İŞLETİM SİSTEMLERİ DERSİ PROJE ÖDEVİ

G191210555 MOHAMEDOU MOHAMED CHERİF

B201210575 MOHAMAD ALBITAR

G221210385 MELEK YAKIN

G221210301 HAMDİ ALANAY

G211210306 SAFA DÖNERGÖZ

Ödev Linki:

https://github.com/halanay/IsletimSistemleriOdevi

Giriş:

İşletim sistemleri ve görevlendirme yönetim sistemleri gibi modern bilgisayar sistemlerinin çeşitli bileşenleri vardır. Bu sistemler, kullanıcıların ihtiyaçlarına hızlı ve etkili bir şekilde cevap verirken kaynakları verimli bir şekilde kullanmayı amaçlar. İşte bu bağlamda, görevlendirici (dispatcher) kabuğu gibi işletim sistemlerinin önemli bir bileşenine odaklanan projeler mevcuttur. Bu projeler, çoklu programlama sistemlerindeki karmaşık işlemleri yönetmek ve optimize etmek için tasarlanmıştır. Bunun gibi projeler, işletim sistemlerinin görevlendirme algoritmalarını geliştirmek veya verimliliklerini artırmak için çalışabilir. Ayrıca işlemlerin önceliklerini düzenlemek, işlemci zamanını etkin kullanmak ve işlemcideki iş yükünü dengelemek gibi hedeflere de odaklanabilirler. Bu tür projeler, işletim sistemlerinin performansını iyileştirmek ve kullanıcı deneyimini optimize etmek için önemli bir rol oynar.

Projenin Amacı ve Kapsamı:

Projenin amacı, dört seviyeli öncelikli bir görevlendirici kabuğunu simüle etmektir. Bu görevlendirici, gelen işlemleri önceliklerine göre sıralayarak kısıtlı kaynakları etkili bir şekilde kullanmaya çalışır. Gerçek zamanlı ve kullanıcı prosesleri arasında ayrım yaparak, gerçek zamanlı proseslere öncelik verir ve bu prosesleri tamamlana kadar kesinti olmadan çalıştırır. Kullanıcı prosesleri ise üç seviyeli geri bildirimli bir görevlendirici tarafından işlenir, bu da sistemdeki kaynakların daha etkili bir şekilde kullanılmasına olanak tanır. Proje kapsamında, işletim sistemleri dersinde öğrenilen temel kavramların pratiğe dökülmesi hedeflenir. Bellek yönetimi, kaynak tahsisi algoritmaları, farklı öncelik seviyelerindeki proseslerin yönetimi gibi konular, projenin ele aldığı temel başlıklardan bazılarıdır. Ayrıca, gerçek zamanlı proseslerin öncelikli olarak ele alınması ve bu proseslerin düşük öncelikli prosesleri etkilemeden çalışmalarının devam etmesi, projenin özgünlüğünü ve önemini artırır.

İşletim Sistemleri Dersi ile Bağlantı:

Öğrencilerin temel bilgisayar sistemlerini anlamalarını ve bu bileşenleri etkili bir şekilde yönetme becerilerini kazanmalarını amaçlayan işletim sistemleri dersi, projenin temelini oluşturur. Ders kapsamında öğrenilen kavramlar, proje üzerinde uygulama alanı bulur. Öğrenciler, bellek yönetimi, görevlendirme stratejileri ve kaynak yönetimi gibi konularda edindikleri bilgileri kullanarak karmaşık bir çoklu programlama sistemini simüle eden bir görevlendirici kabuğu tasarlarlar. Bu proje, öğrencilere işletim

sistemlerinin pratik uygulama yönünü göstererek teorik bilgilerini pekiştirmelerini sağlar. Öğrenciler, gerçek dünya senaryolarına benzer durumları ele alarak görevlendirme stratejilerini optimize etme ve kaynak yönetimi problemlerini çözme becerilerini geliştirirler. Ayrıca, projenin karmaşıklığı, öğrencilere sistemlerin performansını artırma ve kullanıcı deneyimini iyileştirme konularında yeni yaklaşımlar geliştirme fırsatı sunar.

Proje Önem ve Güncellik:

Yönetme ihtiyacı artmaktadır. Bu bağlamda, çoklu programlama sistemleri üzerine yapılan bu tür projeler, öğrencilere teorik bilgilerini pratiğe dökmeleri için önemli fırsatlar sunar ve gerçek dünya uygulamalarına dair değerli deneyimler kazandırır. Ayrıca, bu projenin gerçek zamanlı proseslere odaklanması, endüstriyel uygulamalarda sıklıkla karşılaşılan bir ihtiyaca yönelik bir çözüm sunar.

Bellek Tahsis Algoritmaları:

Bir işletim sisteminde, bellek tahsisi işlemi sistemin performansı açısından son derece kritik bir öneme sahiptir. Bu bölümde, projemizde kullanılan bellek tahsis algoritmalarının seçimini, gerekçelerini ve alternatiflerini tartışacağız. Bellek tahsis algoritmaları, kullanılabilir bellek kaynaklarının etkili bir şekilde yönetilmesini sağlar. Projemizde kullanılan bellek tahsis algoritmalarını seçerken birkaç faktörü göz önünde bulundurduk. Öncelikle, bellek tahsisindeki verimlilik önemli bir faktördür. Algoritmaların bellek kullanımını optimize etmesi ve boş alanları etkin bir şekilde kullanması beklenir. Ayrıca, bellek tahsisinde hızlı erişim ve işlem süreleri sunan algoritmalar tercih edilir. Bu, sistemdeki işlemlerin hızlı ve kesintisiz çalışmasını sağlar. Projemizde kullanılan bellek tahsis algoritmaları arasında "İlk Uygun" (First Fit), "En İyi Uygun" (Best Fit) ve "En Kötü Uygun" (Worst Fit) gibi yaygın olarak kullanılan algoritmalar yer almaktadır. İlk Uygun algoritması, bellek bloklarını sırayla kontrol ederek ilk uygun boş alanı seçer. En İyi Uygun algoritması ise, talep edilen bellek miktarına en yakın uygun boş alanı seçer. En Kötü Uygun algoritması ise, talep edilen bellek miktarını karşılayan en büyük boş alanı seçer.

Alternatif olarak, "Birleştirme ve Bölme" (Compaction and Splitting) gibi dinamik bellek yönetimi teknikleri de kullanılabilir. Bu tekniklerde, sürekli olarak birleştirme işlemi yaparak boş alanları birleştirme veya büyük bir boş alanı parçalara bölmek gibi

yöntemler kullanılır. Bu sayede bellek kullanımı daha verimli hale getirilebilir. Bellek tahsis algoritması seçiminde, sistem gereksinimleri, bellek kullanım örüntüleri ve performans hedefleri gibi faktörler göz önünde bulundurulmalıdır. Her algoritmanın avantajları, dezavantajları ve uygunluk durumları farklı olabilir.

Kullanılan Bellek Tahsis Algoritmalarının Seçimi ve Gerekçeleri

Projemizde, bellek tahsisinde iki temel algoritma kullanılmıştır: İlk Uyumlu İlk Hizmet (First Fit) ve EnUyumlu Hizmet (Best Fit).

- İlk Uyumlu İlk Hizmet (First Fit): Bu algoritma, bellekteki ilk uygun boş alanı kullanır. Prosesin bellekteki ilk uygun yere yerleştirilmesi, hızlı bir şekilde işlem yapabilmesini sağlar. Ancak, bu algoritma zamanla bellekte boş kalan küçük blokların birikmesine neden olabilir ve fragmentasyona yol açabilir. Projemizde bu algoritmanın tercih edilme nedeni, basit bir yapıya sahip olması ve hızlı bir şekilde işlem yapabilmesidir.
- En Uyumlu Hizmet (Best Fit): Bu algoritma, prosesin bellekteki en küçük uygun boş alanı kullanmasını sağlar. Bu şekilde, bellekteki boş alanların daha iyi kullanılması ve fragmentasyonun azaltılması amaçlanır. Ancak, bu algoritmanın dezavantajı, işlem süresince uygun blokları aramanın daha zaman alıcı olmasıdır. Projemizde en uygun hizmet algoritmasını seçme nedenimiz, bellek kullanımını daha verimli hale getirmek ve fragmentasyonu en aza indirmek istememizdir.

Alternatif Algoritmaların Karşılaştırılması:

Projemizde kullanılan algoritmaların yanı sıra, bellek tahsisinde kullanılabilecek diğer önemlialgoritmalar şunlardır:

- En Kötü Durumda Uyumlu Hizmet (Worst Fit): Bu algoritma, prosesin bellekteki en büyükuygun bloğa yerleştirilmesini sağlar. Bu şekilde büyük bloklar hızla kullanılabilir, ancak fragmentasyon sorunlarını artırabilir.
- Döngüsel Tarama (Circular Scan): Bu algoritma, bellekte döngüsel bir tarama yapar ve uygun bloğu bulana kadar devam eder. Bu şekilde fragmentasyonu kontrol altında tutmak mümkünolabilir, ancak arama süresi daha uzun olabilir.

Kaynak Yönetimi:

Kaynak yönetimi, çoklu programlama sistemlerindeki görevlendirici kabuğunun temel bir

bileşenidir.

Proje kapsamında kullanılan kaynak yönetimi stratejileri ve bu stratejilerin uygulanma mantığıaşağıda ele alınmaktadır:

1. İlk Gelen İlk Hizmet (First-Come, First-Served - FCFS): Bu strateji, gelen görevlerin sırayla işlemciye erişim hakkı kazanmasını sağlar. Gelen görevlerin sırasına göre işlemciye atanmasıyla kaynaklar yönetilir. İşlemci, mevcut görevi tamamladıktan sonra bir sonraki görevi işlemeye alır. Bu strateji basittir ancak adil olmayabilir, çünkü öncelikli görevler diğerlerini bekletebilir.

B. Görevlendirici Yapısı ve Modülleri:

Projemizin temel bileşeni olan görevlendirici modülü, işletim sistemi görevini üstlenir. Proseslerin varış zamanlarına göre sıralanmış kuyruklardan işlem yapar ve uygun sıralayıcılara gönderir.

- FCFS Yüksek Öncelikli Sıralayıcı (FCFS.java)

İlk Gelen İlk Çalışır (FCFS) algoritması temelinde çalışan yüksek öncelikli sıralayıcı modülüdür. Gerçek zamanlı prosesleri, gelen sıraya göre işler ve diğer düşük öncelikli proseslerden öncelikli olarak çalıştırır.

- Kullanıcı Geri Beslemeli Sıralayıcı (GBG.java)

Üç seviyeli geri beslemeli sıralayıcı modülüdür. Temel zamanlama kuantumu 1 saniyedir ve geri besleme sıralayıcısı bu kuantum üzerinden işlem yapar. Prosesleri öncelik düzeyine göre sıralar veuygun kuyruğa yerleştirir.

- Round Robin Sıralayıcı (RR.java)

Kullanıcı geri beslemeli sıralayıcı içinde kullanılan round robin modülüdür. Eğer tüm prosesler en alt seviye kuyrukta ise, basit bir çevrimsel sıralı (round robin) algoritma çalıştırır. Belirli bir zamandiliminde her prosese eşit süre tahsis eder.

-Mantık Akışı Modülü

Görevlendirici içindeki proseslerin işleyişini adım adım açıklayan bir modüldür. Proseslerin varışından başlayarak, işlemin askıya alınması, devam etmesi veya sona ermesi durumlarını yönetir.

Proje Çıktısı ve Sonuçları:

Proje, çoklu programlama sistemlerinde kullanılan temel kavramları ve işleyişi simüle ederek bir program sunar. Gerçek zamanlı ve kullanıcı proseslerinin etkileşimli bir şekilde

yönetilmesini sağlar.

Görevlendirici Modülleri:

Proje, farklı öncelik düzeylerindeki prosesleri işleyen ve yöneten çeşitli modülleri içerir. FCFS yüksek öncelikli sıralayıcı, kullanıcı geri beslemeli sıralayıcı ve round robin sıralayıcı gibi modüllerin işleyişi simüle edilir.

Bellek ve Kaynak Yönetimi:

Proje, bellek tahsis algoritmalarını kullanarak proseslerin bellek ihtiyaçlarını karşılamayı hedefler. Ayrıca, kaynak yönetimi stratejileri kullanarak giriş/çıkış kaynaklarını etkili bir şekilde tahsis eder veprosesler arasında paylaşımı sağlar.