SAKARYA
ÜNİVERSİTESİ

İŞLETİM SİSTEMLERİ DERSİ PROJE ÖDEVİ

Hazırlayanlar:

2B – G211210090 SELİM ŞENER

2B – G211210072 BEYZA NUR ÖZDEMİR

2B – G201210054 ATAKAN PAŞALI

2B – G211210060 RÜMEYSA GÜNAY

1B – B201210048 MUSTAFA BAYRAKTAR

❖ **Projenin GitHub Linki:** https://github.com/selimsener/grup39

❖ **Hangi bellek tahsis algoritmalarını kullanabileceğinizi açıklayın ve tartışın ve son tasarım seçiminizi gerekçelendirin.**

- Bellek ayırma algoritmaları seçiminin tanımı, tartışılması ve gerekçesi : Veri dosyasındaki bazı proseslerin kullandığı bellek miktarı sistemimizin bellek miktarından büyük olduğu için ve best fit gereken bellek miktarının fazla olduğu yapılarda daha iyi performans gösterdiği için seçildi.

❖ **Görevlendirici tarafından bellek ve diğer kaynakları kuyruğa almak, göndermek ve tahsis etmek için kullanılan yapıları tanımlayın ve açıklayın.**

- Gerçek zamanlı prosesler için FCFS yapısını kullanır. Gerçek zamanlı prosesler daha önceliklidir ve ilk olarak öncelikli prosesler işleme alınır. Daha sonra sırasıyla öncelik sıralarına göre 3 Seviyeli geri beslemeli kuyruk yapısı kullanılır. Son kuyruk için özel bir Round robin algoritması yazılmıştır.

❖ **Çeşitli modülleri ve ana işlevleri açıklayarak programınızın genel yapısını tanımlayın ve gerekçelendirin ('arayüzler' işlevinin açıklamaları beklenir).**

- Program çalıştırıldığında dosyadaki verileri tek tek alıp bir arraya ekler. Her bir oluşturulan arrayin ilk elemanı id olarak belirlenir. Oluşturulan bu arraylerin elemanları processin özelliklerini belirler ve process oluşturulur. Oluşturulan her bir process bir listeye atılır. Ana görevlendirici bu listedeki prosesleri varış zamanına göre ait oldukları kategoriye ekler. Gerçek zamanlı ise FCFS, kullanıcı processi ise firstQueue, secQueue ve Round Robin kategorilerine processin önceliğine göre ekler. Daha sonra bu eklenen kuyruklarda ilk öncelik gerçek zamanlılara ait olduğu için FCFSnin boş olup olmadığı kontrol edilip timer arttırılır. FCFS kuyruğu boşaldığı taktirde diğer kuyruklara geçilir. Eğer tüm kuyruklar boş ise program sonlandırılır.

❖ **"Gerçek" işletim sistemleri tarafından kullanılan şemalarla karşılaştırarak, böyle çok düzeyli bir görevlendirme şemasının neden kullanılacağını tartışın. Olası iyileştirmeler önererek, böyle bir şemadaki eksiklikleri ana hatlarıyla belirtin. Tartışmalarınıza bellek ve kaynak ayırma şemalarını dahil edin.**

- Çok düzeyli görevlendirme şemaları, gerçek işletim sistemlerinde kullanılan ve farklı görevlendirme stratejilerini içeren bir yaklaşımdır. Bu şemalar, genellikle işletim sistemindeki çoklu görevlendirme (multitasking) ve zaman paylaşımı (time-sharing) gibi temel işlevleri yerine getirir.

- Performans ve Kaynak Kullanımı:

  - Gerçek İşletim Sistemleri: Gerçek sistemler, çok düzeyli görevlendirme stratejileriyle işlemci kaynaklarını etkin bir şekilde yönetir. İşlemci zamanını en iyi şekilde kullanarak, farklı görevlere adil bir paylaşım sağlar.

  - Neden Kullanılır?: Çok düzeyli görevlendirme şemaları, farklı önceliklere sahip görevleri ayrı ayrı ele alarak, yüksek öncelikli görevlere öncelik verir. Böylece kritik işler hızlı bir şekilde işlenirken, düşük öncelikli işler de zamanla işlenmiş olur.

- Öncelik Yönetimi:

  - Gerçek İşletim Sistemleri: Gerçek sistemlerde, öncelikli görevler veya acil işler daha yüksek öncelikle işlenir.

  - Neden Kullanılır?: Çok düzeyli görevlendirme şemaları, farklı öncelik seviyelerine sahip işleri ayrı ayrı ele alarak, önceliklendirme ve öncelik yönetimi sağlar. Özellikle acil durumlar veya kritik işlerin öncelikle işlenmesi gerektiğinde bu şemalar etkilidir.

- Adil Görevlendirme:

  - Gerçek İşletim Sistemleri: Gerçek sistemler, işlemci zamanını adil bir şekilde paylaşır ve her görevin eşit şekilde işlenmesini sağlar.

  - Neden Kullanılır?: Çok düzeyli görevlendirme şemaları, düşük öncelikli görevlerin de zamanla işlenmesini sağlar. Böylece işlemci zamanının adil bir şekilde paylaşılmasına olanak tanır.

- Çeşitlilik ve Esneklik:
  - Gerçek İşletim Sistemleri: Gerçek sistemler, farklı görevlendirme stratejilerini bir araya getirerek esneklik sağlar.
  - Neden Kullanılır? Çok düzeyli görevlendirme şemaları, farklı görevlendirme stratejilerini birleştirerek, sistemdeki çeşitli iş yüklerini ve öncelikleri karşılayabilir. Örneğin, Round Robin, FCFS ve öncelikli kuyruklar gibi farklı algoritmaların bir arada kullanılması mümkündür.

- Gelişmiş İşlemci Yönetimi:
  - Gerçek İşletim Sistemleri: Gerçek sistemlerde, işlemci yönetimi daha kapsamlıdır ve işlemci zamanının verimli kullanımı sağlanır.
  - Neden Kullanılır?: Çok düzeyli görevlendirme şemaları, işlemci zamanının daha verimli kullanılmasını sağlar. Örneğin, zaman dilimleri (time-slice) belirlenerek işlemciden bekleyen görevlere eşit paylaşımlar sağlanabilir.