

# İşletim Sistemleri Uygulama 8

## Linux İş Sıralayıcısı

Bilgisayar Mühendisliği

İstanbul Teknik Üniversitesi  
34469 Maslak, İstanbul

April 11, 2011



**Bugün**

## **İşletim Sistemleri Uygulama 8**

İş Sıralama Esasları

Linux İş Sıralayıcıları

CFS iş sıralayıcısı

Diğer İşletim Sistemleri - İş Sıralayıcılar



# Çok işli çalışma

Çok işli çalışma algoritmaları süreçlerin nasıl kesildiğine göre ikiye ayrılır.

- ▶ Cooperative multitasking: Bir süreç işlemciyi ne zaman bırakacağına kendisi karar verir.
- ▶ Preemptive multitasking: Her işin işlemciyi meşgul edebileceği süre bir (alt ve) üst sınırla belirlenir. Süreçler işlemcide kalınacak süre hakkında karar veremezler.



# Timeslice - Quantum

Bir süreç scheduler tarafından duraksatılıncaya kadar geçecek en uzun süreye timeslice denir.

- ▶ Timeslice kısa olursa hem bağlam değişimiyle zaman yitilir hem de ön bellek taze kalmaz.
- ▶ Timeslice uzun olursa hem bağlam değişimi süreleri çalışma zamanına eklenir hem de ön bellek taze kalır.



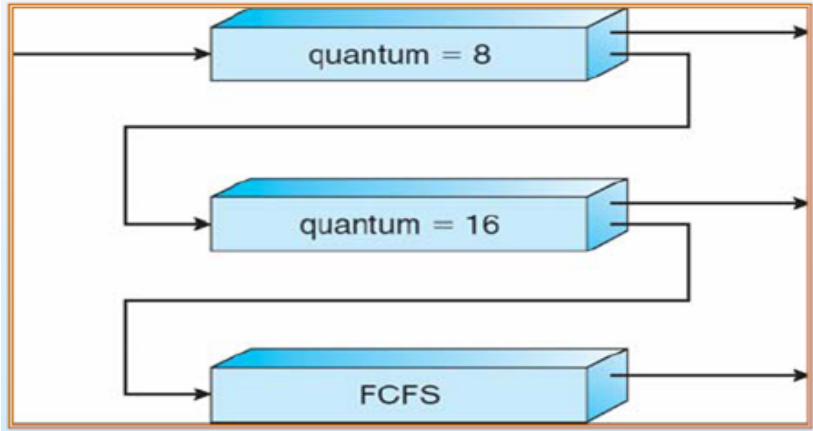
# Öncelik

- Bir sürecin önceliği iki parametreden biriyle belirlenir: nice veya RTPRIO
- nice,  $[-20, 19]$  aralığında, ne kadar düşükse süreç o kadar önceliklidir.
- RTPRIO,  $[0, 99]$  aralığında, ne kadar yüksekse süreç o kadar önceliklidir.
- Gerçek zamanda çalışan süreçler diğer süreçlerden önceliklidir.



# MLFQ - Multi Level Feedback Queues - Çok Düzeyli KuyruklarÇDK

Süreçler önceliklerine göre kuyruklarda tutulurlar. Sırası geldiğinde işini bitiremeyen süreç bir alt seviyedeki kuyruğun sonuna eklenir.



$O(1)$ 

- ▶ Sürecin o anki durumuna göre hesaplanan öncelikler sadece biri etkin olan iki listede tutulur.
- ▶ Etkin listede bekleyen süreç kalmayınca diğer listeye yeri değiştirilir.
- ▶ Ayrılan sürede işini bitiremeyen sürecin önceliği güncellenerek boştaki diğer listeye eklenir.
- ▶ Aynı öncelikte birden çok süreç varsa “round robin” algoritması kullanılır.



# CFS - Completely Fair Scheduler

- ▶ Her sürece sabit bir zaman (timeslice) yerine bir oran (proportion) atanır.
- ▶ Her sürecin en az 1 milisaniye çalışması için bir alt sınır belirlenmiştir.
- ▶ Aynı öncelikteki iki süreç aynı işlemci oranını elde eder.
- ▶ Öncelik grupları vardır. Örneğin, gerçek zamanda çalışan süreçler için ayrı bir scheduler kullanılabilir.
- ▶ Süreç seçmek  $O(1)$ , işlemciyi bırakan süreci zamanlamak  $O(\log(n))$  karmaşıklıkta.





# İş Sıralama Sınıfları

- ▶ CFS iş sıralayıcının daha genişleyebilir olması için iş sıralama sınıfları tanımlanmıştır.
- ▶ İş sıralama sınıfları kendilerine özel çalışma kuyrukları tutarlar ve iş sıralayıcının farklı gruplara farklı prensiplerle etkimesine olanak verirler.
- ▶ İş sıralama sınıfları, iş sıralama grupları ile daha etkin biçimde kullanılabilir. (Ar.: İş sıralama grupları)
- ▶ İş sıralama sınıfları `sched_class` isimli çekirdek veri yapısı kullanılarak gerçekleştirilir. Gerçekleştirilen gruplar olay tabanlı çalışan bir takım fonksiyonu programcının hizmetine sunar:
  - ▶ `enqueue_task()`: Bir görev çalışabilir hale geldiğinde çağırılır.
  - ▶ `dequeue_task()`: Bir görev çalışabilir halden çıktığında çağırılır.
  - ▶ `yield_task()`: Bir görev iş sıralamadan çıkıp diğer görev iş sıralamaya sokulmadan çağırılır
  - ▶ `check_preempt_curr()`: Çalışabilir hale gelen görev işlemciyi meşgul eden görevin yerini alacak mı?
  - ▶ `pick_next_task()`: Çalışmasını bitiren görevden sonra sıradaki görevi seçmek için çağırılır.



# İş Sıralama Prensipleri

- ▶ Bir süreç karakteristik olarak aşağıdaki iki profilden birine dahil edilebilir
  - ▶ G/Ç bağımlı
  - ▶ İşlemci bağımlı
- ▶ Sistemde koşulacak süreçlerin genel karakteristiği göz önüne alınarak aşağıdaki sıralama prensipleri, sıralama sınıflarına atanmıştır:
  - ▶ SCHED\_NORMAL(POSIX:SCHED\_OTHER) : CFS'in uygulandığı iş sıralama sınıfı. `kernel/sched_fair.c`'de tanımlanmıştır.
  - ▶ SCHED\_BATCH: CFS'in daha az süreç değiştiren varyantının kullanıldığı prensip. Cep belleklerden daha iyi faydalanmayı sağlar.
  - ▶ SCHED\_FIFO/\_RR: Gerçek zaman süreçler için kullanılan iş sıralama algoritmasının kullanılmasını sağlar. `sched_rt.c`'de tanımlanır.



## CFS iş sıralama

Linux öncelikle iş sıralama sınıfları ve prensiplerini kullanarak iş sıralama yapar. Böylece en yüksek öncelikli iş sıralama sınıfları öncelikli olarak kendileri için çekirdekte belirtilen prensibi(algoritmayı) kullanarak çalışır. Öte yandan kullanıcı süreçlerini sıralamak için CFS kullanılır:

- ▶ CFSde öncelikli olarak ele alınan problem etkileşimli uygulamaların da işlemci bağımlı uygulamaları etkilemeden doğru zamanda işlemciye erişmesine izin vermesine olanak vermektir.
- ▶ Bu nedenle süreçlere zaman dilimi vermek yerine, işlemci kullanım oranı atanır ve her sürecin kendi oranına ulaşmasını sağlayacak kesintiler sağlanır.
- ▶ Örneğin sistemde eşit öncelikli iki süreç varsa her birine %50 kullanım oranı atanır ve işlemciyi isteyen süreç bu oranı doldurma miktarına göre işlemciye atanır.
- ▶ Daha heterojen öncelikler için örneğin bir sürecin işlemci zamanının %70ini diğerinin %30unu kullanmasına olanak verilir.
- ▶ Çalışabilir süreçleri sıralamak için r/b ağaçlar kullanılır. Karar verme  $O(1)$  süreç ekleme  $O(\log(N))$  karmaşıklığındadır.
- ▶ CFS süreçlerin çalışma zamanı yerine sürecin işlemci kullanımının toplam kullanıma oranını hesaba kattığı için bekleme süreleri öne çıkartılır. Öncelik hesabıyla da en uzun süredir bekleyen sürece işlemci verilir.



# İş sıralama veri yapısı ve vruntime

- ▶ Her ne kadar CFS süreçlerin işlemciyi meşgul etme zamanlarından ziyade meşgul etme oranlarına bağlı çalışsa da süreç çalışma zamanlarını bir veri yapısında tutmaya ihtiyaç duyar. Bu veri yapısı `sched_entity` olarak adlandırılır ve `<linux/sched.h>` içinde tanımlanır.
- ▶ Bu yapıda `vruntime` isimli değişken önemlidir. Bu değişken sürecin *sanal çalışma zamanını* tutar. Bu değer sürecin gerçek çalışma zamanının o anda sistemde işlemciyi bekleyen süreçlere göre normalize edilmiştir.
- ▶ CFS `vruntime`'ı en düşük süreci seçmeye çalışır.



## EDF - Earliest Deadline First Scheduling

- ▶ EDF(Earliest Deadline First Scheduling) iş sıralama adından da anlaşılacağı üzere sonlanma zamanına en çok yaklaşan sürece işlemciyi sunmak üzerine kuruludur. Gelecekte Linux'e de adapte edilmesi düşünülmektedir(SCHED\_DEADLINE).
- ▶ Kesintili iş sıralama yapan tek işlemcili sistemler için optimal olduğu kanıtlanmıştır.
- ▶ Her süreç için sonlanma zamanı belirlemenin mümkün olmaması dezavantajlarındandır.



## Diğer İşletim Sistemleri

Birçok işletim sistemi ve kullandıkları iş sıralayıcılar aşağıdaki gibidir

Windows 3.1x	Yok	kesintisiz
Windows 95, 98, Me	Yarı	32-bit kesintili, 16-bit kesintisiz
Windows NT	Var	ÇDK
Mac OS 9 öncesi	Yok	kesintisiz
Mac OS 9	Yarı	yönetim görevleri kesintili, Süreç ve iplikler kesintisiz
Mac OS X	Var	ÇDK
Linux 2.6 öncesi	Var	ÇDK
Linux 2.6-2.6.23	Var	$O(1)$
Linux 2.6.23 sonrası	Var	CFS
Solaris	Var	ÇDK
NetBSD	Var	ÇDK
FreeBSD	Var	ÇDK

