Bilgisayar İşletim Sistemleri

Linux İş Sıralayıcı

İstanbul Teknikl Üniversitesi
34469 Maslak, İstanbul

26 Nisan 2017

Bugün

Bilgisayar İşletim Sistemleri

İş Sıralama Esasları

Linux İş Sıralayıcı

Çok işli çalışma (Multitasking)

Multitasking algoritmaları süreçlerin nasıl kesildiğine göre ikiye ayrılır:

- Cooperative multitasking (a.k.a. non-preemptive multitasking): Bir proses işlemciyi ne zaman bırakacağına kendisi karar verir. Windows 95 ve Windows NT den önce 16 bit işlemler için Microsoft Windows da kullanılmıştır. Ayrıca OS X den önce Mac OS da kullanılmıştır. Halen RISC işletim sistemlerinde kullanılmaktadır.
- Preemptive multitasking: Her işin işlemciyi meşgul edebileceği süre, alt ve üst sınırlarla belirlidir. Prosesler işlemcide kalacakları süre ile ilgili kararı kendileri veremezler.

Timeslice - Quantum

Bir proses scheduler tarafından duraksatılıncaya kadar geçen en uzun süreye timeslice (ya da quantum) denir.

- Timeslice çok küçük olursa bağlam değiştirme (context switching) ile çok zaman kaybedilir.
- Timeslice çok uzun olursa prosesler işlemciyi elde edebilmek için çok beklerler. (Poor concurrency)

Öncelik

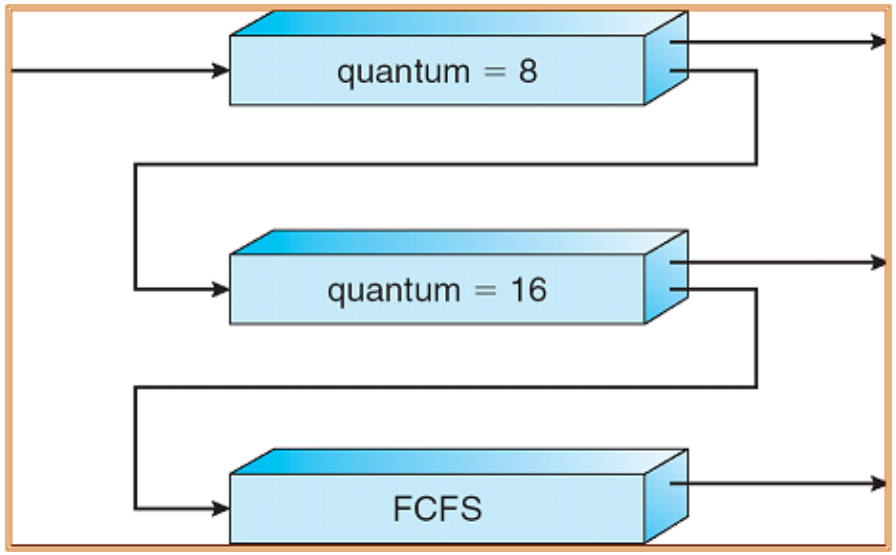
- Bir prosesin önceliği iki parametreden biriyle belirlenir: nice and RTPRIO
- nice: [-20, 19] aralığında, ne kadar düşükse proses o kadar önceliklidir.
- RTPRIO (the realtime or idle priority): [0, 31] aralığında, 0 değerine sahip proses en yüksek önceliğe sahiptir.
- Gerçek zamanlı prosesler diğer proseslere göre daha önceliklidir.

MLFQ - Multi-Level Feedback Queues- Çok düzeyli kuyruklar (Linux 2.5 öncesi)

Prosesler önceliklerine göre kuyruklarda tutulurlar (kısa prosesler ve G/Ç bağımlı prosesler daha önceliklidir).

Eğer bir proses verilen quantum süresinde işini bitiremezse daha düşük seviyedeki bir kuyruğun sonuna eklenir.

Üçüncü kuyruk FCFS (First-come, first-served).



O(1) İş sıralayıcı (Linux 2.5-2.6.23)

- O(1) iş sıralayıcı (`O(1)`) karmaşıklığında çalışır, dolayısıyla görev sayısı arttıkça da iyi bir performans gösterir.
- İki öncelik dizisi tutulur: *active* ve *expired*.
- İlk başta tüm görevler *active* öncelik dizisindedir.
- Belirlenen timeslice da işini bitiremeyen proses kesilir ve *expired* öncelik dizisine taşınır.
- *active* dizisinde bekleyen görev kalmayınca, *expired* dizisiyle yer değiştirilir.
- Aynı önceliğe sahip birden çok proses varsa "round robin" yaklaşımı kullanılır.

CFS - Completely Fair Scheduler (Linux 2.6.23 after)

- Her prosese sabit bir zaman (timeslice) yerine bir oran (proportion) atanır.
- Aynı öncelikteki iki proses aynı işlemci oranı elde eder.
- Öncelik grupları vardır. Örneğin, gerçek zamanda çalışan süreçler için ayrı bir scheduler kullanılabilir.
- Proses seçmek O(1), iş sıralama O(log(n)) karmaşıklığındadır.

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 `sched_class` isimli çekirdek veri yapısı kullanılarak gerçekleştirilir. Gerçekleştirilen gruplar olay tabanlı çalışan bir takım fonksiyonu programcının hizmetine sunar:
  - `enqueue_task()`: Bir görev(task) çalışabilir durumuna geçtiğinde çağrılır.
  - `dequeue_task()`: Bir görev çalışabilir durumdan çıktığında çağrılır.
  - `yield_task()`: Bir görev iş sıralamadan çıkıp diğer görev iş sıralamaya sokulmadan çağrılır.
  - `check_preempt_curr()`: "Çalışabilir hale gelen görev işlemciyi meşgul eden görevin yerini alacak mı? Yani çalışan program kesilecek mi?" kontrollerini yapar.
  - `pick_next_task()`: Çalışacak sıradaki görev seçilir.

İş Sıralama Prensipleri

- Bir proses karakteristik özelliklerine göre aşağıdaki iki profilden birine dahil edilebilir:
  - G/Ç bağımlı
  - İşlemci bağımlı
- Sistemde koşulacak proseslerin genel karakteristiği göz önüne alınarak aşağıdaki sıralama prensipleri, sıralama sınıflarına atanmıştır:
  - SCHED_NORMAL(POSIX:SCHED_OTHER)): Genel görevler için kullanılan sıralama prensibi.
  - SCHED_BATCH: CFS' nin daha az proses değiştiren varyantının kullandığı prensip. Cep belleklerden daha iyi faydalanmayı sağlar.
  - SCHED_FIFO/_RR: Gerçek zamanlı prosesler için kullanılan iş sıralama algoritmasının kullanılmasını sağlar.

İş Sıralama veri yapısı ve vruntime

- İş sıralamayla ilgili bilgiler `<linux/sched.h>` kütüphanesinde yer alan `sched_entity` veri yapısında tutulur.
- Bu yapıda `vruntime` isimli değişken önemlidir. Bu değişken prosesin sanal çalışma zamanını (virtual run time) tutar. Bu değer prosesin gerçek çalışma zamanının, o anda sistemde işlemciyi bekleyen proses sayısına göre normalize edilmiş halidir.
- CFS `vruntime` değeri en düşük prosesi seçmeye çalışır.

EDF - Earliest Deadline First Scheduling

- EDF(Earliest Deadline First Scheduling): iş sıralama yöntemi sonlanma zamanına en çok yaklaşan prosese işlemciyi sunmak üzere kuruludur.
- Kesintili iş sıralama yapan tek işlemcili sistemler için optimaldir.
- Her proses için sonlanma zamanı belirmenin mümkün olmaması dezavantajlarındandır.