İşletim Sistemleri

Bellek Yönetimi

Emir ÖZTÜRK

Bellek Yönetimi

Bellek türleri

- Ucuz, büyük, hızlı
- Bellek seviyeleri
- Geçici / Kalıcı
- Yazılabilir / Okunabilir
- Bellek yönetimi

Bellek Yönetimi

Bellek soyutlaması olmaması

- En temel bellek yönetimi belleğin soyutlamasının yapılmamasıdır
- Her işlem belleğe direkt erişim sağlar
- Böyle bir durumda aynı anda bellekte yalnızca 1 program bulunabilir
- İki işlem bellekte her yere erişebileceğinden birbirlerinin verisini bozabilirler
- Her işlem işletim sisteminin bulunduğu belleğe erişebilir
    - İşletim sisteminin çökmesi de mümkün

Bellek Yönetimi

Bellek soyutlama olmadan birden fazla program çalıştırma

- Bellek belirli aralıklarla parçalara bölünür
- Her işlem farklı bölgeye paylaştırılır
- İşletim sistemi her programa belirli bölgeleri ayırabilir
- Burada işletim sisteminin çakışacak alanları vermemesi gerekir
- Çözüm
  - Her işlemin kendine ait erişilemez bir alanı olması

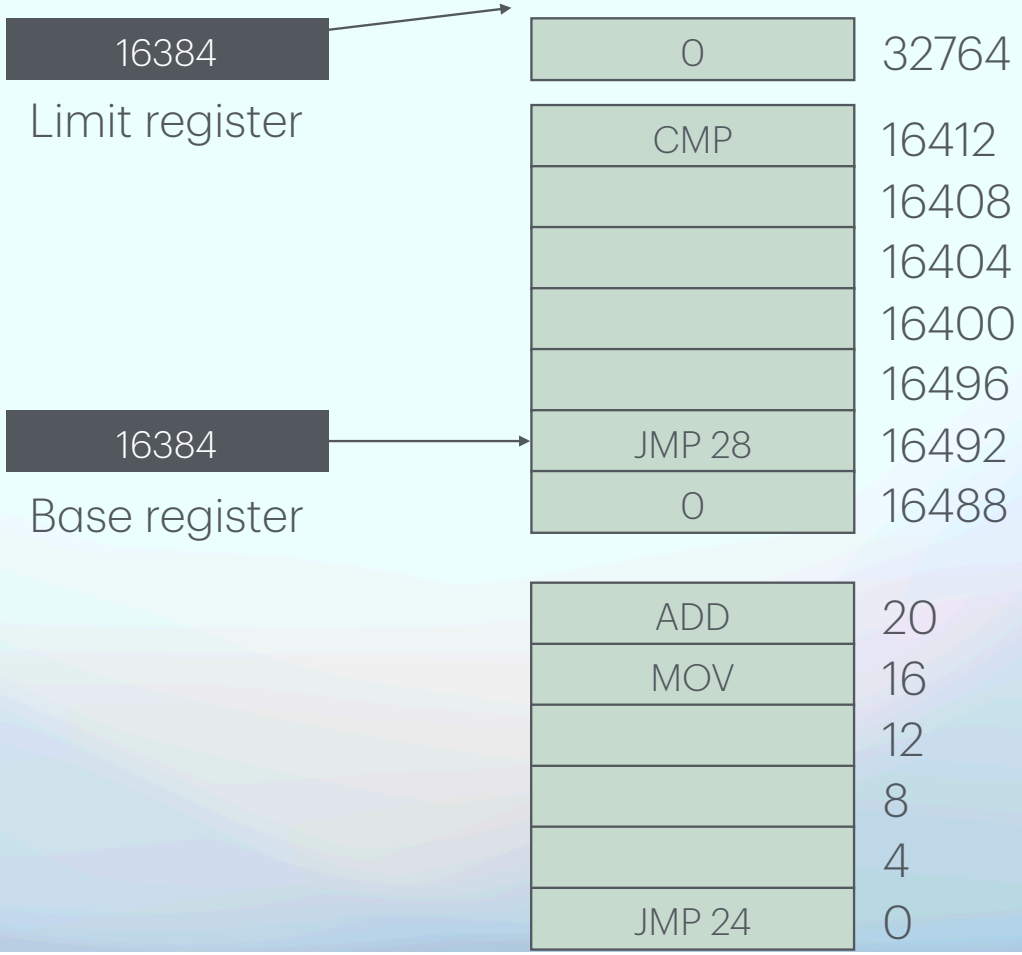
Bellek Yönetimi

Adres uzayları

- Her işleme ayrı bir bellek alanı ayrılması ve bu alanların işlemler arasında gizli olması gerekir
- Bu durumda çözülmesi gereken iki problem bulunmaktadır.
  - Korumanın nasıl olacağı
  - Yer değişikliği durumunun nasıl çözüleceği
- Adres uzayları bu sorunlara çözüm getirmektedir
  - Telefon numarasının alan kodu örneğinde olduğu gibi

Bellek Yönetimi

Adres uzayları



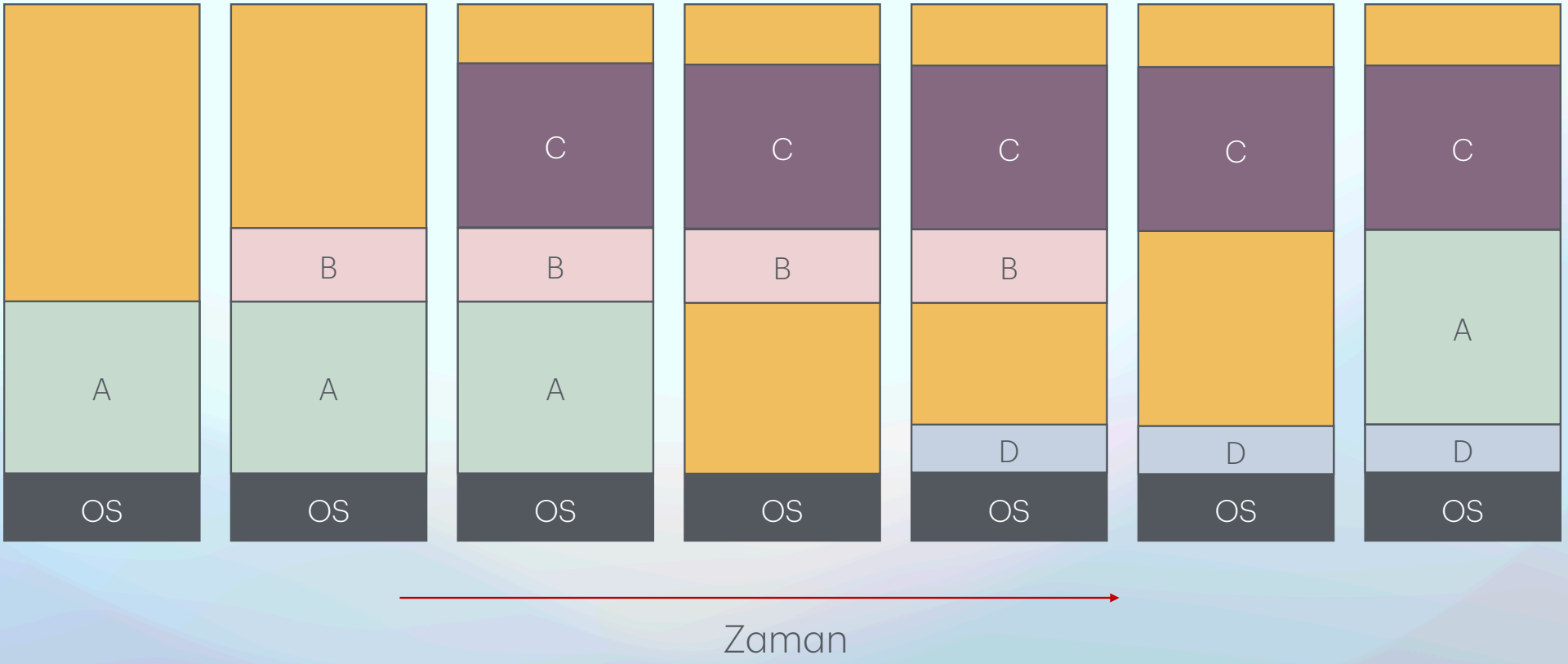
Bellek Yönetimi

Yer değiştirme (Swapping)

- Birden fazla işlem olsa da bellek sınırı bulunmaktadır.
- Belirli işlemlerin kullanılmadığı sürece belleği bırakması gerekir
- Çok büyük bellek ihtiyacı duyan uygulamaların sanal bellek mekanizmasına ihtiyacı bulunur
- İşlemlerin yer değiştirmesi sırasında boşluklar

Bellek Yönetimi

Swapping



Bellek Yönetimi

Compaction

- Bellekteki boşlukların kapatılması gerekmektedir
- Tüm işlemlerin bellekte yanyana getirilmesi
- Memory compaction
- Çok fazla işlem zamanı gerektirmekte

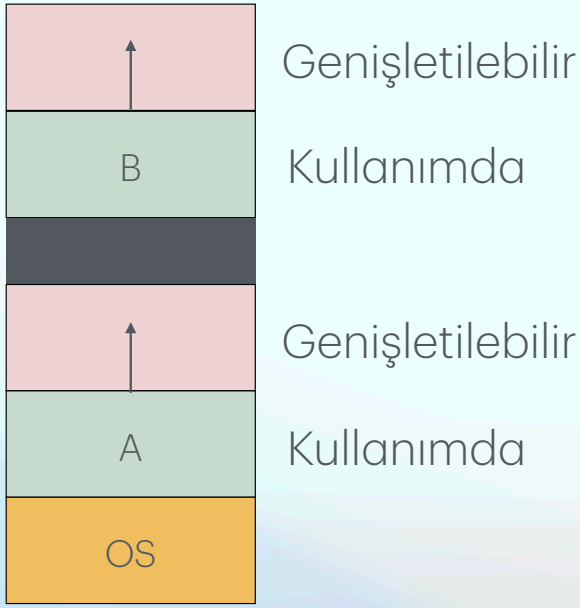
Bellek Yönetimi

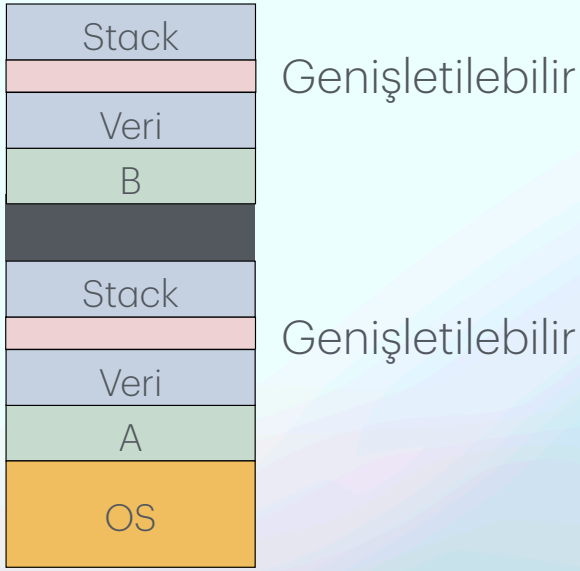
İşlemlere ayrılması gereken bellek miktarı

- Her işleme eşit boyutta bellek verilebilir
- Dinamik bellek kullanan uygulamalar?
  - Her işleme başlangıcından bir miktar fazla genişleme alanı sağlanabilir
- Yine de daha fazla alan ihtiyacı olursa?
  - İşlemi bellekte daha büyük bir yere taşıma
- İşlem belleği yetersiz ve bellekte daha büyük bir yer yoksa?
  - İşlemi durdurma ve belleği diske aktarma (Swap)

Bellek Yönetimi

Data ve stack bölümleri için bellek genişleme stratejileri





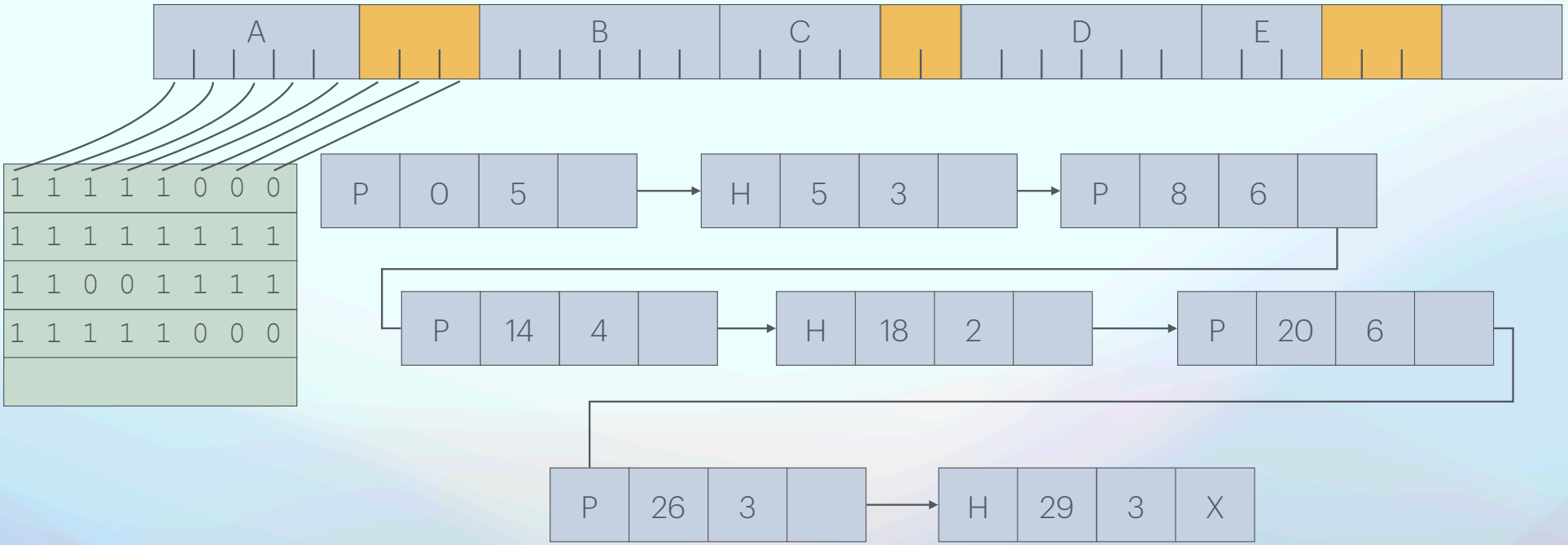
Bellek Yönetimi

Boş belleğin yönetimi

- Bellekte boş veya dolu yerler olabileceği söylenmişti
- Bu yerlere işlemlerin atanması gerekir
- İşletim sisteminin bu boş ve dolu yerleri tutma stratejisi?
  - Bitmap
  - Bağlı liste

Bellek Yönetimi

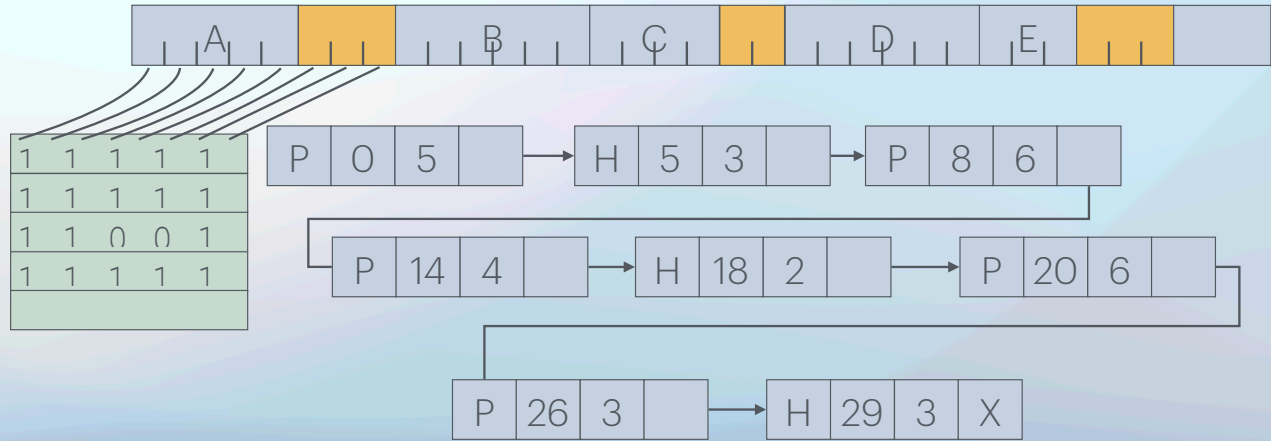
Bitmap ve bağlı listeler



Bellek Yönetimi

Bitmap

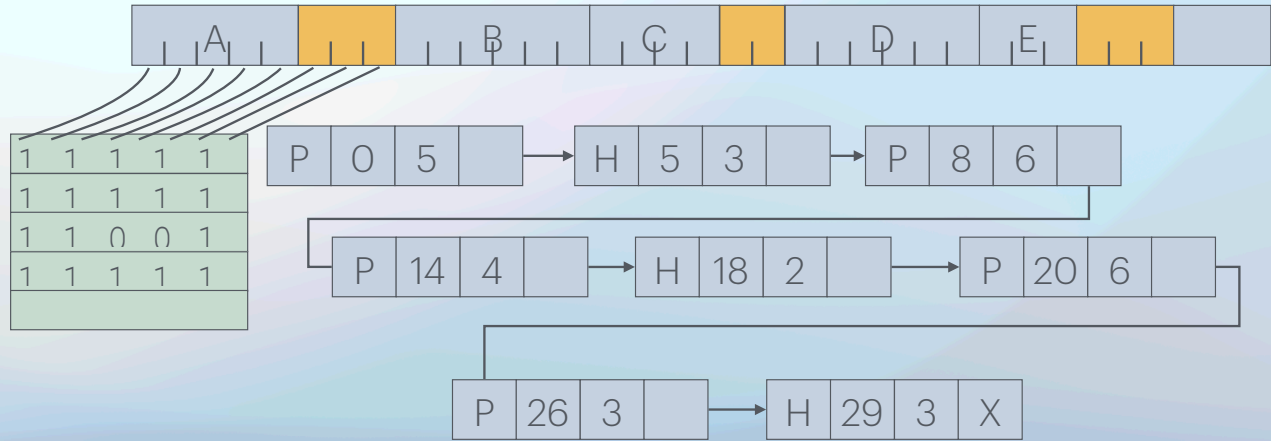
- Bitmap belirlenirken en küçük birimin belirlenmesi kritiktir
- Birim çok küçük tutulursa bitmap büyür
- Birim çok büyük tutulursa gereksiz yer kaybı



Bellek Yönetimi

Bağlı Liste

- Bağlı listeler ile her segment bir düğümde tutulabilir
- Düğümler işlemi veya boşluğu içerebilir
- Güncelleme işlemi bitmap'e göre daha az maliyetli
- Yerleşim için bağlı liste üzerinde gezme işlemi gerekli



Bellek Yönetimi

Boş bellek arama yöntemleri

- First fit
  - İlk boş bulunan yere işlem yerleştirilir
- Next fit
  - First fit gibi fakat liste baştan taranmaz, kaldığı yerden devam eder
- Best fit
  - En maliyetli yöntem - Tüm listenin gezilmesi gerekir
  - Parçalanma (fragmentation) minimum olur
- Quick fit
  - Hibrit bir algoritma
  - Farklı boyutlar için farklı bağlı listeler tutulur

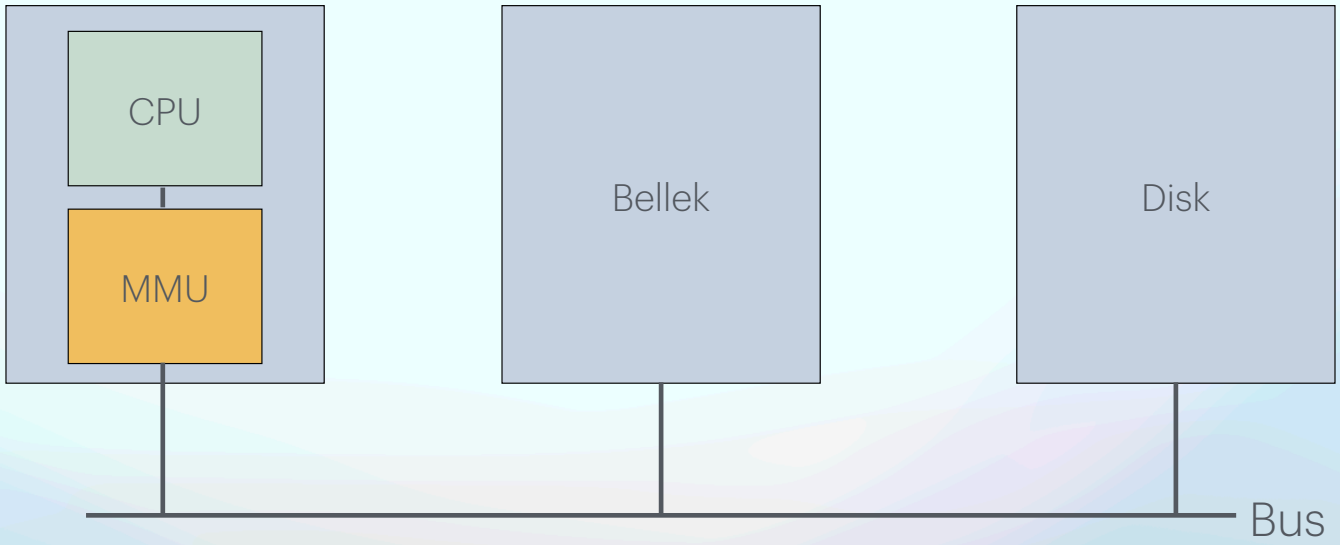
Bellek Yönetimi

Sanal bellek

- Belleğin bir uygulama için yetersiz olma durumu
- Programın çalışacağı bloklar bölünüp sırayla çalıştırılabilir
  - Yazılım geliştirenlerin işi
- Sanal bellek
  - Belleğin sayfalara (page) bölünmesi
- Bellek yönetim birimi (Memory management unit)
  - Elde olan bellekten daha fazla bit ile adresleme

Bellek Yönetimi

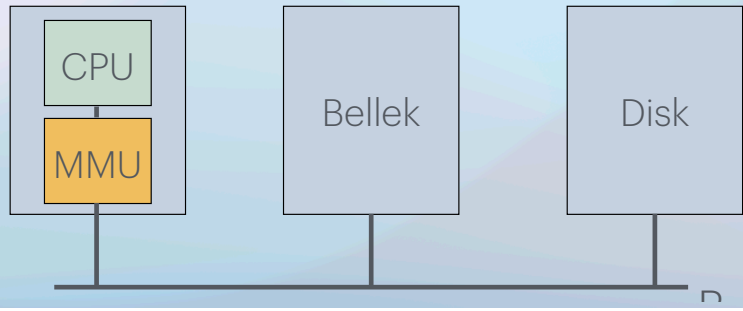
MMU



Bellek Yönetimi

MMU

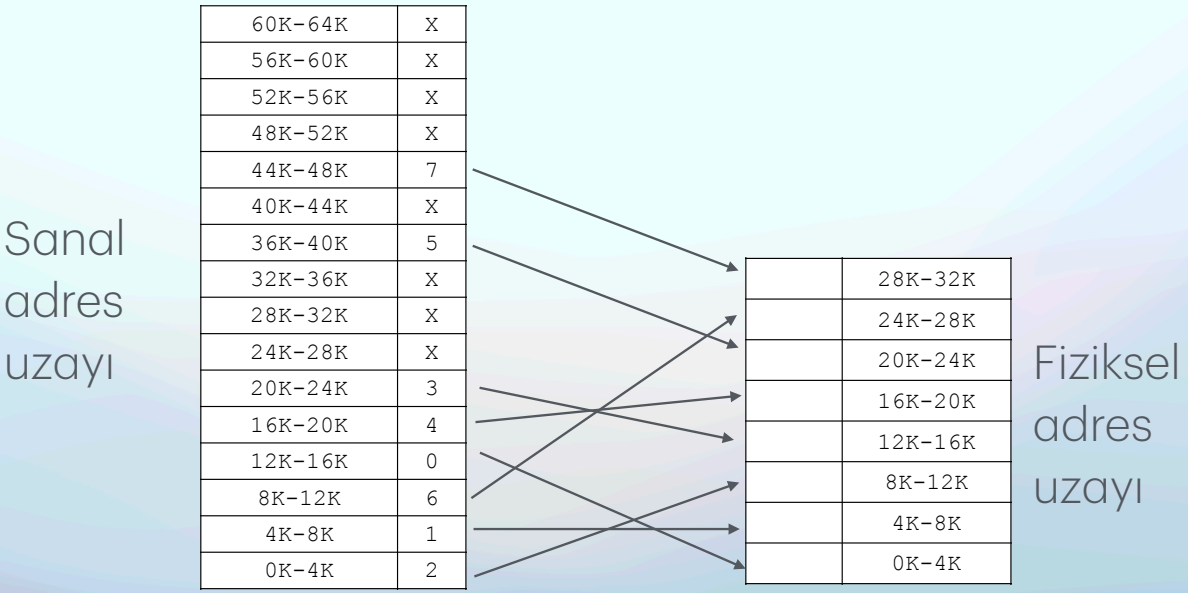
- Program MMU'ya bellek erişimi için başvurur
- MMU sayfaların fiziksel bellekte nereye karşılık geldiğini kontrol eder
- Fiziksel bellekte bulunan sayfalar geri döndürülür
- Eğer fiziksel bellekte bulunmuyorsa?
  - Var-yok (Present-absent) biti
  - Page Fault



Bellek Yönetimi

Page fault

- İşletim sistemi kullanılmayan bir sayfayı diske yazar
- Diskten ihtiyaç duyulan sayfayı okur



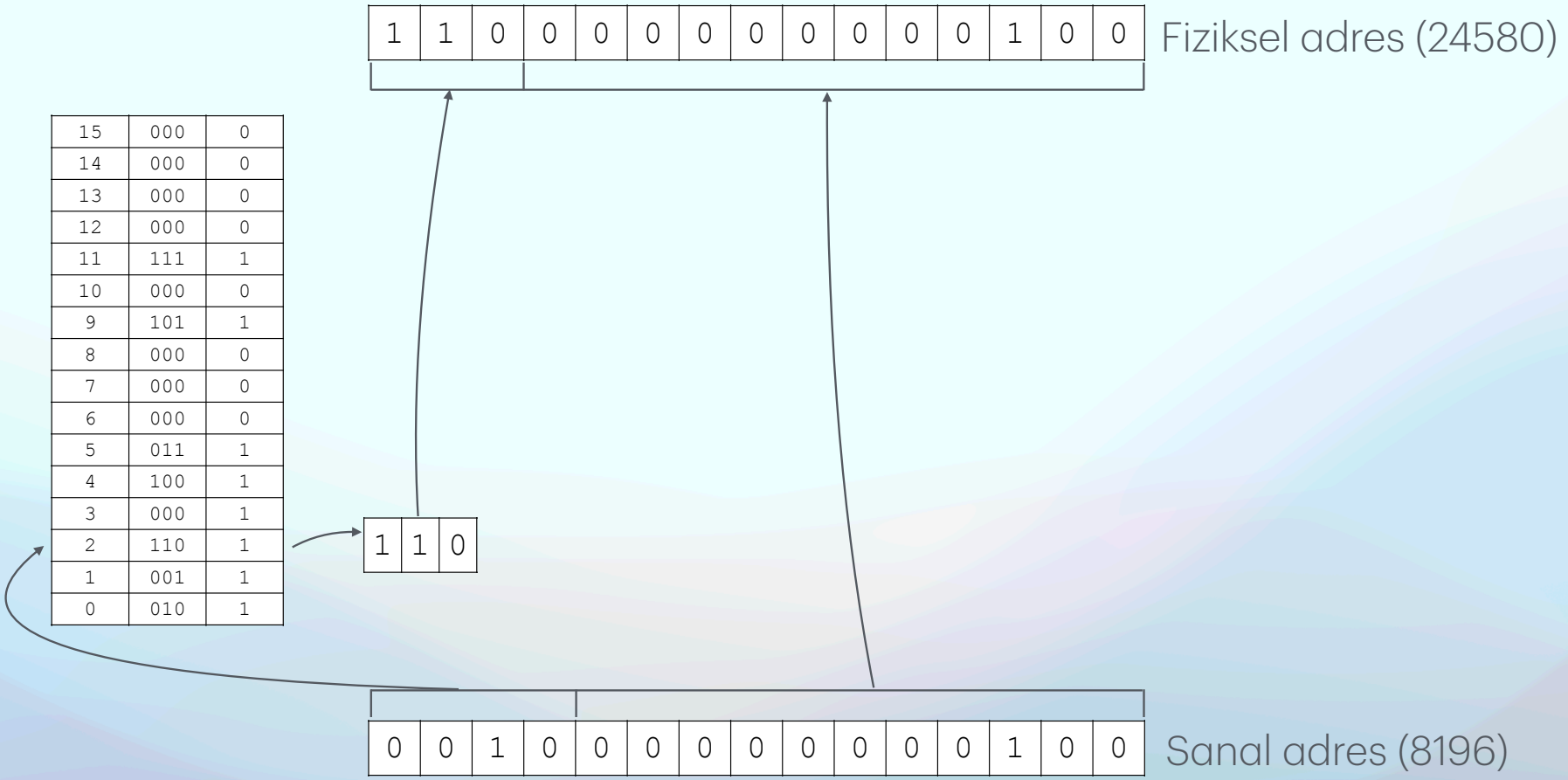
Bellek Yönetimi

Sayfa tabloları

- Sanal adresler gerçek adreslerden daha uzun olmalı
- Gerçek adrese dönüştürülen bir fonksiyon veya bir tabloya ihtiyaç duyulur
- Tablonun aynı zamanda değerin bellekte olup olmadığına dair bir bit tutması gerekir

Bellek Yönetimi

Sayfa tabloları



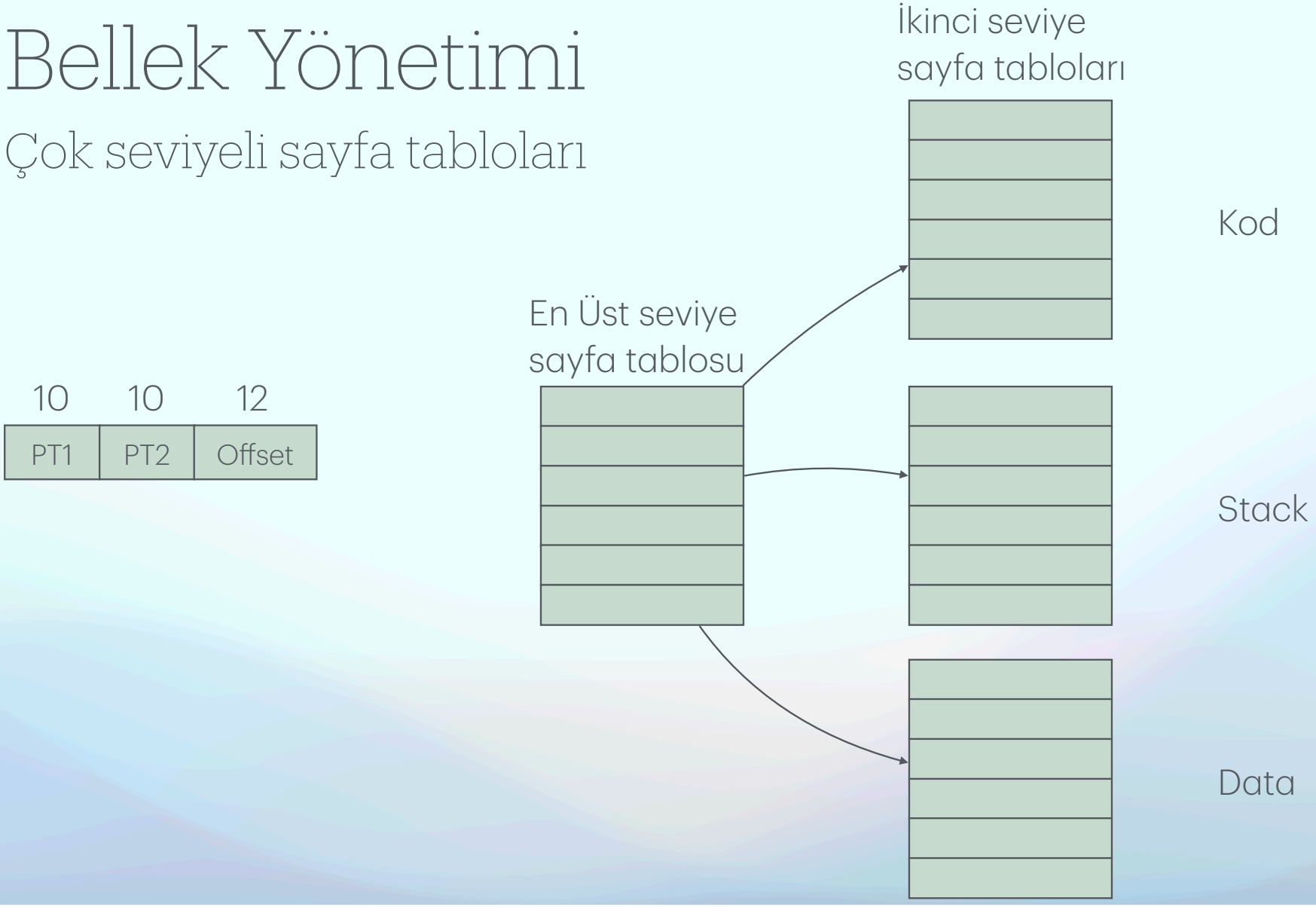
Bellek Yönetimi

Sayfa tabloları

- Bir sayfa tablosu kaydında başka özellikler de bulunabilir
  - Değiştirilme bayrağı
  - Koruma bayrağı
    - Okuma / Yazma koruması
  - Supervisor bayrağı
    - Sadece işletim sistemi mi erişebilir yoksa kullanıcı programları da erişebilir mi?
  - Referans bayrağı
    - Silinme durumunda karar vermek için

Bellek Yönetimi

Çok seviyeli sayfa tabloları



Bellek Yönetimi

Sayfa değiştirme algoritmaları

- Optimal değiştirme
- Sıklıkla kullanılmayan sayfaların değiştirilmesi
- İlk giren ilk çıkar
- İkinci şans
- Saat yönü değiştirme
- En az kullanılan sayfanın değiştirilmesi

Bellek Yönetimi

Optimal değiştirme algoritması

- Hangi sayfanın ne zaman isteneceği belli değil
- Bunun için bir simulator çalıştırılabilir
  - İki aşama performanslı değil
- Farklı durumlarda farklı algoritmalar optimum olabilir

Bellek Yönetimi

Sıklıkla kullanılmayan sayfaların değiştirilmesi

- Bellekte belirli bayraklar tutuluyordu
  - Okundu ve değiştirildi (R ve M) bayrakları
- İşlem başladığında değerler 0 yapılır
- Erişilmeyen alanların R biti belirli periyotlar ile sıfırlanır
- Page fault olduğunda kontrol edilir

Bellek Yönetimi

Sıklıkla kullanılmayan sayfaların değiştirilmesi

- Kontrol sırasında 4 farklı ihtimal bulunabilir
  - Sınıf 1: R=0, M=0
  - Sınıf 2: R=0, M=1
  - Sınıf 3: R=1, M=0
  - Sınıf 4: R=1, M=1
- İkinci durum imkansız gözükse de son durumdaki bir sayfanın bir sonraki çevrimde R bitinin temizlenmesi ile elde edilebilir

Bellek Yönetimi

Sıklıkla kullanılmayan sayfaların değiştirilmesi

- Algoritmada amaç en düşük sınıfta bulunan bir sayfanın silinmesi
  - Sınıf 1: R=0, M=0
  - Sınıf 2: R=0, M=1
  - Sınıf 3: R=1, M=0
  - Sınıf 4: R=1, M=1
- Temel amaç erişilmeyen sayfanın silinmesi

Bellek Yönetimi

FIFO

- İlk giren ilk çıkar
- Bellekte page fault olduğu an sıradaki çıkartılır ve yeni okunan sayfa sona eklenir
- Çok okunan ama çıkartılan sayfa olma ihtimali çok yüksek

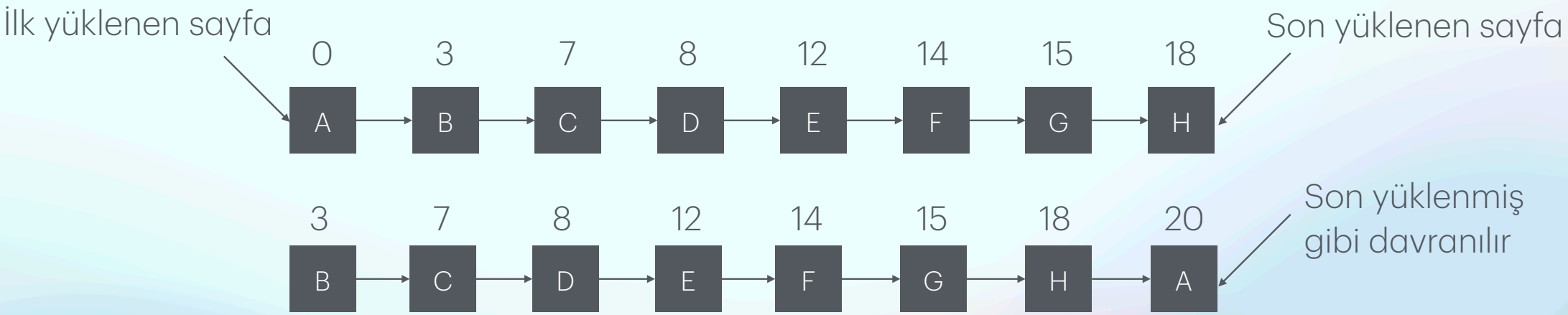
Bellek Yönetimi

İkinci şans

- FIFO algoritmasının modifikasyonu
- Page fault oluştuğunda ilk sayfanın R bitinin kontrolü
- R=0'sa sayfa hem kullanılmıyor hem de eski
  - Silinir
- R=1'se 0 yapılır ve listenin sonuna atılır ve yeni listede silinecek bir düğüm aranmaya devam edilir

Bellek Yönetimi

İkinci şans



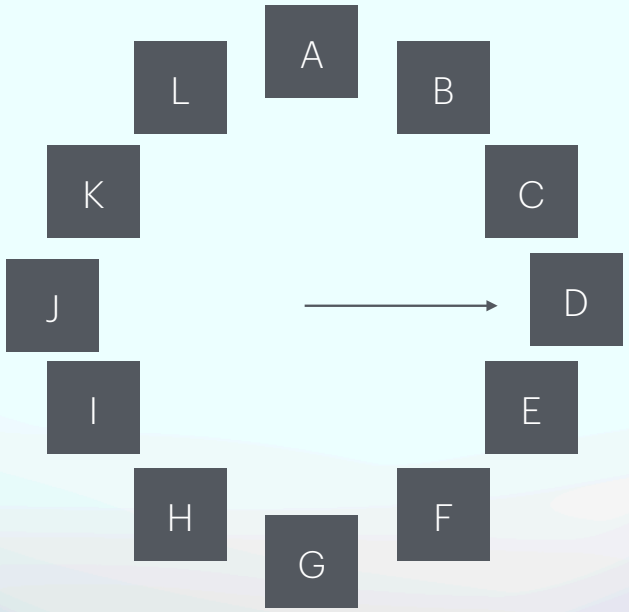
Bellek Yönetimi

Saat yönü değiştirme

- İkinci şans algoritması fazla bağlı liste işlemi yapar
- Aynı işlem döngüsel bir bağlı liste ile gerçekleştirilebilir
- Yine R biti kontrol edilir ve liste sonuna gitmek yerine işaretçi ilerletilir

Bellek Yönetimi

Saat yönü değiştirme



Page Fault durumunda

R=0: Sayfayı çıkar

R=1: R'yi temizle kolu ilerlet

Bellek Yönetimi

En az kullanılan sayfanın değiştirilmesi (LRU)

- Her sayfanın ne zaman erişildiğinin tutulması gerekir
  - Bağlı listede tutulabilir
- Listenin her döngüde güncellenmesi gerekir
- Sayfa arama, silme, taşıma işlemlerinin maliyeti bulunur
- Donanımsal bir şekilde hızlandırma ihtimali bulunmaktadır

Bellek Yönetimi

Yük kontrolü

- Birden fazla işlemin çalıştığı durumda bazı işlemleri belleğe alabilmek için belleğin boşaltılması zorunlu olabilir
- Bu durumda bellekten bazı işlemlerin atılması gerekir
- İşletim sistemi bir işlem kapatma işlemi (OOM killer) içerir
- Bu işlem tüm çalışan işlemleri kontrol eder ve belirli kurallar dahilinde işlemleri kapatmayı seçebilir.

Bellek Yönetimi

Yük kontrolü

- Çok fazla bellek kullanan işlemler kapatma için yüksek seviyeli aday olabilir
- İşletim sistemi işlemleri düşük seviyeli kabul edilir
- Aynı zamanda olabilecek en az sayıda işlemin kapatılması amaçlanır
- İşlemler kapatılmadan swap işlemi de denenebilir
- Bu işlemlerin dışında bellek sıkıştırılabilir
  - Memory compression

Bellek Yönetimi

Sayfa paylaşımı

- Ortak kullanılan sayfalar paylaşılabilir
- Sayfaların birden fazla kopyasını engeller
- Yalnızca okuma izni verilir
- Yazma işlemi yapıldığında sayfanın bir kopyası alınır
  - Copy on write

Bellek Yönetimi

Paylaşılan kütüphaneler

- Sayfaların paylaşıldığı gibi belirli ortak kütüphaneler de paylaşılabilir
- Birden fazla uygulama bir kütüphane içerisindeki metotları kullanabilir
- Her işlemin bu kütüphanenin bir kopyasını belleğe çıkarmasına gerek yoktur
- DLL
- Problem MMU biriminin farklı sanal bellek adreslerini bu kütüphaneye vermesidir
- Bunun için derleme esnasında statik adres verilmemesi ve relative adresleme yapılmasıdır