - Kolay olan: bellekte zaten yeterince yer var
  - Zor olan: bir yada daha fazla prosesi diske alarak bellekte yer açmak lazım



# Kim Nereye?

- Bu algoritma şu iki şarttan biri gerçekleşene kadar devam eder
  - Diskte çalışmaya hazır proses kalmamıştır
  - Daha henüz taşınmış o kadar “taze” proses vardır ki bellekte yer kalmamıştır. Bu durumda bir proses 2 saniye bellekte kalmadan diske tekrar alınmaz.

# Unix'de Sayfalama

- Tüm Unix ailesi yer değiştirme sistemi kullanır ancak büyük proseslerin daha etkin çalışmasını sağlamak için sisteme sayfalama eklenmiştir.
- Temel iddia: bir programın çalışması için tüm kodun bellekte bulunmasına gerek yok.
- Gereken
  - Gerekli yapı
  - Sayfalama tablosu

# Sayfalama

- Kod metni
- Veri
- Yığın
- Referans verildikçe belleğe dinamik olarak taşınırlar
- Sayfalama
  - Çekirdek ve
  - Page Daemon tarafından gerçekleştirir

Swapper 0

Init 1

Page daemon 2

# Sayfa Değiştirme Algoritması

- 250 milisaniyede bir belleğin  $\frac{1}{4}$  oranında boş olup olmadığını kontrol eder
- Yer yoksa sayfalar diske alınır
- Yer varsa uyumaya devam eder
- Bu daemon saat algoritmasının bir versiyonunu kullanır
  - Kimin sayfası olduğu ile ilgilenmez

# Sayfa Değiştirme Algoritması

- İlk turda kullanım biti sıfırlanır
- İkinci turda sayfaya erişim olmamışsa diske alınır ve boş diye işaretlenir
  - Sayfanın üzerine bir şey yazılmaz
- Eğer bellek çok büyükse turlar arası çok fazla zaman aralığı oluşur

# İki Kollu Saat Algoritması

- İki işaretçi peş peşe sayfaları tarar.
- Aralarında çok fazla fark yoksa tek kollu saat algoritmasına dönüşür
- Eğer sayfalama oranı çok fazlaysa yer değiştirme algoritması uyandırılır ve bir veya daha fazla proses diske alınır