30.12.2010

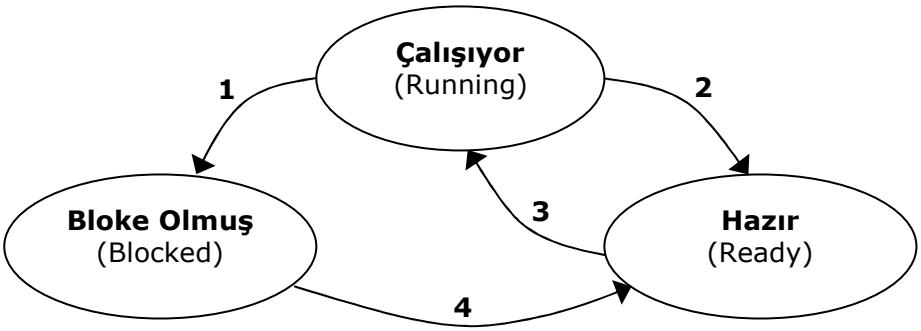
İ.K.Ü. FEN-EDEBİYAT FAKÜLTESİ
MATEMATİK-BİLGİSAYAR BÖLÜMÜ

2010-2011 GÜZ YARIYILI "MC 78K İŞLETİM SİSTEMLERİ"
II.VİZE ÇÖZÜMLERİ

1) **a)** Proseslere ilişkin **yaşam çevrimi** diyagramını çiziniz ve durumlar arası geçişlerin hangi olaylarla sağlandığını belirtiniz.

b) İkili semafor (*binary semaphore*) ve **sayaç semafor** (*counting semaphore*) kavramlarını açıklayınız ve bu semaforların ne amaçlarla kullanıldığını yazınız.

a)



1 : Çalışıyor durumda olan proses, o an için işlemcinin atandığı prosestir; dolayısıyla tek işlemcili sistemlerde çalışıyor durumda en fazla bir proses olabilir. Bu proses çalışmasına devam edebilmek için bir giriş/çıkış işlemine (örneğin; *kullanıcı girişi, disk erişimi*) ihtiyaç duyduğunda veya proses kendisini bloke eden bir sistem çağrısı (örneğin : *sleep*) gerçekleştirdiğinde çalışıyor durumdan bloke duruma geçer.

2 : İşletim sistemi çalışmakta olan prosesi belirli nedenlerden dolayı kesebilir (örneğin; *prosese ayrılan çalışma süresinin dolması, daha yüksek öncelikli bir prosesin hazır kuyruğuna varması*). Böyle bir durumda proses, çizelgeleyici modül (*scheduler*) tarafından çalışıyor durumdan hazır durumuna alınır.

3 : Çizelgeleyici işlemciye yeni bir proses atayacağı zaman hazır durumdaki proseslerden birini belirli bir algoritmaya göre (örneğin; *round-robin, öncelikli sıralama*) işlemcide çalışmak üzere seçer.

4 : Bloke durumdaki proses, bloke olmasına neden olan dış işlemi tamamladıktan sonra (veya *sleep* çağrısı ile bloke olan bir proses, *wakeup* çağrısı ile uyandırıldığı zaman) hazır durumuna geçer.

b) **İkili Semafor (*Binary Semaphore*) :** Sadece 0 ve 1 değerlerini alabilen semafor. İkili semafor kritik bölge giriş/çıkış kontrollerinin kilidi olarak (karşılıklı dışlamayı gerçekleyebilmek amacıyla) kullanılabilir. Kritik bölgesine girmek isteyen bir proses, ikili semafor üzerinde *down* işlemi yaparak semaforun değerini 0'a çeker. Bu proses kritik bölgesindeyken, başka prosesler kendi kritik bölgelerine giremeyeceklerdir; zira bu prosesler ikili semafor üzerinde *down* işlemi yaptıklarında bloke olurlar. Kritik bölgesine girmiş olan proses, kritik bölgeden çıkarken ikili semafor üzerinde *up* işlemini gerçekleştirir; eğer bu semafor üzerinde bloke olmuş prosesler varsa biri uyandırılır; yoksa semaforun değeri 1 yapılır.

Sayaç Semafor (*Counting Semaphore*) : İkili semaforuna aksine 1' den büyük değerler alabilen semafor. Genellikle bir kaynağa erişim kontrolünde kullanılır; semaforun başlangıç değeri boştaki (kullanılabilir) kaynak sayısı olacak şekilde ayarlanır. Kaynağa erişen her proses sayaç semafor üzerinde *down* işlemi yaparak semaforun değerini 1 azaltır; kaynak kullanımını sonlandıran her proses ise sayaç semafor üzerinde *up* işlemi yaparak semaforun değerini 1 arttırır.

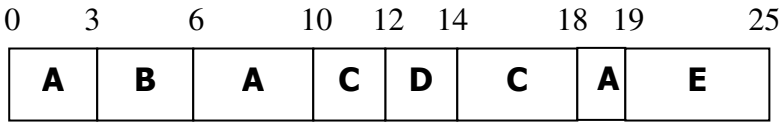
2) İş sıralama yöntemi olarak *çok seviyeli kuyruk* modeli kullanılan bir sistemde yer alan 3 kuyruğa ilişkin algoritmalar aşağıda verilmiştir. Buna göre yine aşağıda verilen tablodaki proseslerin sisteme dahil olmaları durumunda oluşacak zaman diyagramını çizerek, proseslere ilişkin ortalama bekleme süresini hesaplayınız.

- **1. kuyruk :** En kısa iş kalan ilk (*Shortest Job Remaning First*)
- **2. kuyruk :** Çevrimsel Sıralamalı (*Round Robin, quantum = 4 mSn*)
- **3. kuyruk :** FIFO

Prosesler	**Varış Zamanı**	**İşlem Süresi**
A	0	8
B	3	3
C	7	6
D	12	2
E	19	6

Sistem Modeli :

Zaman diyagramı :



Bekleme Süreleri :

A → (0-0) + (6-3) + (18-10) = 11 mSn
B → (3-3) = 0 mSn
C → (10-7) +(14-12) = 5 mSn
D → (12-12) = 0 mSn
E → (19-19) = 0 mSn

Ortalama Proses Bekleme Süresi :
(11+0+5+0+0) / 5 = 16/5 = **3.2** mSn

3) Bir sistemdeki ana belleğin belirli bir andaki durumu aşağıdaki gibidir :

25K (Boş) - 30K (*Dolu-A*) - **30K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)

a) Aşağıdaki sıraya uygun olarak ana belleğe yerleşmek isteyen prosesleri; *first fit*, *best fit* ve *worst fit* algoritmalarına göre yerleştiriniz; dış parçalanma durumlarını inceleyiniz.
P0 → 15K P1 → 20K P2 → 5K P3 → 25K
b) Sisteme bir **P4** prosesi dahil olduğunda, *next fit* algoritmasına göre bir dış parçalanma oluştuğuna göre bu P4 prosesinin minimum boyutunun ne olabileceğini bulunuz.
c) Sistemde daha önceden var olan 3 prosesten (A,B,C) sadece **B prosesi** çalışma anında bellekte yer değiştirebildiğine göre ve b şıkkında oluşan dış parçalanmanın bellek sıkıştırma (*compaction*) yöntemiyle bile çözülemediği bilindiğine göre **P4** prosesinin minimum boyu ne olabilir?

a) İlk Uygun Boşluk (*First Fit*) :
P0 → 15K (Dolu-P0)-10K (Boş) - 30K (*Dolu-A*) - **30K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P1 → 15K (Dolu-P0)-10K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P2 → 15K (Dolu-P0)- 5K (Dolu-P2) - 5K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)

P3 → 15K (Dolu-P0)- 5K (Dolu-P2) - 5K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **25K (Dolu-P3) - 10K (Boş)** - 10K (*Dolu-C*)

En Uygun Boşluk (*Best Fit*) :
P0 → 15K (Dolu-P0)-10K (Boş) - 30K (*Dolu-A*) - **30K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P1 → 15K (Dolu-P0)-10K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P2 → 15K (Dolu-P0)- 5K (Dolu-P2) - 5K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)

P3 → 15K (Dolu-P0)- 5K (Dolu-P2) - 5K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **25K (Dolu-P3) - 10K (Boş)** - 10K (*Dolu-C*)

En Büyük Boşluk (*Worst Fit*) :
P0 → 25K (Boş) - 30K (*Dolu-A*) - **30K (Boş)** - 5K (*Dolu-B*) - **15K (Dolu-P0) - 20K (Boş)** - 10K (*Dolu-C*)
P1 → 25K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **15K (Dolu-P0) - 20K (Boş)** - 10K (*Dolu-C*)
P2 → 5K (Dolu-P2) - 20K (Boş) - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **15K (Dolu-P0) - 20K (Boş)** - 10K (*Dolu-C*)
P3 → 25K boyutundaki bu proses hiçbir boşluğa yerleşemez. **Dış Parçalanma !**

b) Next-fit algoritmasına göre proseslerin boşluklara ne şekilde yerleştiklerini bulmamız gerekir.
Sıradaki Uygun Boşluk (*Next Fit*) :
P0 → 15K (Dolu-P0) - **10K (Boş)** - 30K (*Dolu-A*) - **30K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P1 → 15K (Dolu-P0) - **10K (Boş)** - 30K (*Dolu-A*) - **20K (Dolu-P1)** - **10K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P2 → 15K (Dolu-P0) - **10K (Boş)** - 30K (*Dolu-A*) - **20K (Dolu-P1)** – **5K (Dolu-P2) - 5K (Boş)** - 5K (*Dolu-B*) - **35K (Boş)** - 10K (*Dolu-C*)
P3 → 15K (Dolu-P0) - **10K (Boş)** - 30K (*Dolu-A*) - **20K (Dolu-P1)** – **5K (Dolu-P2) - 5K (Boş)** - 5K (*Dolu-B*) – **25K (Dolu-P3)** - **10K (Boş)** - 10K (*Dolu-C*)

Bu son durumda oluşan boşlukların en büyüğü 10K boyutundadır. Bir dış parçalanma oluşması için yeni gelecek olan P4 prosesinin boyutunun 10K'dan büyük olması gerekmektedir. Bunu sağlayan minimum değer 10K+1 byte = **10241** byte'dır.

c) Bir önceki şıktaki son durumda B prosesini yer değiştirdiğimizde (başka bir boşluğa taşıdığımızda) burada oluşan yeni boşluk 10K olacaktır (5K= B prosesinin boyu + 5K= B prosesinin solundaki boşluk). Dolayısıyla b şıkkındaki çözüm geçerliliğini korur.

4) Bir sistemde bellek yönetimi için **sayfalama** (*paging*) yöntemi kullanılmaktadır. Ana belleğin boyunun **2MB**, sayfa boyunun ise **4KB** olduğu bilinmektedir. **71.300 byte** uzunluğundaki bir **P** prosesinin sayfa tablosunun ilk 3 satırında sırası ile 80,26,11 sayıları bulunmaktadır. Buna göre;

a) Görüntü ve fiziksel adresin ve alt bölümlerinin boyutlarını bulunuz.
b) **4095.** ve **4096.** görüntü adreslerin ana bellekteki konumları arasında **kaç byte fark** olduğunu bulunuz.
c) P prosesinin boyutunda **en az kaç adet proses** tamamen belleğe yerleşirse, oluşan toplam iç parçalanma bir P prosesi boyutlarına ulaşır ?

a) Sayfa boyu = 4KB = $\mathbf{2^{12}}$ byte
Ana bellek boyu = 2MB = $\mathbf{2^{21}}$ byte
Ana bellekte toplam $2^{21} / 2^{12} = \mathbf{2^9}$ çerçeve vardır.
Bu bilgiler ışığında fiziksel adresin toplam **21 bit** olduğunu; bu 21 bitin ilk 9 bitinin çerçeve, geri kalan 12 bitinin ise sayfa içi adres olduğunu bulmuş oluruz.

Proses boyu 71300 byte olarak verilmiş. Bu proses toplam 71300/4096 = 17.40 sayfalık yer kaplar, yani prosesin 18 sayfası vardır. [0-17] arasındaki sayıları **5** bit ile gösterebiliriz.
Bu bilgiler ışığında prosese ilişkin görüntü adresin ilk 5 biti sayfa numarası, bundan sonraki 12 biti ise sayfa içi adres bilgisinden oluşur. Görüntü adresin toplam uzunluğu 5+12=17 bittir.
Bu 17 bit sonucuna daha kolay bir yoldan da ulaşabiliriz : 71300 sayısı minimum **17 bit** ile gösterilebilir.

f (9 bit)	d (12 bit)

Fiziksel Adres

p (5 bit)	d (12 bit)

Görüntü Adres

b) Sayfa boyu 4096 byte olduğu için **4095.** görüntü adres, 0. sayfanın 4095. (yani son) adresidir. **4096.** adres ise 1. sayfanın 0. (yani ilk) adresidir.
0. sayfanın ana bellekte 80. çerçevede olduğu biliniyor. Buna göre 4095. adres 80x4096+4095 = 331775. fiziksel adrestedir.
1. sayfanın ana bellekte 26. çerçevede olduğu biliniyor. Buna göre 4096. adres 26x4096+0 = 106496. fiziksel adrestedir.
Sonuç : 4095 ve 4096 görüntü adresleri, proses adres uzayında ardışıl adreslerdir. Fakat bu adreslerin farklı sayfalarda olmasından ve bu sayfaların ana bellekte ardışıl olmayan çerçevelerde olmasından dolayı bu iki adresin ana bellekteki konumları arasındaki fark 331775-106496 = **225279** byte'dır.

Diğer bir çözüm yolu : Sorulan görüntü adreslerden biri 26. çerçevenin ilk adresi, diğeri 80. çerçevenin son adresidir. 81. çerçevenin ilk adresini baz alırsak, bu iki adres arasında tam olarak 81-26=55 çerçeve fark vardır. 55 çerçeve → 55x4096=225280 byte uzunluğundadır. Bu değerden eklediğimiz 1 byte'ı düşersek **225279** sonucuna ulaşırız.

c) P prosesi (a) şıkkında da bulduğumuz gibi 18 sayfaya bölünmüştür. Son (18.) sayfanın tamamı kullanılmamakta olup, kullanılmayan kısım iç parçalanma sebebidir :
(18 x 4096)-71300 = **2428** byte iç parçalanma vardır.

P prosesi boyutundaki her proses ana bellekte 2428 byte'lık bir iç parçalanma oluşturacaktır.

71300 / 2428 = 29.36

Sonuç olarak, P prosesi boyutlarında **en az 30 proses** ana belleğe yerleştiğinde oluşan toplam iç parçalanma boyutu bir P prosesi boyutuna ulaşır.

5) Sayfa boyu **2KB** ve ana belleği **4 çerçeveden** oluşan bir sistemde sayfa hatası oluşması durumunda yapılan sabit disk+bellek işlemi **300 nSn**, sayfa hatası olmaması durumunda yapılan bellek erişimi ise **5 nSn**'dir. Böyle bir sistemde çalışmakta olan **16KB** uzunluğundaki bir **P** prosesi için belirli bir zaman diliminde şu görüntü adreslere referanslar üretildiği bilinmektedir:

1060, 4096, 15200, 8340, 12010, 8192, 8191, 14500, 8842, 8844, 5000, 12750, 4100, 10439, **X**.

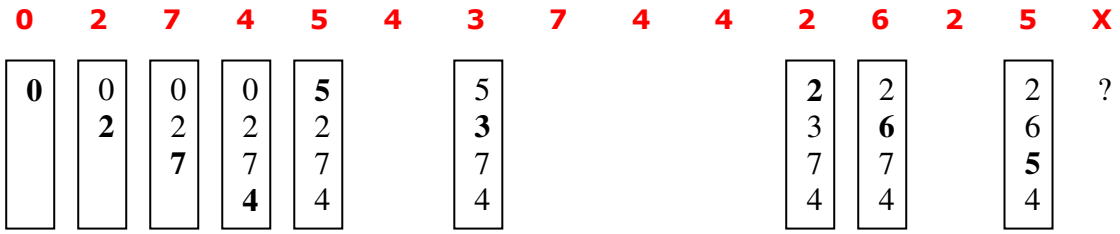
Bu sistemde sayfa yerleşim algoritması olarak (LRU-*Least Recently Used*) yöntemi kullanıldığında yukarıdaki adresler için ortalama bellek erişim süresi **182 nSn** olduğuna göre; **X** adresinin alabileceği değer aralıklarını bulunuz.

Çözüm :

Öncelikle verilen görüntü adreslerin prosesin hangi sayfalarında olduklarını bulmak gerekir. Sayfa boyu 2KB (2048 byte) olarak verildiği için her adresin hangi 2048'lik dilimin içine düştüğünü bulmak basit bir bölme işlemi ile bulunabilir.

Adres:1060, 4096, 15200, 8340, 12010, 8192, 8191, 14500, 8842, 8844, 5000, 12750, 4100, 10439, **X**.
Sayfa: 0 , 2 , 7 , 4 , 5 , 4 , 3 , 7 , 4 , 4 , 2 , 6 , 2 , 5 , X.

Ana belleğin 4 çerçeveden oluştuğu bilgisi verilmiş. Sayfa referans dizisi de elimizde olduğuna göre LRU algoritmasına göre sayfa yerleşimini çizebiliriz :



Toplam 15 bellek referansından 9 tanesi sayfa hatası üretmiş, 5 tanesinde ise sayfa hatası yok. Son bellek referansı (X) hakkında ise henüz bir bilgimiz yok.

Ortalama bellek erişim süresi **182 nSn** olarak verilmiş. Bu durumda toplam bellek erişim süresi :
182x15= **2730** nSn'dir.

Bilinen 9 sayfa hatası → 9x300 = 2700 nSn
Bilinen 5 normal erişim → 5x5 = 25 nSn
Bilinen toplam erişim süresi : 2725 nSn

Bu sonuçtan hareketle en son bellek referensanın 2730-2725 = **5 nSn**'lik bir sürede gerçekleştiğini; dolayısıyla **sayfa hatası oluşturmadığını** bulabiliriz. Yani X adresi o an için ana bellekte bulunan 2,6,5 veya 4 numaralı sayfalardan birinin içindedir. Bu adresin alabileceği değer aralıkları :

2. sayfa → [4096-6143]
4. sayfa → [8192-10239]
5. sayfa → [10240-12287]
6. sayfa → [12288-14335]

şeklindedir.