Ana Bellek
(Main Memory)

Dr. Günay TEMÜR

Bölüm 8 : Ana Bellek

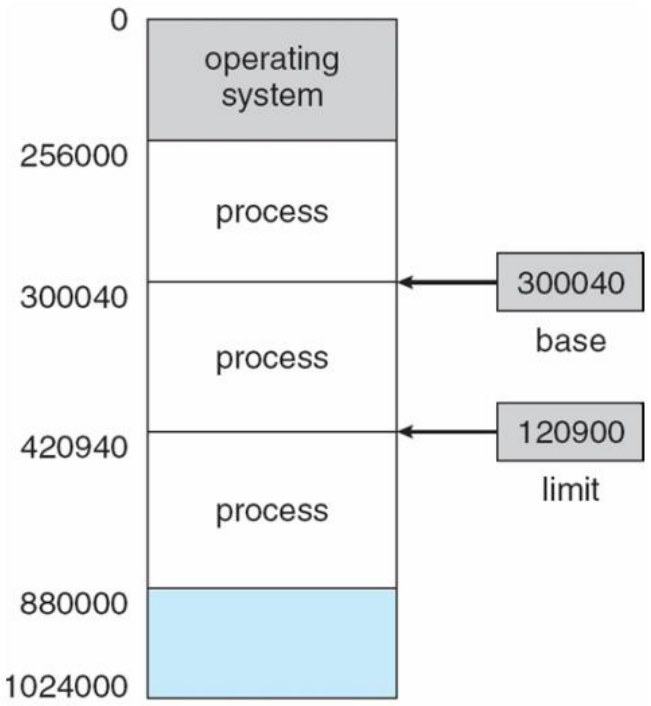
- Background
- Yer Değiştirme
- Bitişik Bellek Yerleşimi (Contiguous Memory Allocation)
- Sayfalama (Paging)
- Sayfa Tablosunun Yapısı (Page Table)
- Bölütleme (Segmentation)
- Örnek: Intel Pentium

Background

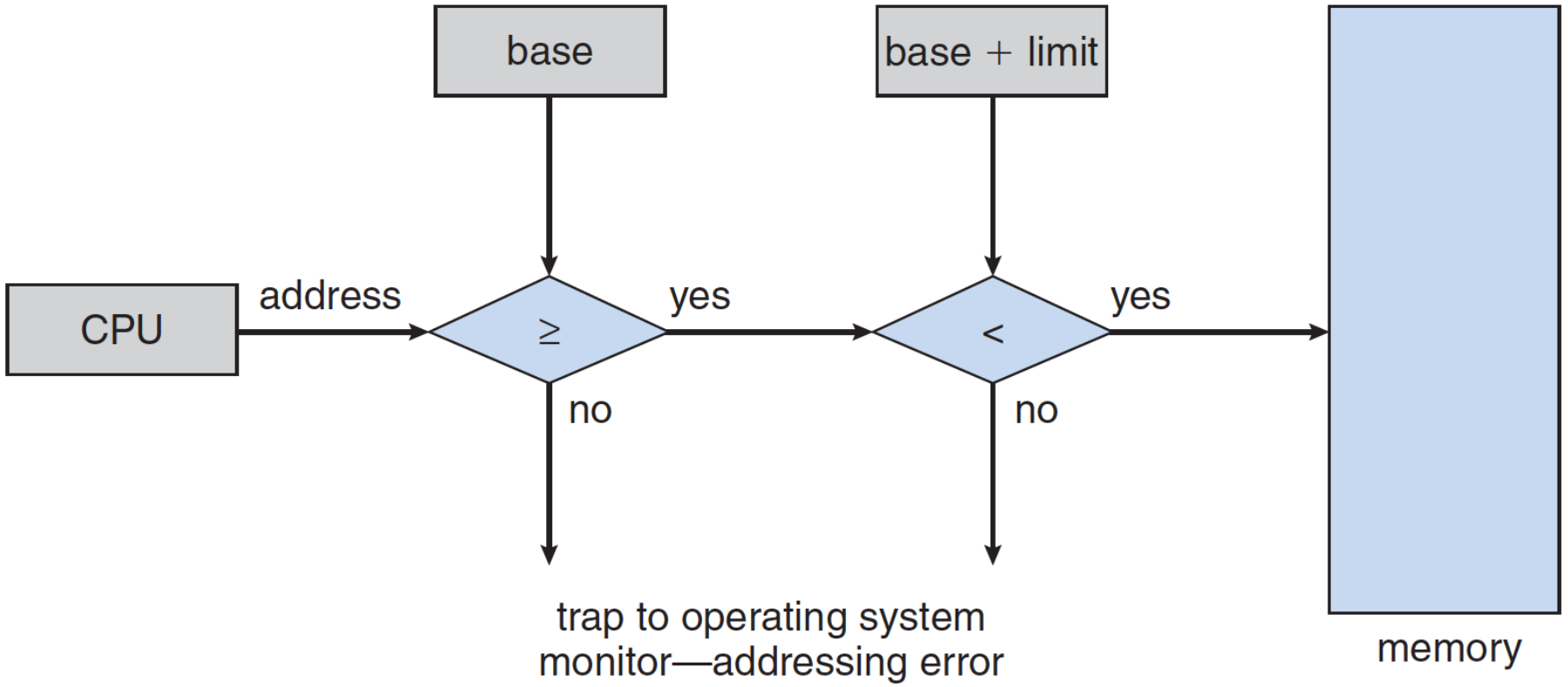
- Çalıştırılmak istenen program öncelikle diskten belleğe alınmalı ve bir işleme dönüştürülmelidir
- CPU'nun direk olarak erişebileceği kayıt birimleri yalnızca ana bellek (main memory) ve yazmaçlardır (registers)
- CPU'nun yazmaçlara erişimi bir CPU birim zamanı (veya daha az) sürer
- Ana belleğe erişim pek çok CPU birim zamanı sürebilir
- Ön bellek (cache) ana bellek ile CPU yazmaçları arasında yer almaktadır
- Ana belleğin korunması sistemin doğru çalışması için şarttır

Taban ve Sınır Yazmaçları

- Taban (base) ve sınır (limit) yazmaçları mantıksal adres uzayını tanımlar



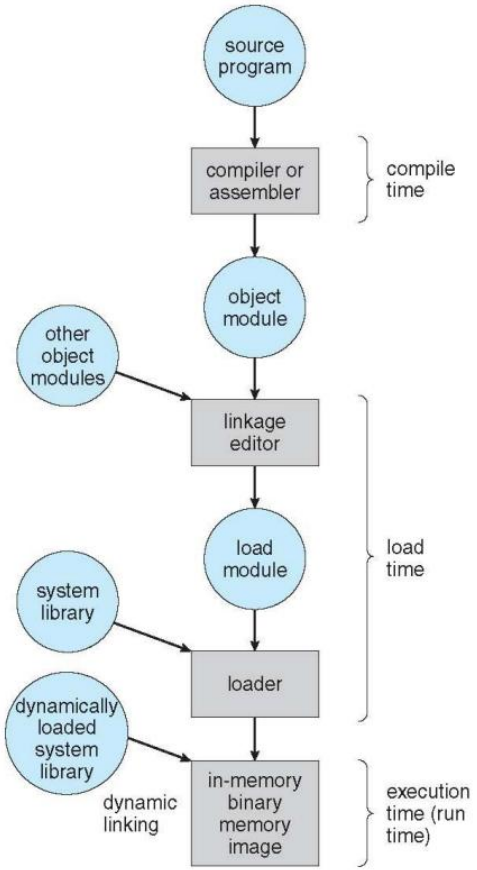
Donanım Adres Koruması



Komutların ve Verilerin Hafızaya Bağlanması

- Programlardaki komutların ve verilerin adreslerinin hafızadaki adreslere bağlanması üç farklı aşamada gerçekleşebilir
  - **Derleme zamanı (compile time):** Eğer ana bellekteki pozisyon önceden biliniyorsa, **mutlak kod (absolute code)** oluşturulabilir. Başlangıç adresi değişirse kodun yeniden derlenmesi gerekir
  - **Yükleme zamanı (load time):** Eğer derleme zamanında ana bellek pozisyonu bilinmiyorsa, **yeniden yerleştirilebilir kod (relocatable code)** oluşturulmalıdır
  - **Çalışma zamanı (execution time):** Eğer işlem çalışırken bir hafıza bölümünden diğerine taşınabiliyorsa, bağlama çalışma zamanına kadar ertelenir
    - Adres haritaları (address maps) için donanım desteği gerekir (örn., taban ve sınır yazmaçları)

Kullanıcı Programının Çok Adımlı İşletimi



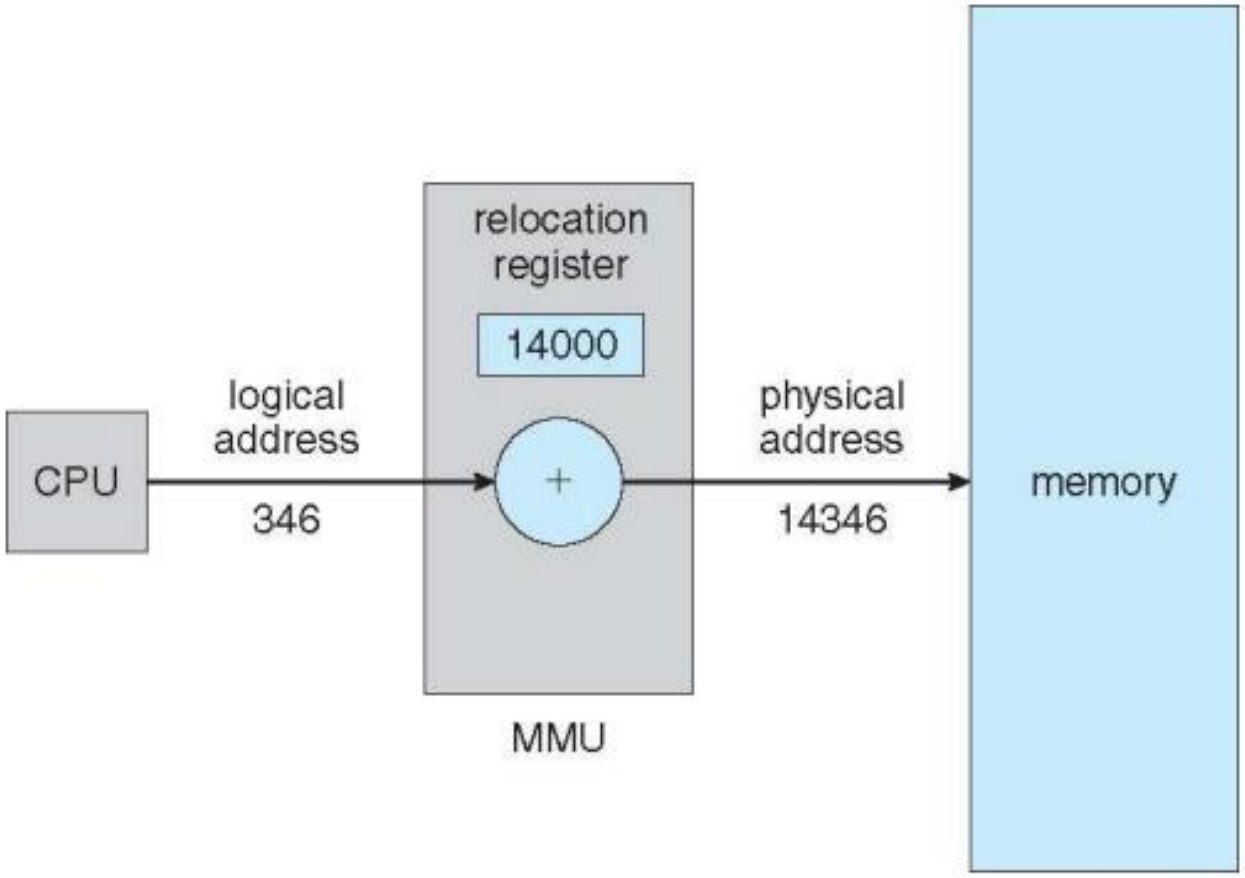
Mantıksal veya Fiziksel Adres Uzayı

- Mantıksal bir adres uzayının ayrı bir fiziksel adres uzayına bağlanması düzgün bir hafıza yönetimi için ön şarttır
  - **Mantıksal adres (logical address)** –CPU tarafından oluşturulur.
    Sanal adres (virtual address) olarak da adlandırılır
  - **Fiziksel adres (physical address)** – hafıza birimi tarafından bilinen adrestir
- Derleme-zamanı ve Yükleme-zamanı adres bağlama yaklaşımlarında mantıksal ve fiziksel adresler aynıdır
- Çalışma zamanı adres bağlama yaklaşımında mantıksal ve fiziksel adresler farklılık gösterir

Hafıza Yönetim Birimi (MMU)

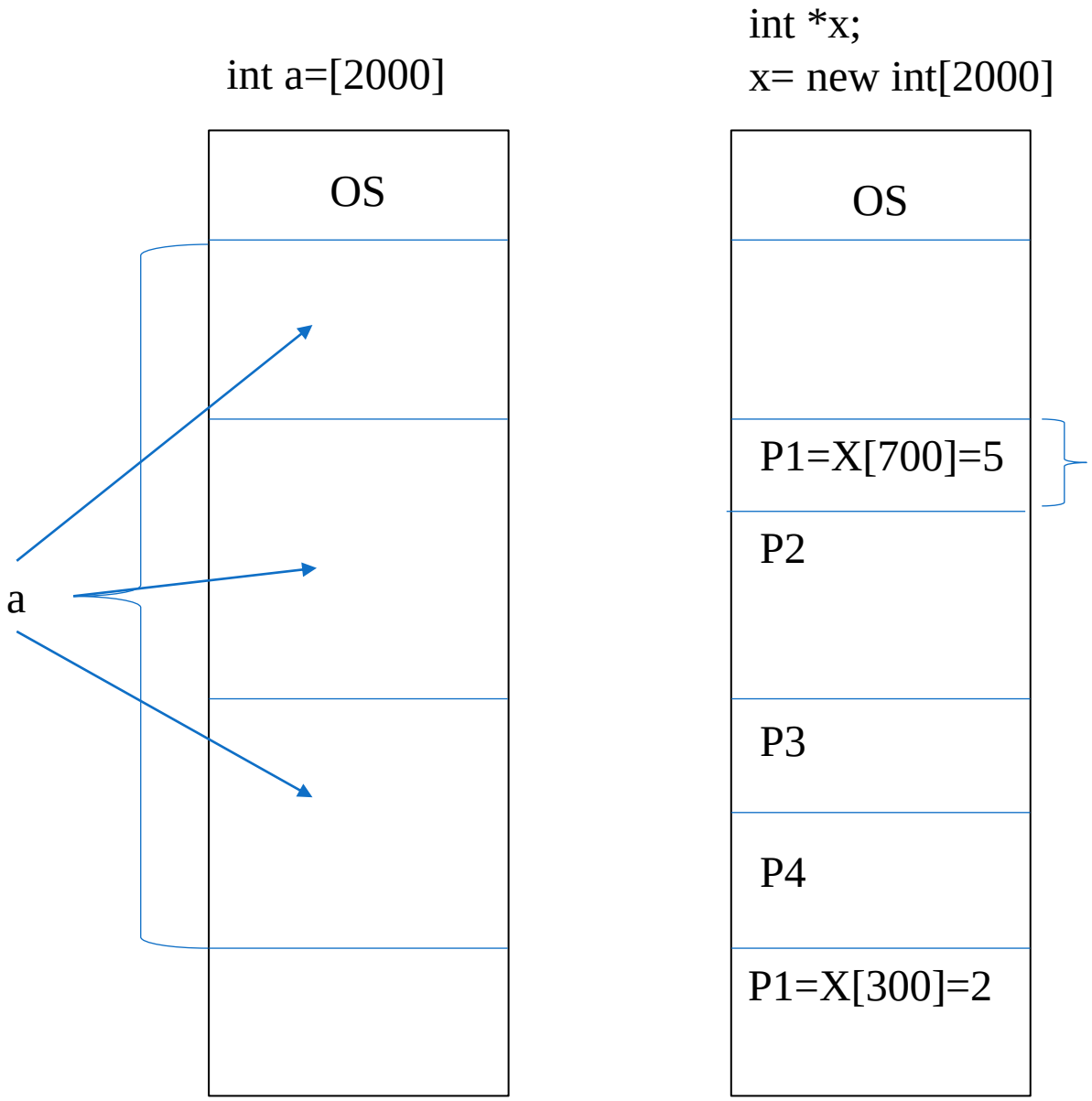
- Sanal adresleri fiziksel adreslere çeviren donanım
- MMU birimine kullanıcı programı tarafından gönderilen her adrese **yeniden yerleştirme yazmacındaki** (relocation register) değer eklenir
- Elde edilen adres hafızaya gönderilir
- Kullanıcı programları mantıksal adresleri kullanırlar; gerçek fiziksel adresleri asla görmezler

Yeniden Yerleştirme Yazmacını Kullanarak Dinamik Yeniden Yerleştirme



Dinamik Yükleme (Dynamic Loading)

- Metot çağırılmadığı sürece yüklenmez
- Daha iyi hafıza uzayı yönetimi; kullanılmayan metot asla yüklenmez
- Nadiren gerçekleşen durumlara karşılık gelen büyük miktarda kod olduğunda faydalı
- İşletim sisteminden özel destek gerektirmiyor – program tasarımına dikkat edilmesi yeterli


int a=[2000]
OS
a
int *x;
x= new int[2000]
OS
P1=X[700]=5
P2
P3
P4
P1=X[300]=2

Dinamik Bağlama (Dynamic Linking)

- Bağlama çalışma zamanına kadar ertelenir
- Dinamik bağlama özellikle kütüphaneler için faydalıdır
- **Stub**: hafızadaki kütüphane metodunun yerini bulmakta kullanılan küçük kod parçası
- Stub kendini kütüphane metodunun adresi ile değiştirir ve metodu çalıştırır
- İşletim sistemi, kütüphane metodunun, işlemin bellek adresi içinde olduğunu kontrol etmelidir
- Bu mekanizma aynı zamanda **paylaşımlı kütüphane (shared libraries)** olarak bilinmektedir

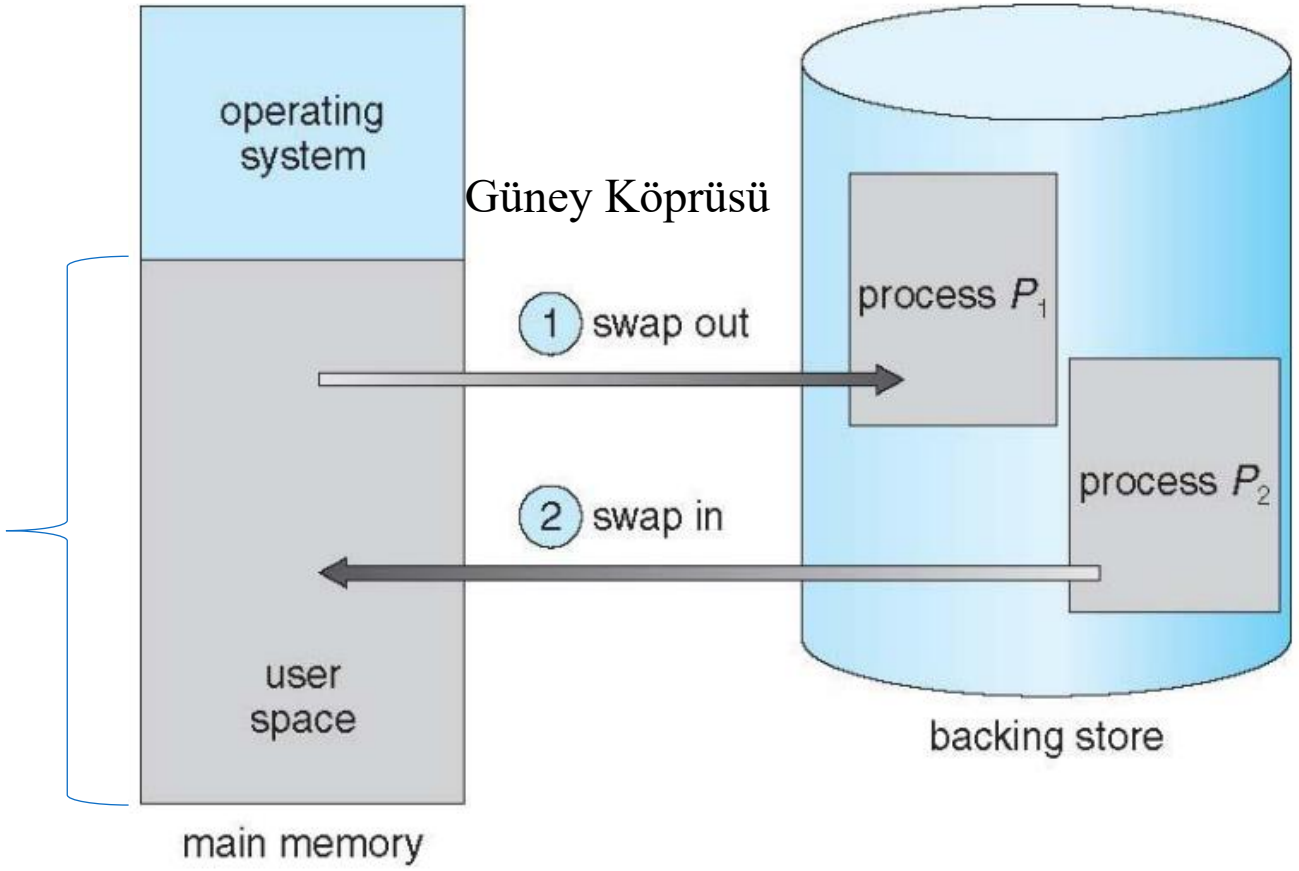
Yer Değiştirme (Swapping) (1/2)

- Bir işlem geçici olarak hafızadan çıkarılıp ikincil bir kayıt birimine alınabilir
- İşlem daha sonra çalışmaya devam etmek üzere yeniden ana belleğe alınabilir
- **Roll out, roll in** – öncelik tabanlı zamanlama algoritmaları tarafından kullanılan yer değiştirme varyasyonu
  - Düşük öncelikli işlemler, daha yüksek öncelikli işlemlerin yüklenip çalıştırılabilmesi için geçici olarak hafıza dışına alınır

Yer Değiştirme (Swapping) (2/2)

- Yer değiştirme zamanının önemli bir kısmı verinin transferinde kullanılır. Toplam transfer zamanı yer değiştirilen hafıza boyutu ile orantılıdır
- Yer değiştirmenin farklı varyasyonları pek çok işletim sisteminde bulunur (örn: UNIX, Linux ve Windows)
- Sistem çalışmaya hazır olan ancak kodu hafızada olmayan işlemleri hazır kuyruğunda (ready queue) tutar ve bu kuyruğu günceller

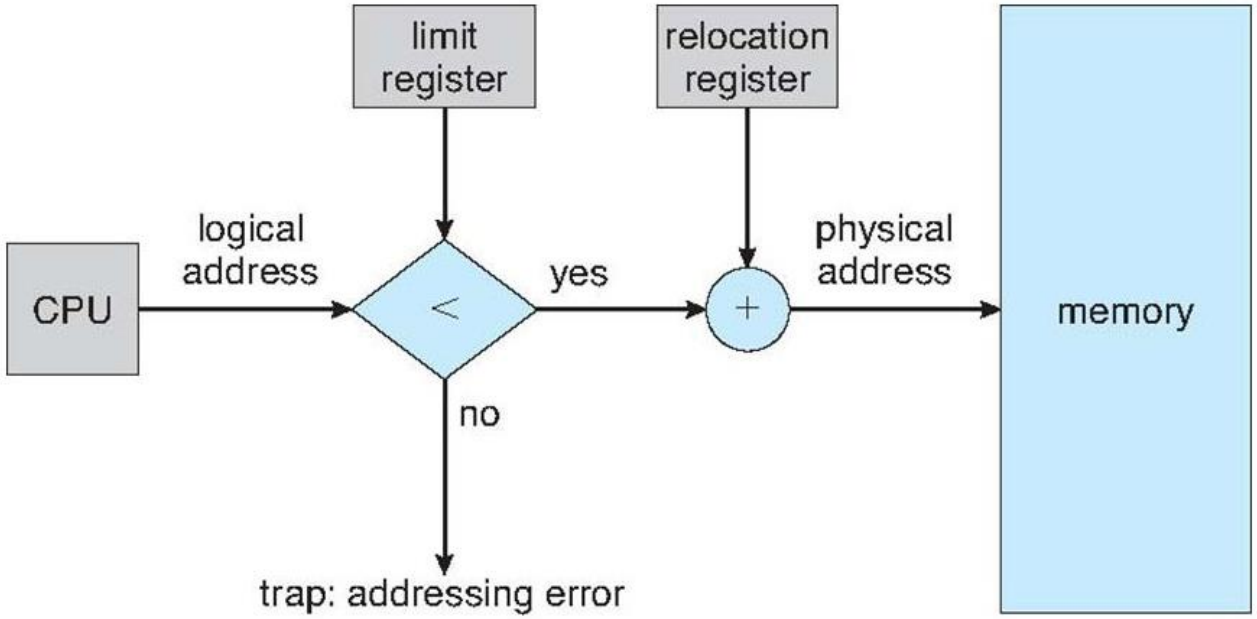
Yer Değiştirme



Yeniden Yerleştirme ve Sınır Yazmaçları için Donanım Desteği

- Contiguous Memory Allocation
- Ana bellek genellikle iki bölüme ayrılır:
  - İşletim sistemi, genellikle kesinti vektörü ile birlikte hafızanın alt kısmında yer alır
  - Kullanıcı işlemleri, hafızanın üst kısmında yer alır
- Yeniden yerleştirme yazmaçları kullanıcı işlemlerini birbirinden korumak, işletim sistemi kodu ve verilerinin değiştirilmesini önlemek için kullanılır
  - Taban yazmacı, en küçük fiziksel adresin değerini tutar
  - Sınır yazmacı, mantıksal adreslerin sınır değerini tutar – tüm mantıksal adresler sınır yazmacında tutulan değerden daha küçüktür
  - MMU mantıksal adresleri dinamik olarak çevirir

Yeniden Yerleştirme ve Sınır Yazmaçları için Donanım Desteği

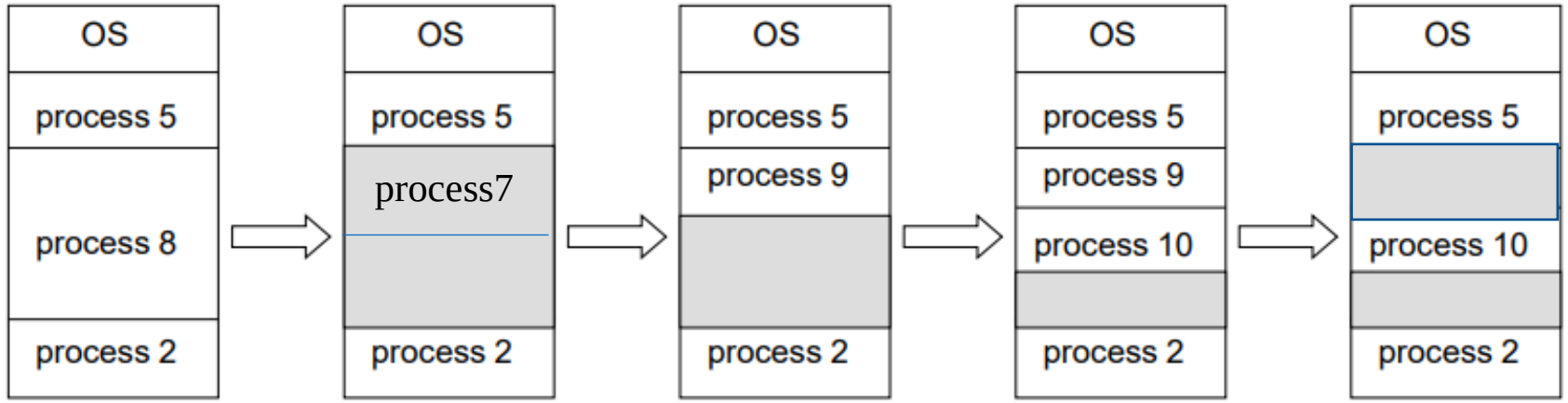


Bitişik Bellek Yerleşimi (Devam)

- Çoklu-bölüm ayırımı
  - Boşluk (Hole) – kullanılabilir bellek bloğu; hafıza üzerinde çeşitli boyutlarda boşluklar dağınık bir şekilde bulunur
  - Bir işlem geldiğinde, bu işlemi tutabilecek kadar büyük bir hafıza boşluğuna yerleştirilir
  - İşletim sistemi şu bilgileri tutup günceller:

    a) işlemlere ayrılmış bölümler b) boş bölümler (boşluklar)

Sonraki slayt için örnek



Dinamik Kayıt Birimi Ayırım Problemi

Bir boş bölümler listesi elimizdeyken, *n* boyutunda bir işlemi nasıl bir boşluğa atayabiliriz?

- **İlk-uyum (first-fit):** yeterince büyük olan ilk boşluğa ata
- **En-iyi-uyum (best-fit):** yeterince büyük olan en küçük boşluğa ata
  - boyuta göre sıralı değilse, tüm liste aranmalıdır
  - geriye en küçük boşluğu bırakır
- **En-kötü-uyum (worst-fit):** mevcut en büyük boşluğa ata
  - gene, tüm liste aranmalıdır
  - geriye en büyük boşluğu bırakır

İlk-uyum ve en-iyi-uyum, hız ve kayıt biriminin verimli kullanımı açısından en-kötü-uyum'a göre daha iyi sonuç verir

Parçalanma (Fragmentation) (1/2)

- **Dışsal Parçalanma (External Fragmentation)** – isteği karşılamak için hafıza alanı mevcut fakat bitişik değil
- **İçsel Parçalanma (Internal Fragmentation)** – işleme ayrılan hafıza gerekenden biraz fazla
  - boyut farkı işleme ayrılan hafıza alanında oluşuyor ve bu alan kullanılamıyor

Parçalanma (Fragmentation) (2/2)

- Dışal parçalanmayı **sıkıştırma (compaction)** ile azalt
  - Hafıza bloklarını, tüm boş blokları bir araya getirecek şekilde yeniden düzenle
  - Sıkıştırma sadece, yeniden yerleştirme dinamik ise, çalışma zamanında gerçekleştirilir
  - I/O problemi
    - İşlemler I/O gerçekleştirirken hafızadaki yerlerini sabitle
    - I/O işlemlerini sadece işletim sistemine ait tampon bellekler üzerinde gerçekleştir

Sayfalama (Paging) (1/2)

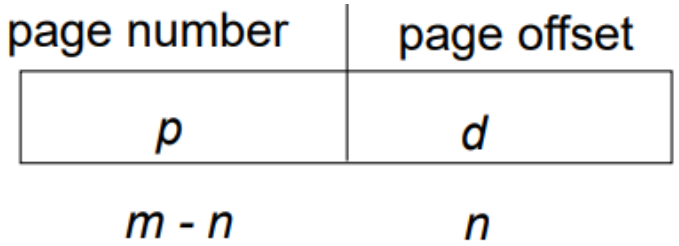
- Sayfalama, bir işleme ayrılan fiziksel adres uzayının bitişik olmamasına (noncontiguous) izin veren bir hafıza yönetim şeklidir
- Fiziksel hafıza sabit büyüklükte bloklara, **çerçevelere (frames)**, bölünür (belirlenen boyut 2'nin katları halindedir, 512 bayt ile 8,192 bayt arası)
- Mantıksal hafıza alanı da aynı boyutta bloklara, **sayfalara (pages)**, bölünür
- Tüm boş çerçeveler takip edilir: **çerçeve tablosu (frame table)**
- *n* sayfalık bir programı çalıştırmak için, n tane boş çerçevenin bulunması ve bu çerçevelere programın yüklenmesi gerekir
- Mantıksal adreslerin fiziksel adreslere dönüştürülmesi için bir **sayfa tablosuna (page table)** ihtiyaç duyulur

Sayfalama (Paging) (2/2)

- Dışsal parçalanma yok fakat içsel parçalanma var
- İçsel parçalanmayı azaltmak için, sayfa boyutu küçültülmelidir
- Sayfa tablosuna her erişim sistem kaynaklarını kullanmayı gerektirdiğinden, sayfa tablosuna erişim sayısını azaltmak için sayfa boyutu olabildiğince büyük olmalıdır
- Ayrıca, disk I/O işlemleri genellikle daha büyük miktarlarda veri transfer edildiğinde daha verimli çalışır
- Ödünleşme (tradeoff)
- İşlem boyutları arttıkça, sayfa boyutları da zamanla artmıştır
- Günümüzde tipik olaraksayfa boyutları 4KB ile 8KB arasındadır
- Bazı sistemler çok daha büyük sayfa boyutlarını desteklerler
- Solaris: 8KB ve 4MB

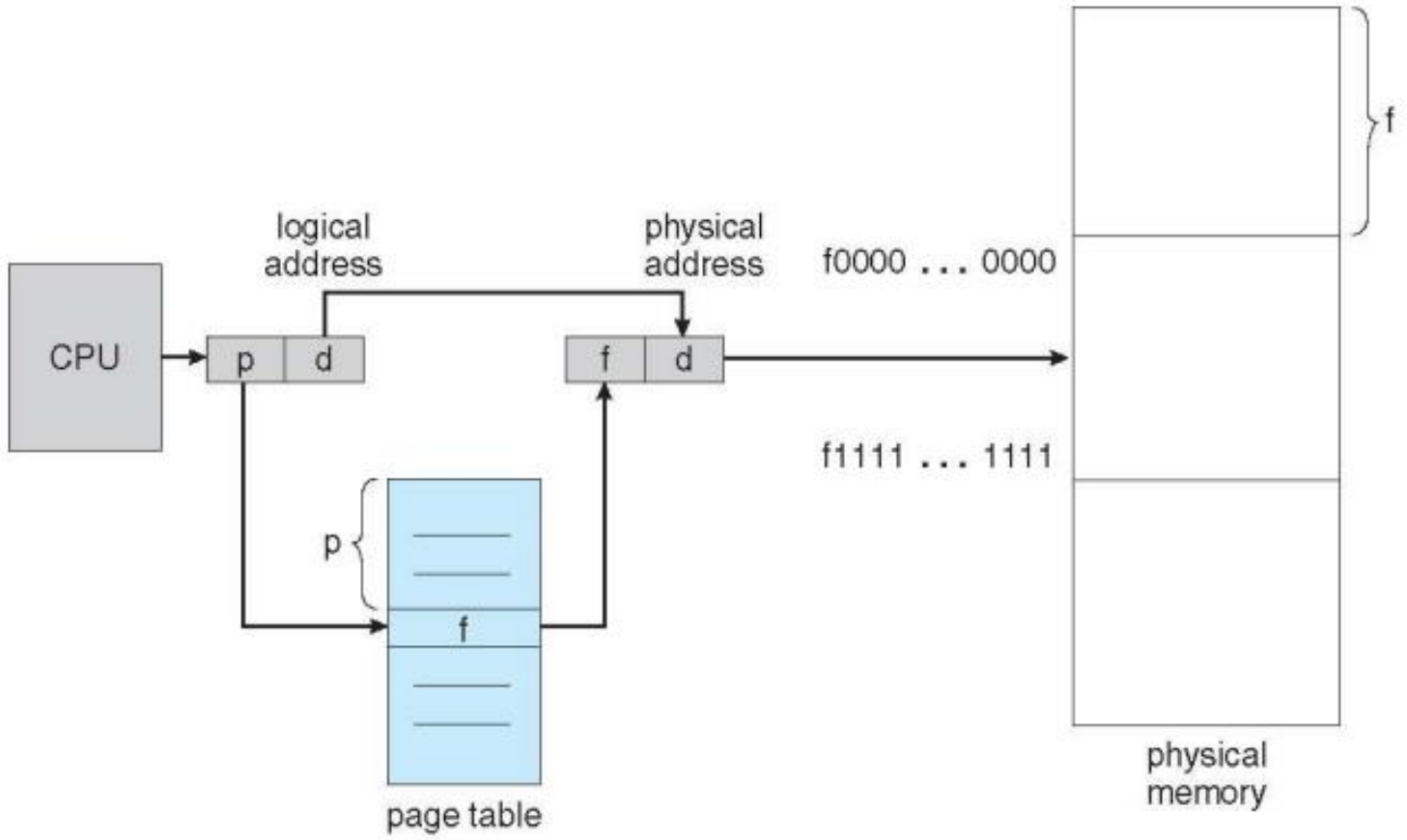
Adres Dönüşüm Mekanizması

- CPU tarafından üretilen adresle ikiye bölünür:
  - **Sayfa numarası (page number) (p)** – fiziksel hafızadaki her bir sayfanın temel adresini içeren sayfa tablosunun indeksi olarak kullanılır
  - **Sayfa ofseti (page offset) (d)** – temel adres ile birleştirilerek hafıza birimine gönderilecek fiziksel hafıza adresi elde edilir

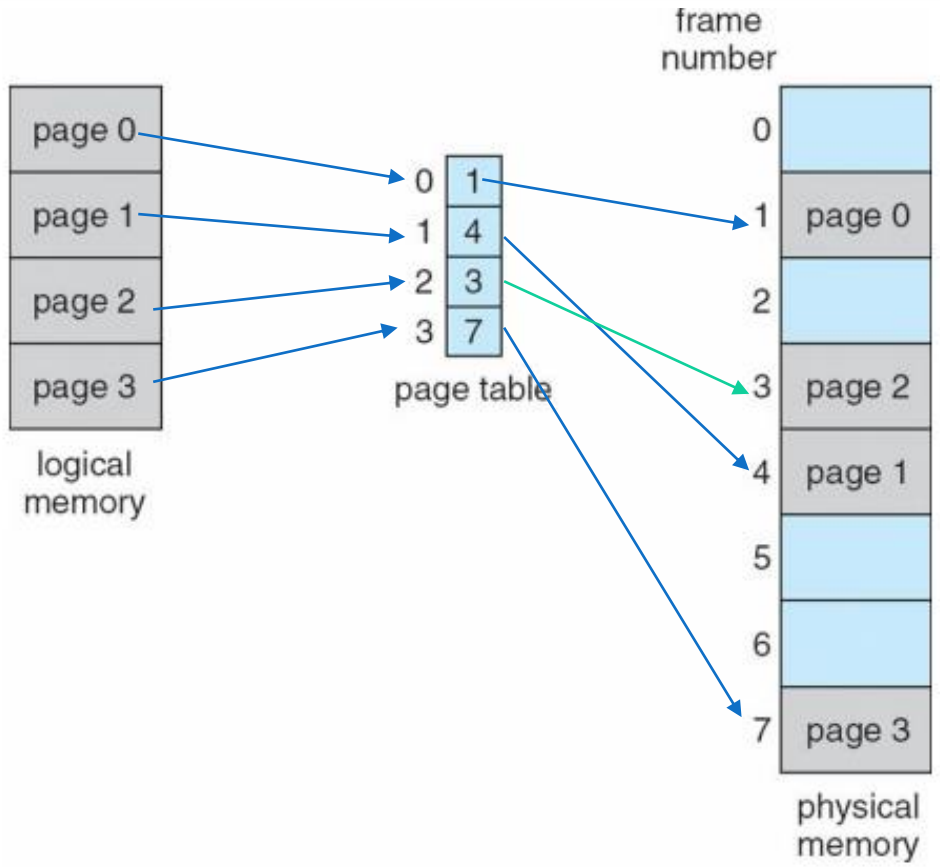


  - 2^m mantıksal adres uzayı ve 2^n sayfa boyutu ile

Sayfalama Donanımı



Mantıksal ve Fiziksel Hafızanın Sayfalama Modeli



BİTTİ