Bellek Yönetimi - 2

Görüntü Bellek

- proses çalışması için bellekte olmalı
  - tamamı bellekte olmalı mı?
- fiziksel adresler proses belleğe yüklendikten sonra belli olur
  - tüm çalışması boyunca aynı olmayabilir
- prosesin parçaları bellekte birbirini izleyen gözlerde olmak zorunda değil
- prosesin tamamı bellekte olmak zorunda değil

Görüntü Bellek

- kullanılmayan bölgeler ikincil bellekte
- başlangıçta prosesin bir kısmı ana belleğe yüklenir
  - yerleşik küme (resident set)
- Erişilmek istenen kısım bellekte yoksa
  - Sayfa hatası - kesme oluşur
  - proses bloke edilir
  - istenen kısım belleğe yüklenir
    - işletim sistemi G/Ç isteği üretir
    - G/Ç bitince kesme oluşur; bekleyen proses uyandırılıp HAZIR durumuna geçirilir

Görüntü Bellek

- ana bellekte HAZIR durumunda daha fazla proses
  - çoklu programlama daha etkin
  - belleğe sadece proseslerin gerekli parçaları yüklenir
- boyu ana bellekten büyük olan prosesler de çalışabilir
- gerçeklemede sayfalama ve/veya segmanlama
  - donanım desteklemeli

Görüntü Bellek

- ana ve ikincil bellekte nasıl yer ayrılmalı?
  - sayfalamada sorun değil
  - segmanlar değişken boylu!
- ana bellek ⇔ ikincil bellek arası sayfa/segman taşımada neler göz önüne alınmalı?
- ana bellek doluysa hangi parça ikincil belleğe aktarılmalı?

Değişken Uzunluklu Segmanlar için Alan Ayırma Yöntemleri

- boş alanlar bağlantılı listede tutulur
  - büyüyen adreslere doğru sıralı
- her kayıtta:
  - boş alanın adresi
  - boş alanın boyu
  - sonraki boş alana işaretçi
- boşalan bellek bölgesi listeye eklenir
  - öncesi ve sonrası ile birleştirilir

Değişken Uzunluklu Segmanlar için Alan Ayırma Yöntemleri

- ilk uygun yöntemi (first-fit)
  - liste başından itibaren boyu uyan ilk boş alan ayrılır
  - artan kısım boşlar listesine eklenir
- sıradaki uygun yöntemi (next-fit)
  - liste en son ayrılan yerden itibaren taranır
  - boyu uyan ilk boş alan ayrılır
  - çevrel liste olması daha iyi
- en uygun yöntemi (best-fit)
  - boyu en uygun olan boş alan seçilir
  - tüm liste her seferinde baştan sona taranır
- en az uygun yöntemi (worst-fit)

Değişken Uzunluklu Segmanlar için Alan Ayırma Yöntemleri

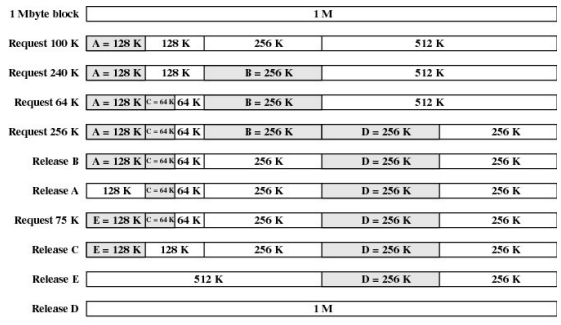
- listede boş alanlar boylarına göre sıralanırsa:
  - en uygun = ilk uygun
  - birbirini izleyen boş alanların birleştirilmesi zor
- veya belli uzunluktaki alanların listedeki yerine işaretçi tutulur
  - işaretçilerin güncellenmesi vakit alır

Değişken Uzunluklu Segmanlar için Alan Ayırma Yöntemleri

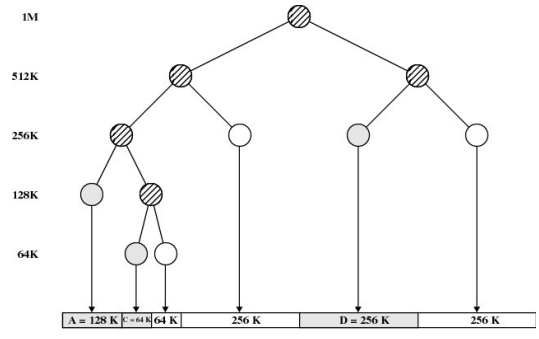
- "buddy" yöntemi
  - tüm bellek 2^k uzunluklu bloklar halinde ayrılır
  - tüm bellek 2^s uzunluğunda olsun
    - (s+1) adet bağlantılı liste
    - $2^0, 2^1, 2^2, ..., 2^s$
  - liste(k): 2^k uzunluklu bloklara işaretçi (k=0,1,...,s)
  - başlangıçta liste(s) belleğin ilk gözüne işaret eder
    - Diğer listeler başlangıçta boş

"Buddy" Yöntemi

- örneğin 2^k boyutunda bölge istensin
  [$>2^{k-1}$ ve $\leq 2^k$]
  - liste(k) boş ise liste(k+1) denenir
    - boş değilse alınan alan ikiye bölünür
    - bir alan liste(k)'ya eklenir
    - diğer alan ile istek karşılanır
  - tüm listeler boş ise bellek isteği karşılanamaz
- geri verilen bölgeler listenin uygun yerlerine eklenir
  - "buddy" alanlar birleştirilir



Buddy sistemi örneği



Buddy yönteminin ağaç ile temsili

Sıkıştırma

- bellek doldukça sorun olabilir
  - en büyük boş alandan daha büyük yer istenebilir
    - istek geri çevrilir
    - bir alan ana bellekten ikincil belleğe atılır
    - bellekteki boş alanlar sıkıştırılıp büyük yer açılır

İkincil Bellekten Sayfa Taşıma Yöntemleri (Fetching)

- ikincil bellek $\Rightarrow$ ana bellek arası sayfalar hangi kritere göre taşınacak?
  - önceden sayfalama (pre-paging)
    - yakın gelecekte erişilecek sayfalar kestirilebilir
    - istek oluşmadan sayfaları belleğe yükle
    - sayfa hatası daha az
    - hatalı kestirimde yüksek maliyet
    - örneğin, veri sayfaları için uygun bir yöntem
  - istek üzerine sayfalama (demand paging)
    - sayfaları istek olduğu zaman belleğe al

Ana Bellekteki Sayfaların Değiştirilmesi

- ana bellekte boş yer yoksa, bir sayfa taşınıp yer açılmalı
  - oluşabilecek sayfa trafiğine dikkat
  - bellekten atılan sayfaya hemen ihtiyaç duyulmamalı
    - "thrashing": vakit kaybı
  - temel hedef YARARLI sayfaların atılmaması
    - yakın zamanda erişilmeyecek sayfalar atılmalı
  - bazı işletim sistemi sayfaları bellekten atılamaz
    - çerçeve kilitleme bir bit ile yapılır
  - Sayfa seçiminin hedefi iki farklı alan olabilir :
    - yerel: yürütülmekte olan prosesin sayfaları arasından
    - global: tüm sayfalar arasından

Ana Bellekteki Sayfaların Değiştirilmesi

- rasgele seçim
  - gerçeklemesi kolay
  - YARARLI sayfa da seçilebilir!
- ilk giren ilk çıkar – FIFO
  - bellekte en uzun süre kalmış olan sayfa seçilir
  - başarım kötü olabilir-en eski sayfa artık kullanılmayan sayfa olmayabilir
- BIFO (biased FIFO)
  - i. prosese ayrılmış n_i sayfa arasından sayfa seçilir, n_i sayfa arasında FİFO uygulanır
  - Farklı proseslerin sayfa sayıları farklı olabilir ($n_i \neq n_j$)
  - zaman içinde n_i değerleri değişebilir- farklı prosesler daha fazla bellek alanına sahip olurlar

Ana Bellekteki Sayfaların Değiştirilmesi

- en uzun zamandır erişilmemiş olan – LRU (Least Recently Used)
  - Gerçekleme maliyeti yüksek, donanım desteği gerekli
  - Her sayfa için bir kaydın yer aldığı bir tablo oluşturulur ve tabloda her sayfa için en son erişimden bu yana geçen süre tutulur
  - bellekteki her sayfanın erişim biti (use bit) vardır, her referanstan sonra bu bit birlenir
  - her zaman birimi sonunda, sayfa bitleri taranır
    - 1 olanlar 0 yapılır ve tablodaki süre sayacları sıfırlanır
    - 0 olanların kullanım süre sayaçları bir arttırılır
    - Bellekten bir sayfa boşaltılacak ise, süre sayaç değeri en yüksek olan (en uzun zamandır erişilmemiş) sayfa seçilir.

Ana Bellekteki Sayfaların Değiştirilmesi

- önceden tanımlı öncelikler
  - Derleyici hangi sayfaların yüksek öncelikli olması gerektiği konusunda bilgi verebilir (bellekte kalırlar)
  - prosesin kodunun yapısal özellikleri bu bilgiyi sağlar
    - yerellik prensibi (principle of locality): kod ve veri erişimlerinin aynı bölgede kalması eğilimi
    - döngüler

Ana Bellekteki Sayfaların Değiştirilmesi

- sistem tarafından tanımlanan öncelikler
  - iş sıralamada kullanılan öncelikler burada da kullanılabilir
  - sayfa hatası durumunda en düşük öncelikli prosesin bir sayfası atılır
    - örneğin LRU ile
  - en düşük önceliklinin son sayfası da atılırsa bellekte yer açılana kadar bekler
    - SORUN: yüksek öncelikli proseslerin kullanılmayan bazı sayfaları bellekte sürekli durabilir

Ana Bellekteki Sayfaların Değiştirilmesi

- birleşik yöntemler
  - bazı kurallar birleştirilebilir
  - azalan tercih sırasına göre:
    - pasif (bloke) prosesin okuma-erişimli sayfası
    - pasif (bloke) prosesin yazma-erişimli sayfası
    - önceki ½ saniyede erişilmemiş olan işletim sistemi sayfası
    - G/Ç bekleyen prosesin sayfası
    - aktif bir prosesin bir sayfası