

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Ders : İşletim Sistemleri
Görevli Öğretmen : Dr. Öğr. Üyesi ABDULLAH SEVİN
Dönem : 2023-2024 Güz Dönemi
Grup No : 20
Konu : Görevlendirici Kabuğu (Dispatcher Shell)
Grup Üyeleri : Selin ŞAHİN G211210087
Mustafa İkbāl AVCI B201210070
Fuad GARİBLİ G201210558
Başak MASKAR G211210093
Mustafa KURT B201210052

Github :
<https://github.com/ikbalavci/OperatingSystem2023Guz>

TASARIMIN OLUŞTURULMASI

İlk öncelikle işe ödevi anlamakla başladık. Derste edindiğimiz bilgilerin üstüne internet üzerinden ve çeşitli kaynaklardan, proje için bizden istenen özelliklere göre araştırmalar yaparak bilgi düzeyimizi arttırmaya çalıştık. Bu araştırmalar doğrultusunda öğrendiğimiz bilgilerle projeyi karşılaştırarak en doğru tasarımı bulmak için çaba sarfettik. Edindiğimiz bilgileri de bu raporda topladık.

İLK GELEN İLK ÇALIŞIR (FCFS) ALGORİTMASI

Bu algoritmaya göre bir kaynak veya bir sıraya ilk gelen prosesin ilk önce işini bitirerek çıkması beklenir. Uygulanması ve kullanılması kolay bir algoritmadır.

FCFS algoritmasında bir işlem eğer işlemci tarafından kabul edildiyse bu o işlem bitene kadar başka bir işlem araya giremez demektir.

Performansı düşüktür ve genel bekleme süresi de oldukça yüksektir. Bu yüzden gerçek zamanlı sistemler için veya yüksek öncelikli sistemler için her zaman iyi bir seçenek değildir. Mevcut istek ile bir sonraki istek arasındaki mesafeyi hesaba katmayan, önleyici olmayan bir algoritmadır.

ÇEVİRİMSEL SIRALI (RR) ALGORİTMA

Round-Robin (RR) planlayıcısı, işletim sistemlerinde kullanılan bir zaman paylaşım algoritmasıdır. RR, işlerin veya iş parçacıklarının (threads) eşit ve adil bir şekilde zaman dilimlerine ayrılmasını sağlar. Her iş ya da iş parçacığı, sırayla belirli bir zaman dilimi (genellikle birkaç milisaniye kadar) boyunca CPU zamanına erişir.

Round-Robin algoritması aşağıdaki şekilde çalışır:

- **Zaman Dilimi Atama:** Planlayıcı, her işe eşit bir zaman dilimi atar (örneğin, 10 milisaniye).
- **İşlerin Yürütülmesi:** İlk iş, atanan zaman dilimi boyunca çalıştırılır. Eğer iş, zaman dilimi süresince tamamlanmazsa, sonraki işe geçilir.
- **Kuyruğa Ekleme:** Yürütme sırası sona eren ancak tamamlanmamış işler, kuyruğun sonuna eklenir.
- **Sonraki İş Seçme:** Kuyruğun başındaki sonraki iş, yürütülmek üzere CPU'ya gönderilir.
- **Tamamlanmış İşlerin Kaldırılması:** Bir iş tamamlandığında, kuyruktan çıkarılır.
- **Devam:** Bu süreç, tüm işler tamamlanana kadar devam eder.

AVANTAJLAR

- **Adillik:** Her iş eşit CPU zamanına sahip olduğundan, hiçbir iş diğerinden daha fazla zaman alamaz.
- **Öngörülebilirlik:** İşler, varış sırasına göre yürütüldüğünden, kullanıcılar ne zaman işlerinin tamamlanacağını daha iyi tahmin edebilir.

DEZAVANTAJLAR

- **İlgisiz Yükseklik:** RR, işlerin önceliklerini veya iş yükünü dikkate almaz. Bu, önemli veya acil işlerin daha az önemli işlerle aynı tedaviyi görmesine neden olabilir.
- **Düşük Verimlilik:** Küçük işler büyük işlerle aynı zaman dilimine sahip olduğundan, küçük işlerin hızla tamamlanmasını beklemek yerine, büyük işlerle aynı süreyi beklemek gerekebilir.
- Round-Robin, dairesel bir kuyruk kullanarak uygulanabilir, bu da işlerin kuyruğun sonuna döndüğü anlamına gelir. Dairesel kuyruk yapısı, işlerin yürütülme sırasının sürekli bir döngü içinde devam etmesine yardımcı olur.
- Bu basit ve adil planlama stratejisi, gerçek zamanlı sistemlerde, sunucu yük dengelemesinde ve diğer zaman hassas uygulamalarda yaygın olarak kullanılmaktadır.

GÖREVLENDİRİCİ(DISPATCHER)

Temel olarak görevi işletim sisteminin çekirdeğinin, yürütülmeyi bekleyen farklı işlemlere veya iş parçacıklarına göre işlemciyi tahsis etmekten sorumlu olan bir parçasıdır. Bu parçaların veya süreçlerin hangi zamanda hangi kontrole girmesi gerektiğini ayarlamaktan sorumludur.

Görevlendirici, bu işlemleri yapmak ve hangisinin önce veya sonra yapılması gerektiği gibi konularda farklı farklı algoritmaları kullanır. Bu algoritmalar farklı önceliklere göre tasarlanabilir yani sürecimizin kaynak kullanımı, varış süresi veya alanı gibi önceliklere göre belirlenebilirler.

Görevlendirici işletim sistemimizin en yararlı şekilde çalışmasında büyük rol oynamaktadır. Kaynakların verimini en yüksek seviyeye çıkarmak, önceliği belirlemek gibi önemli görevlerde karar mekanizması olarak iş görür.

PROGRAMIN GENEL YAPISI

Programımızda bir işletim sisteminde bulunması gereken komut satırı arayüzü (CLI) ve grafik kullanıcı arayüz kullanılmaktadır.

Komut Satırı Arayüzü (Command-Line-Interface/CLI):

- Bu arayüz sayesinde kullanıcı metin tabanlı komutlar yazarak işletim sistemimizde ve uygulamalarda etkileşimde bulunabilir.
- Kullanıcı, belirli komutlar sayesinde dosya yönetimini, uygulama çalıştırmasını, ağ ayarlarını değiştirme gibi istekleri gerçekleştirebilir.
- Kullanıcıya genellikle çok fazla esneklik sunar fakat bilmeyenler için kullanımı teknik açıdan da zorlayıcı olabilir.

Grafik Kullanıcı Arayüzü (Graphical User Interface-GUI):

- Kullanıcının görsel olarak etkileşim sağlamasına imkan tanır.

- Kullanıcının menü, pencere, düğme vb. öğelerle işletim sistemindeki işlemlerinde kolay işlem yapmasını sağlar
- Kullanıcıya anlamakta daha kolay bir mekanizma sunar ve hızı artırır.

İşletim sistemleri bu iki arayüzü birlikte sunarak kullanıcılara istediklerini seçebilecekleri bir ortam sunar ve bu ortam sayesinde kullanıcı daha çok verim alarak işlemlerini halletmeye devam edebilir.

TARTIŞMA

Linux ve Windows işletim sistemleri, temel olarak işlemlerin yürütülmesini planlamak ve CPU'yu etkili bir şekilde kullanmak için görevlendirici (scheduler) içerir. İşlemler, bir hazır işlemler kuyruğuna alındığında, görevlendirici bu kuyruktan bir işlem seçer ve CPU'yu bu işleme tahsis eder. İşlem, yürütme tamamlandığında veya kaynakları beklerken bloke olduğunda sona erer, ve ardından görevlendirici bir sonraki işlemi seçer ve bu işlemi CPU'ya tahsis eder.

Linux ve Windows çekirdekleri arasındaki önemli bir fark, zamanlama algoritmalarında ve zamanlama politikalarında ortaya çıkar. Linux, genellikle Tamamen Adil Zamanlayıcı (CFS) ve Round Robin Zamanlayıcı gibi çeşitli zamanlama algoritmalarını kullanırken, Windows, önceliğe dayalı zamanlama ve zaman dilimleme kombinasyonunu benimser.

Özellikle, Linux'un zamanlayıcısı tamamen önleyici (preemptive) bir yapıya sahiptir. Bu, çekirdeğin, bir işlemin CPU'yu işgal ettiği süre boyunca herhangi bir zamanda başka bir işlemi yürütmeye geçebileceği anlamına gelir. Diğer taraftan, Windows zamanlayıcısı kısmen önleyicidir; yani bir işlemi yalnızca belirli koşullar altında, örneğin sistem çağrısını engellemesi veya kendisine ayrılan zaman dilimini kullanması durumunda kesintiye uğratır.

Dispatcher sınıfı, bir simüle edilmiş işletim sisteminde işlemlerin yürütülmesini planlamak ve yönetmekten sorumlu bir yapıdır. Bu sınıf, işlemleri önceliklerine göre düzenler ve bir işlemin çok uzun süre bekletilmeden önce çalıştırılmasını sağlar. Başlatma yöntemi, sürekli bir döngü içinde çalışır ve yeni işlemleri kontrol eder. Aynı zamanda, 20 saniyelik bir bekleme süresi sınırını aşan işlemleri sonlandırır ve hazır işlemleri yürütür. Bu işlemleri yönetmek için processReader nesnesini, işlemleri sonlandırmak için terminateTimeout yöntemini ve hazır işlemleri yürütmek için processor nesnesini kullanır.

Dispatcher sınıfının QueueProcess yöntemi, bir işlemi önceliğine göre uygun kuyruğa yerleştirirken, getAppropriateProcess yöntemi kuyrukları belirli bir sırayla kontrol ederek bir sonraki çalıştırılacak işlemi alır. Bu şekilde, Dispatcher sınıfı, işlemlerin öncelik sırasına göre düzenlenmesini sağlar ve uzun süre bekleyen işlemlerin önlenmesini amaçlar. Bu, simüle edilmiş bir işletim sisteminin etkili bir şekilde çalışmasını sağlayarak kaynakları daha verimli bir şekilde kullanmaya yönelik bir strateji sunar.