

6

İŞ SIRALAMA

İş Sıralamanın Amaçları

- İşleri zaman içinde işlemciye yerleştirmek
- Sistem hedeflerine uygun olarak:
 - İşlemci verimi
 - Cevap süresi (response time)
 - Debi (throughput)

İş Sıralama

- Çok programlı ortamlarda birden fazla proses belirli bir anda bellekte bulunur
- Çok programlı ortamlarda prosesler:
 - işlemciyi kullanır
 - bekler
 - giriş çıkış bekler
 - bir olayın olmasını bekler

İş Sıralama Türleri - 1

- Uzun vadeli iş sıralama: Yürütülecek prosesler havuzuna ekleme kararı
- Orta vadeli iş sıralama: Kısmen veya tamamen bellekte yer alacak prosesler arasından alma kararı

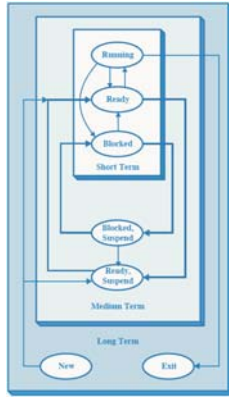
İş Sıralama

- Çok programlı ortamlarda işlemcinin etkin kullanımı için iş sıralama yapılır
- İş sıralama işletim sisteminin temel görevleri arasında
 - işlemci de bir sistem kaynağı

İş Sıralama Türleri - 2

- Kısa vadeli iş sıralama: Koşabilir durumdaki proseslerden hangisinin seçileceği kararı
- G/Ç sıralama: G/Ç bekleyen hangi prosesin isteğinin karşılanacağı kararı

İş Sıralama Türleri



Uzun Vadeli İş Sıralama

- İki tip karar söz konusu:
- sisteme yeni proses eklenebilir mi?
 - çoklu programlama düzeyi tarafından belirlenir
 - Proses sayısı arttıkça her çalışmada proses başına düşen işlemci zamanı azalır : bu nedenle bu sınırlanabilir

Uzun Vadeli İş Sıralama

- İşlenmek üzere sisteme hangi proseslerin alınacağına karar verilmesi
- Çoklu programlamanın düzeyini belirler

Uzun Vadeli İş Sıralama

- uzun vadeli iş sıralama,
 - sistemde herhangi bir iş sonlandığında çalışabilir
 - işlemcinin boş bekleme zamanı bir eşik değerini aşarsa çalışabilir

Uzun Vadeli İş Sıralama

- Sisteme alınan iş:
 - bazı sistemlerde kısa vadeli sıralama için ilgili kuyruğa yerleşir
 - bazı sistemlerde orta vadeli sıralama için ilgili kuyruğa yerleşir (yeni gelen proses doğrudan belleğe alınmaz)

Uzun Vadeli İş Sıralama

- hangi prosesleri kabul etmeli? hangilerini etmemeli?
 - geliş sıralarına göre seçilebilir
 - bazı kriterlere göre seçilebilir
 - öncelik
 - beklenen toplam çalışma süresi
 - giriş/çıkış istekleri

İş Sıralama

6

Uzun Vadeli İş Sıralama

Örneğin gerekli bilgi varsa,

- ▶ G/Ç yoğun işlerle işlemci yoğun işleri dengeli tutmak istenebilir
- ▶ G/Ç isteklerini dengeli tutmak için istenen kaynaklara göre karar verilebilir

İşletim Sistemleri

245

İş Sıralama

6

Kısa Vadeli İş Sıralama

- ▶ Belirli sistem başarımlarını gerçekleştirme amacı
- ▶ Algoritmaların başarımlarını kıyaslaması için kriterler belirlenir

İşletim Sistemleri

248

İş Sıralama

6

Uzun Vadeli İş Sıralama

- ▶ Etkileşimli kullanıcılar (zaman paylaşımı bir sistemde)
 - sisteme bağlanma istekleri oluşur
 - bekletilmez
 - belirli bir doyma noktasına kadar her gelen kabul edilir
 - doyma değeri belirlenir
 - sistem doluysa kabul edilmeyip hata verilir

İşletim Sistemleri

246

İş Sıralama

6

Kısa Vadeli İş Sıralama

- ▶ İş Çalıştırıcı (dispatcher)
- ▶ Bir olay olduğunda çalışır
 - Saat kesmesi
 - G/Ç kesmesi
 - İşletim sistemi çağrıları
 - Sinyaller

İşletim Sistemleri

249

İş Sıralama

6

Orta Vadeli İş Sıralama

- ▶ “Swap” işleminin bir parçası
- ▶ Çoklu programlamanın derecesini belirleme gereği
- ▶ Sistemin bellek özellikleri ve kısıtları ile ilgilidir
- ▶ Belleğe alınacak proselerin bellek gereksinimlerini göz önüne alır
- ▶ Bellek yönetim birimine de bağlıdır

İşletim Sistemleri

247

İş Sıralama

6

Kısa Vadeli İş Sıralama Kriterleri

- ▶ Kullanıcıya yönelik
 - Yanıt süresi: isteğin iletilmesinden çıkış alınana kadar geçen süre
 - Amaç kabul edilen yanıt süresi ile çalışan iş/kullanıcı sayısını maksimize etmek
 - doğrudan kullanıcılara iyi servis vermek amaçlı

İşletim Sistemleri

250

İş Sıralama 6

Kısa Vadeli İş Sıralama Kriterleri

- ▶ Sisteme yönelik
 - İşlemcinin etkin kullanımı
 - Örnek: debi (işlerin tamamlanma hızı)
 - önemli bir kriter ancak kullanıcı prosesler doğrudan bunu göremez
 - sistem yöneticileri açıısından daha önemli bir kriter

İşletim Sistemleri

251

İş Sıralama 6

Önceliklerin Kullanımı

- ▶ İş sıralayıcı yüksek öncelikli işleri önce seçer
- ▶ Her öncelik düzeyine ilişkin *Hazır* kuyruğu tutar
- ▶ Düşük öncelikli prosesler *Açlık* durumu ile karşılaşabilir
 - prosesin önceliğinin yaşı ve geçmişi ile bağlantılı olarak değiştirilmesi

İşletim Sistemleri

254

İş Sıralama 6

Kısa Vadeli İş Sıralama Kriterleri

- ▶ Başarıma yönelik
 - Nicel
 - Debi ve yanıt süresi gibi ölçülebilir
- ▶ Başarımla ilgili olmayan
 - Nitel ve çoğu durumda ölçülemez
 - Tahmin edilebilirlik; kullanıcılara sunulan hizmetin, diğer kullanıcılardan bağımsız olarak her zaman belirli bir düzeyde olması istenir

İşletim Sistemleri

252

İş Sıralama 6

İş Sıralama

- ▶ çalışan proses ortasında kesilip “hazır” kuyruğına eklenebilir (preemptive)
 - çalışma anları:
 - yeni bir iş gelince
 - bloke olan bir iş hazır olunca
 - belirli zaman aralıkları ile
 - sisteme yük getirir ama proseslere hizmet kalitesi artar

İşletim Sistemleri

255

İş Sıralama 6

Kısa Vadeli İş Sıralama Kriterleri

- ▶ Tüm kriterler birden sağlanamayabilir
 - bir kriterin sağlanamaması bir diğerini engelleyebilir
 - örneğin yanıt cevabını kısa tutma amaçlı bir algoritmaya göre işler sık sık değiştirilir. Bu sisteme yük getirir ve debi düşer.

İşletim Sistemleri

253

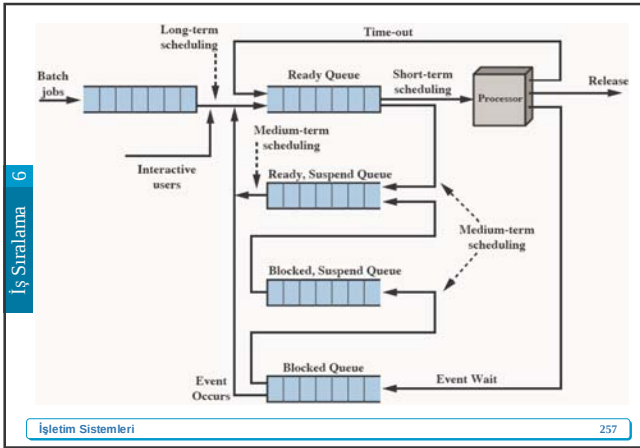
İş Sıralama 6

İş Sıralama

- ▶ işlemcide koşturmakta olan proses
 - sonlanana
 - bloke olana
- kadar kesilmeden koşar (nonpreemptive).

İşletim Sistemleri

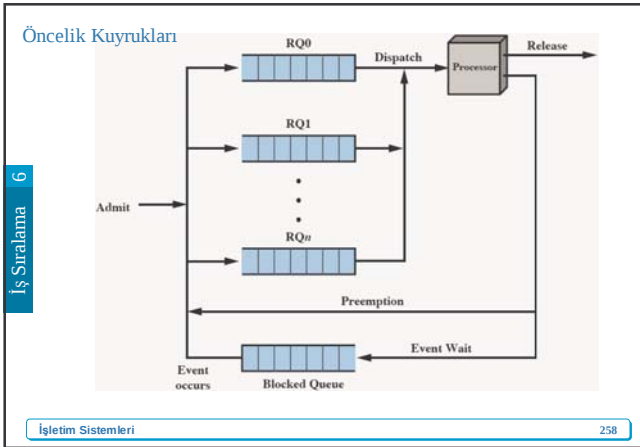
256



Geliş Sırasına Göre (FCFS)

- Her proses Hazır kuyruğuna girer
- Çalışmakta olan prosesin çalışması durunca *Hazır* kuyruğundaki en eski proses seçilir
- Uzun süre çalışacak prosesler için daha iyi
 - Kısa bir proses çalışmaya başlamadan önce çok bekleyebilir

İşletim Sistemleri 260



Geliş Sırasına Göre (FCFS)

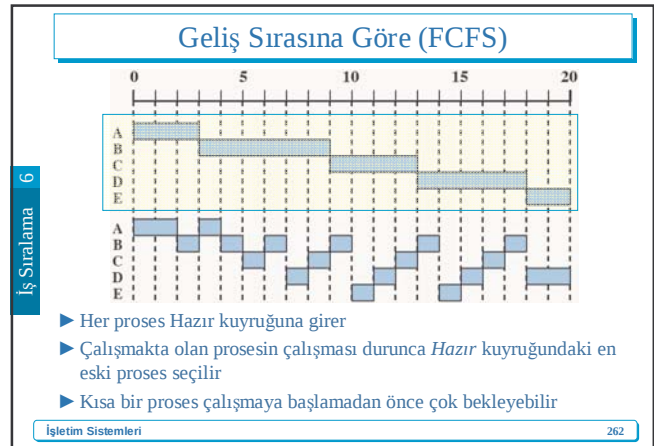
- G/Ç yoğun prosesler için uygun değil
 - G/Ç birimleri uygun olsa da işlemciyi kullanan prosesleri beklemek zorundalar
 - kaynakların etkin olmayan kullanımı

İşletim Sistemleri 261

İş Sıralama Örneği

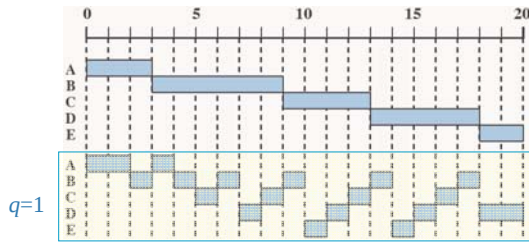
Proses	Varış Zamanı	Servis Süresi
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

İşletim Sistemleri 259

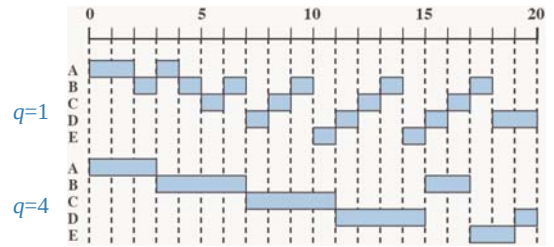


Taramalı (Round-Robin)

- Saate bağlı olarak prosesleri işlemciden almak (preemptive)
- İşlemcinin her seferde kullanılacağı süre birimi belirlenir



Taramalı (Round-Robin)



Taramalı (Round-Robin)

- Saat kesmeleri periyodik zaman aralıkları ile üretilir - *zaman dilimi*
- zaman dilimi uzunluğu belirlenmesi gerekir
- kısa zaman dilimleri uygun değil
 - kısa prosesler hızla tamamlanır ama
 - sisteme çok yük oluşur

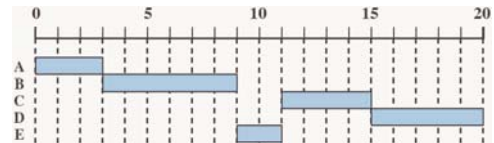
En Kısa İş Önce

- Prosesler kesilmeden çalışıyor
- Servis süresi en kısa olacağı bilinen/tahmin edilen proses seçilir
 - süre tahmin edilmesi veya bildirilmesi gerekir; tahmin edilenden çok uzun süren prosesler sonlandırılabilir
- Kısa prosesler önce çalışmış olur
- Uzun prosesler için *Açlık* durumu olası

Taramalı (Round-Robin)

- Zaman dilimi yaklaşık olarak ortalama bir etkileşim süresi kadar olmalı
- Zaman dilimi en uzun prostesten uzun olursa geliş sırasına göre olan yöntemle aynı
- Saat kesmesi geldiğinde koştakta olan proses *Hazır* kuyruğuna konur ve sıradaki iş seçilir

En Kısa İş Önce



Proses	Varış Zamanı	Servis Süresi
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

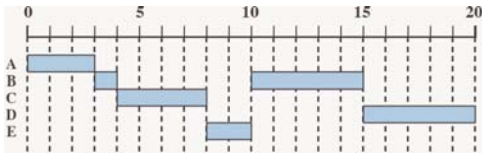
En Az Süresi Kalan Önce

- En kısa iş önce yönteminin proseslerin kesilmesine izin veren hali
- İşleme zamanı önceden kestirilmeli
- Proseslerin kalan süreleri tutulmalı

Geri Beslemeli (feedback)

- İş sıralama
 - zaman dilimi ve
 - dinamik önceliklere göre yapılır.
- Öncelik kuyrukları
 - kesilen proses bir düşük öncelikli kuyruğa geçer

En Az Süresi Kalan Önce



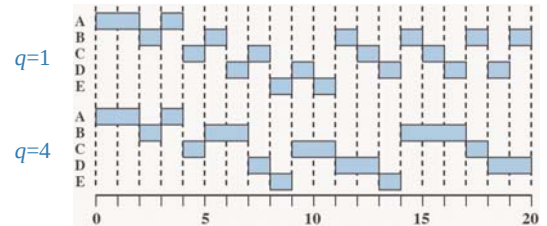
Geri Beslemeli (feedback)

- Yeni ve kısa işler daha çabuk sonlanır
- Öncelik kuyrukları içinde geliş sırası göz önüne alınır
 - en düşük öncelikli kuyruktan daha alta inmek mümkün olmadığından bu kuyrukta kalır (taramalı yöntem uygulanır)

Geri Beslemeli (feedback)

- Proseslerin toplam çalışma süreleri ile ilgili bilgi gerektirmez
- Prosesin daha ne kadar çalışma süresi kaldığı bilinmiyor
- Daha uzun süredir koşan prosesleri cezalandır

Geri Beslemeli



Yöntemlerin Başarım Karşılaştırması

- İş sıralama yöntemi seçilirken yöntemlerin başarım karşılaştırmaları önem kazanır
- Kesin karşılaştırma mümkün değil

İş Sıralama Yöntemleri – Toplu Bir Bakış

	Selection Function	Decision Mode	Throughput	Response Time	Overhead	Effect on Processes	Starvation
FCFS	$\max[w]$	Nonpreemptive	Not emphasized	May be high, especially if there is a large variance in process execution times	Minimum	Penalizes short processes; penalizes I/O bound processes	No
Round Robin	constant	Preemptive (at time quantum)	May be low if quantum is too small	Provides good response time for short processes	Minimum	Fair treatment	No
SPN	$\min[r]$	Nonpreemptive	High	Provides good response time for short processes	Can be high	Penalizes long processes	Possible
SRT	$\min[r - q]$	Preemptive (at arrival)	High	Provides good response time	Can be high	Penalizes long processes	Possible
HRRN	$\max\left(\frac{w + s}{s}\right)$	Nonpreemptive	High	Provides good response time	Can be high	Good balance	No
Feedback	(see text)	Preemptive (at time quantum)	Not emphasized	Not emphasized	Can be high	May favor I/O bound processes	Possible

Yöntemlerin Başarım Karşılaştırması

- Yöntemlerin bağıl başarımları bazı etkenlere bağlı
 - Proseslerin servis sürelerinin olasılık dağılımı
 - Bağlam anahtarlamasının etkinliği
 - G/Ç isteklerini ve G/Ç alt yapısının özellikleri

Geleneksel UNIX İş Sıralama Yaklaşımı

- kullanılan iş sıralama yaklaşımı
 - çok seviyeli
 - geri beslemeli
 - her öncelik seviyesi içinde taramalı
- 1 saniyelik dilimler
 - 1 saniyede sonlanmayan proses kesilir (preemptive)

Yöntemlerin Başarım Karşılaştırması

- Kuyruk teorisi ile analiz
- Modelleme ve simülasyon ile

Geleneksel UNIX İş Sıralama Yaklaşımı

- öncelikler
 - prosesin türüne ve
 - çalışma geçmişine bağlı.
- İş sıralama algoritmasının gerçekleştirilmesinde bazı denklemler kullanılır

Geleneksel UNIX İş Sıralama Yaklaşımı

$$CPU_j(i) = \left\lfloor \frac{CPU_j(i-1)}{2} \right\rfloor$$

$$P_j(i) = Taban_j + \left\lfloor \frac{CPU_j(i)}{2} \right\rfloor + nice_j$$

$CPU_j(i)$: i. zaman diliminde j prosesinin işlemci kullanım miktarı

$P_j(i)$: i. zaman dilimi başında j prosesinin öncelik değeri
(düşük değerler yüksek öncelik anlamına gelir)

Taban_j: j prosesinin taban öncelik değeri

nice_j: kullanıcı tarafından belirlenen öncelik ayarlama faktörü

Örnek: A, B ve C prosesleri aynı anda, 60 taban önceliğine sahip olarak yaratılmış olsunlar.

- **nice** değeri göz önüne alınmayacak.
- Saat sistemi saniyede 60 kere kesiyor ve bir sayaç değerini artırıyor.

• Varsayımlar:

- Sistemde başka koşturmaya hazır proses yok
- Proseslerin herhangi bir nedenle bloke olmuyorlar.

Proses A, B ve C'nin çalışma sürelerini ve sıralarını gösteren zamanlama tablosunu çizin.

Geleneksel UNIX İş Sıralama Yaklaşımı

- Her prosesin önceliği 1 saniyede bir hesaplanır
 - iş sıralama kararı verilir
- Taban öncelik değerinin amacı, prosesleri türlerine göre belirli gruplara ayırmak

Geleneksel UNIX İş Sıralama Yaklaşımı

- Farklı taban öncelikler ile başlayan proses grupları:
 - swapper
 - blok tipi g/ç aygıt kontrolü
 - dosya yönetimi
 - karakter tipi g/ç aygıt kontrolü
 - kullanıcı prosesleri