

Bilgisayar İşletim Sistemleri BLG 312

İş Sıralama

İş Sıralama

- Çok prosesli ortamda birden fazla proses çalışmaya hazır şekilde bellekte yer alır
- MiB zamanının prosesler arasında paylaşılması: **zaman paylaşımlı çalışma**
- **İş sıralama**: bir algoritma ile hangi prosesin MiB'ni elde edeceğinin belirlenmesi

İş Sıralama Hedefleri

- **MiB kullanımı**: işlemci meşgul tutulmalı
— %40—az yüklü----%90 çok yüklü
- **Çıktı** (throughput):birim zamanda çalışmasını tamamlayan proses sayısı
- **Gerçekleşme süresi** (turnaround time):
Prosesin sistemde geçirdiği toplam süre
- **Bekleme süresi**: prosesin hazır kuyruğunda beklediği süre
- **Yanıt süresi** (response time):yanıt alma süresi

ENİYİLEŞTİRME KRİTERLERİ

- MiB kullanımı → maksimum
- Çıktı → maksimum
- Gerçekleşme süresi → minimum
- Bekleme süresi: → minimum
- Yanıt süresi: → minimum

İş Sıralama

- bir iş sıralama yönteminde göz önüne alınması gereken özellikler:
 - dengeli olmalı
 - tüm prosesler benzer şekilde etkilenmeli
 - hiç bir proses sonsuz ertelemeye maruz kalmamalı
 - birim zamanda maksimum sayıda prosese hizmet verilmeli
 - etkileşimli kullanıcıların yanıt alma süresi en aza indirilmeli

İş Sıralama

- kaynak kullanımı dengelenmeli
 - örneğin: az kullanılan kaynağı isteyen prosesi seç
- yanıt süresi ile kaynak kullanımı arasında denge sağlanmalı
 - her istendiğinde hazır olan kaynaklar bulunması
 - kaynak sayısı artar
 - kaynak kullanımı verimsiz
 - » çoğu zaman boş
 - gerçek zamanlı sistemlerde yanıt süresi önemli olduğundan kullanılabilir bir yaklaşım

İş Sıralama

- öncelik mekanizması olmalı
- sonsuz bekleme önlenmeli
 - “yaşlandırma” olası bir çözüm
 - kaynağı bekleyen prosesin önceliği bekleme süresi ile orantılı olarak artırılır
- kilit kaynakları kullanan prosesler hemen çalıştırılıp kaynakları bırakmaları sağlanmalı
- çok yüklü olma durumunda sistem çökmemeli
 - sisteme yeni proses alınmaması
 - tüm proseslere daha az hizmet vererek idare edilmesi

İş Sıralama Kriterleri

- prosesin G/Ç eğilimleri
 - MiB aldıktan hemen sonra G/Ç mı istiyor? (G/Ç yoğun)
- prosesin MiB kullanım eğilimi
 - zaman diliminin tümünde MiB mi kullanıyor? (işlem yoğun)
- etkileşimli / batch ?
- yanıt alma süresi ne kadar acil?

İş Sıralama Kriterleri

- öncelikler
- ne kadar sıklıkla sayfa hatası üretiyor?
 - çok sık ise öncelik ver, çünkü kısa süre içinde G/Ç isteği üretilip MİB serbest bırakıyor
- daha yüksek öncelikli proses tarafından yarıda kesilme sıklığı
 - çok sık ise düşük öncelik ver, çünkü bağlam değiştirme ile vakit kaybına neden oluyor

İş Sıralama Kriterleri

- ne kadar gerçek yürütme zamanı elde etmiş?
 - az ise daha fazla hizmet görmeli
 - çok ise yakında bitebilir; daha fazla hizmet görmeli
- sonlanmak için ne kadar zamanı kaldı?
 - biliniyorsa
 - az ise bir an evvel bitirmesine izin ver

İş Sıralama Türleri

- kesintili ve kesintisiz sıralama
 - kesintili = preemptive
 - kesintisiz = non-preemptive
- kesintisiz
 - Proses MİB'ni kendi isteği dışında bırakmaz
 - Proses sonlanana veya askıya alınana kadar MİB'ni kullanır

İş Sıralama

- kesintili
 - İşletim sistemi MİB'ni prosesin elinden kendi isteği dışında alabilir
 - daha yüksek öncelikli prosesler varsa
 - kabul edilebilir yanıt süreleri öngörülmediği durumlarda
 - yüksek maliyet
 - bağlam değiştirme zaman alır
 - etkin olması için bellekte çok sayıda proses hazır olmalı
 - sırası geldiğinde hemen çalışabilir olmalı
 - çalışmayan proseslerin bellekte tutulması maliyetli

İş Sıralama Algoritması

- İş sıralama algoritması ne zaman çalışır?
 - sisteme yeni bir proses gelince
 - çalışmakta olan proses sonlanınca
 - askıdaki bir proses hazır duruma geçince
 - çalışan proses askıda durumuna geçince
 - çalışan prosesin zaman dilimi dolunca (kesme)
 - bir hazır prosesin önceliği çalışan prosesin önceliğini aşarsa

Öncelik Değerleri

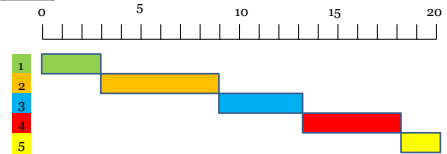
- statik x dinamik öncelikler
- statik öncelikler
 - yürütme sırasında sabit
 - kolay gerçekleştirme
 - etkin değil
- dinamik öncelikler
 - ortam değişimlerine tepki vererek değişen öncelikler
 - zor gerçekleştirme + işlemci zamanı alır
 - sistemin yanıt sürelerini iyileştirir

İş Sıralama Örneği

Proses	Geliş Anı	Hizmet Süresi
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

Proses	Geliş Anı	H.Süresi
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

Önce Gelen Önce Çalışır
First-Come-First-Served (FCFS)



- Her yeni proses hazır kuyruğuna alınır
- Bir proses sonlanınca kuyrukta yer alan en eski proses seçilir

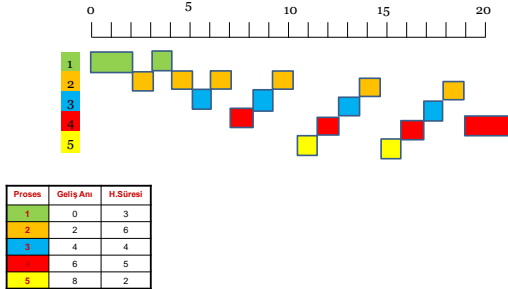
Önce Gelen Önce Çalışır

- en basit yöntem
- Prosesler sisteme giriş sıralarına göre çalışırlar
- Kesintisiz yöntem
- uzun prosesler yüzünden kısa proseslerin bekleme süresi uzayabilir
 - etkileşimli prosesler varsa yanıt alma süresi uzar
- G/Ç yoğun prosesler için uygun değil
 - G/Ç birimi boş olsa bile bekleme söz konusu olur
- ⇒ Tek başına kullanımı uygun olmayabilir, genellikle diğer yöntemlerle birlikte kullanılır
(örneğin: eşit öncelikli prosesler için)

Dönüşümlü Sıralama (Round Robin)

- FCFS benzeri ancak **kesintili** sıralama
- Kesinti zaman dilimleri ile belirlenir: MİB belirli bir zaman dilimi boyunca proseslere sırayla hizmet verir
- **zaman dilimi** (quantum/time slice): zamanlayıcı kesmesi
- Hazır kuyruğu=proses kuyruğu
 - Yeni gelen proses kuyruğun sonuna eklenir
 - Çalışan proses zaman dilimi içinde sonlanmazsa hazır kuyruğunun sonuna eklenir
 - Sıralayıcı kuyruğun başında yer alan prosesi seçer ve MİB'ni verir
- etkileşimli prosesler için etkin bir yöntem
- etkin bağlam değiştirme ile gecikmeler azaltılabilir

Dönüşümlü Sıralama (q=1)



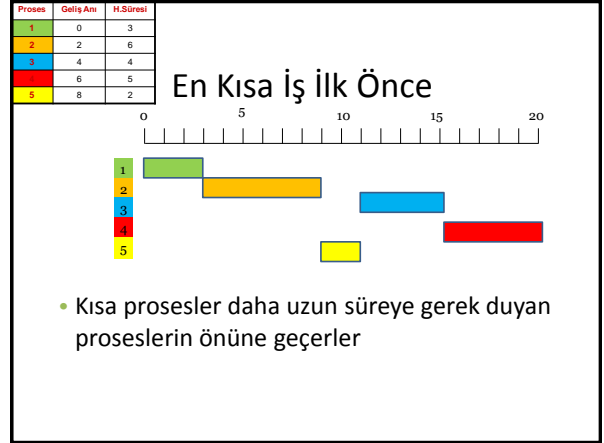
Dönüşümlü Sıralama

– zaman dilimi boyu seçimi

- süre sistemin başarımında etkili
 - küçük x büyük
 - sabit x değişken
 - her kullanıcıya aynı x ayrı
- çok büyük zaman dilimi ⇒ FCFS gibi olur
- çok küçük zaman dilimi ⇒ bağlam değiştirmede çok zaman kaybı olur
- zaman dilimi boyu seçimi sistemden sisteme değişir
- farklı iş yükleri de zaman dilimi süresi belirlemede etkili
 - ortalama etkileşim süresi kadar olmalı
 - genelde saniyenin kesiri mertebelerinde

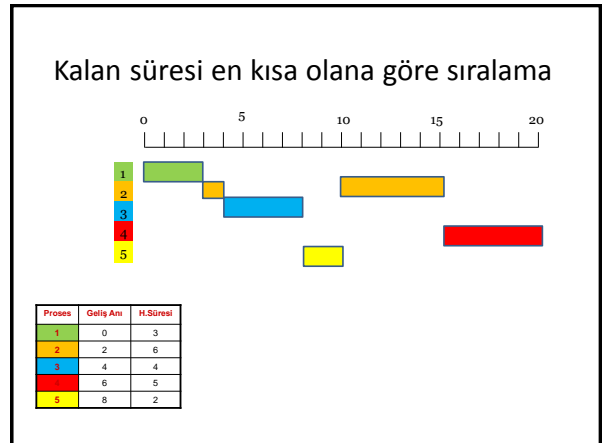
En Kısa İş İlk Önce (shortest job first)

- En kısa iş ilk önce (shortest job first)
 - kesintisiz
 - sonlanmak için en kısa süreye ihtiyaç duyan önce çalıştırılır
 - kısa işlere iyi hizmet verilir
 - etkileşimli ortamlara uygun değil
 - problem: prosese ne kadar süre MİB gerektiği önceden bilinmeli
 - kullanıcı tahmini süre verir
 - verilenden çok uzun sürerse kesilir ve sonra çalıştırılır
 - üretim ortamlarında daha önceden çalışmış prosesin süresi bilinebilir
 - ancak, geliştirme ortamlarında proseslerin çalışma süreleri çoğunlukla bilinemez



Kalan süresi en kısa olana göre sıralama (shortest time remaining)

- En kısa iş ilk önce sıralama yöntemin **kesintili** hali
 - zaman paylaşımlı sistemlerde iyi başarımlı
- yeni gelen prosesler de göz önüne alınarak, sonlanmasına en az zaman kalan proses önce çalıştırılır
 - çalışan proses yeni gelen kısa bir proses yüzünden kesintiye uğrayabilir
- sonlanma süresi önceden bilinmeli
- zaman kaybı fazla
 - kullanılan ve kalan süre hesabı için geçen süre
 - kesintide bağlam değiştirme için geçen süre



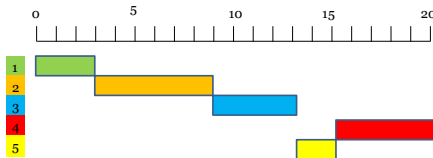
Sonlanma zamanına göre sıralama (Deadline)

- prosesler sonlanmaları gereken zamanlara göre sıralanır
 - proses zamanında biterse değeri var
- prosesin tüm kaynak istekleri önceden belirtilmeli
- sonlanma zamanları göz önüne alınarak kaynak dağıtımı planlanır
- sonlanma zamanı kısıtlı olanlarla ilgilenirken diğerleri aksatılmamalı

Yanıt süresi oranı en yüksek olana göre sıralama (highest response ratio next)

- “en kısa iş önce” yöntemi kısa işlere iyi yanıt verir: bu durumu düzeltme amaçlı bir yöntem
 - kesintisiz
 - prosesler sahip oldukları öncelik değerine göre seçilirler
 - Prosesin öncelik değeri şunlara bağlıdır:
 - prosesin gerek duyduğu hizmet süresi
 - hizmet almak için beklediği süre
- öncelik = (bekleme s. + hizmet s.) / (hizmet s.)**
- uzun süredir bekleyene yüksek öncelik,
 - kısa işe yüksek öncelik verir

Yanıt süresi oranı en yüksek olana göre sıralama



Öncelik hesabı:

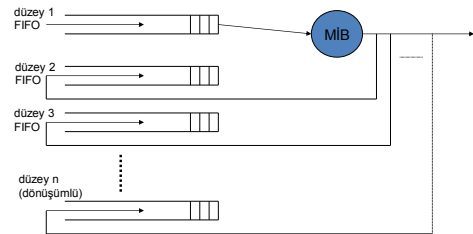
$t=9$
 $P3 \rightarrow 5+4/4=2.25$
 $P4 \rightarrow 3+5/5=1.6$
 $P5 \rightarrow 1+2/2=1.5$

$t=13$
 $P4 \rightarrow 7+5/5=2.4$
 $P5 \rightarrow 5+2/2=3.5$

Proses	Geliş Anı	H.Süresi
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

Çok düzeyli kuyruklar

- Çok düzeyli kuyruklar (her kuyruk farklı bir öncelik düzeyi)



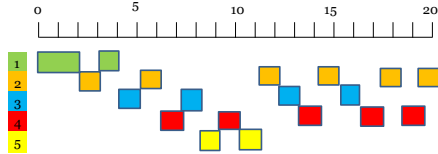
Çok düzeyli kuyruklar

- yeni gelen proses düzey 1 kuyruğunun sonuna eklenir
- düzey içi FIFO kuralı uygulanır
- proses zaman dilimi içinde sonlanmazsa, bir alt düzey kuyruğun sonuna eklenir
- düzey sayısı sınırlıdır
- son düzeyde FIFO yerine dönüşümlü sıralama uygulanır
- yeni ve kısa işler çabuk sonlanır
- bazı sistemlerde alt kuyruklarda prosese tanınan zaman dilimi artar
 - uzun süre beklemiş, karşılığında daha fazla çalışır

Çok düzeyli kuyruklar

- bir kuyruktaki proses üst kuyruklar boşalmadan çalışamaz
- çalışan proses üst kuyruğa gelen bir proses tarafından kesilebilir
- askıya alınan prosesin hangi kuyrukta olduğu bilgisi tutulur
- işlemci zamanı dolmadan kesilen proses aynı kuyruğun sonuna eklenerek iyi hizmet verilir
- adaptif bir yapı
- bazı sistemlerde proses alt kuyruğa düşmeden aynı kuyrukta bir kaç kere döner
 - alt kuyruklarda daha fazla dönebilir

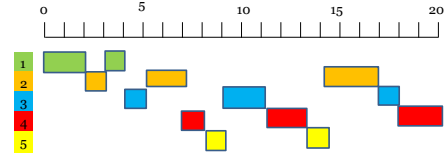
Çok düzeyli kuyruklar ($q=1$)



Varsayım: Sistemde en fazla 3 düzey var!

Proses	Geliş Anı	H.Süresi
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

Çok düzeyli kuyruklar ($q=2^i$)



Farklı düzeylerde farklı zaman dilimleri:

Düzey 1:kuyruk 0 $\rightarrow q=2^0$
 Düzey 2:kuyruk 1 $\rightarrow q=2^1$
 Düzey 3:kuyruk 2 $\rightarrow q=2^2$

Proses	Geliş Anı	H.Süresi
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2