T.C. SAKARYA ÜNİVERSİTESİ BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

İŞLETİM SİSTEMLERİ

GÖREVLENDİRİCİ KABUĞU PROJE RAPORU

65.GRUP

1B B201210036 ÖMER FARUK ÖZEL

1B B191210093 ERKAM HARMAN

2A Y235012008 MUHAMMET FURKAN SARAÇ

2A b231210380 UBEYDULLAH YÜCE

2A G201210596 MAHMUD ALREFAAI

Görevlendirici Kabuğu Program Yapısı Raporu

Bu program, karmaşık bir çoklu programlama sistemini tanımlayan dört seviyeli öncelikli bir proses görevlendiricisini içermektedir. Program, gerçek zamanlı prosesler ve normal kullanıcı prosesleri için ayrı kuyruklar kullanarak, her bir prosesi öncelik düzeyine göre işler. Programda iki ana sınıf bulunmaktadır: "Process" ve "Dispatcher."

**Process Sınıfı:** Bu sınıf, görevlendiricideki bir işlemi temsil eder. Özellikleri şunlardır:

1. **id (Kimlik):** Prosesin benzersiz kimliği, her yeni proses oluşturulduğunda artar.
2. **arrivalTime (Varış Zamanı):** Prosesin görevlendiriciye varış zamanını belirtir.
3. **priority (Öncelik Düzeyi):** Prosesin öncelik seviyesini tanımlar.
4. **burstTime (İşlem Süresi):** Prosesin tamamlanması için gereken süreyi ifade eder.
5. **startingBurstTime (Başlangıç İşlem Süresi):** Prosesin ilk oluşturulduğu andaki orijinal işlem süresi.

Bu sınıf, özelliklere değer atama ve erişim metodları içerir, aynı zamanda işlem süresini değiştirme yeteneğine sahiptir.

**Dispatcher Sınıfı:** Bu sınıf, önceliğe dayalı zamanlamayı kullanan bir görevlendiriciyi temsil eder. Önemli özellikleri ve metodları şunlardır:

1. **Öncelik Düzeyleri:** **REAL\_TIME**, **PRIORITY\_1**, **PRIORITY\_2**, ve **PRIORITY\_3**.
2. **run() Metodu:** Öncelik sıralarındaki işlemleri belirli bir sırayla çalıştırır; FCFS veya RR algoritmalarına göre işlemleri çalıştırır.
3. **Round-Robin (RR) Algoritması:** **PRIORITY\_3** sırasındaki işlemleri belirlenen bir zaman dilimi (kuantum) için çalıştırır.
4. **runProcess() ve runProcessForQuantum() Metotları:** İşlemleri belirli bir süre veya tamamlanana kadar çalıştırır.

Programda prosesler, rastgele renk şemaları kullanılarak görselleştirilir ve işlendiklerinde ilgili bilgiler gösterilir. Sistemdeki bellek yönetimi, gerçek zamanlı prosesler için ayrılan minimum bellek miktarını içerir. Kullanıcı prosesleri, sistem kaynaklarını kullanabilir ve her proses, başlangıçta belirtilen kaynaklara sahiptir.

**Tartışma ve Öneriler:**

* Dört seviyeli öncelik sistemi, yüksek öncelikli işlemleri hızlı bir şekilde çalıştırarak sistem verimliliğini artırır.
* Ancak, düşük öncelikli işlemlerin uzun süre beklemesi ve kaynak dağılımındaki dengesizlik sorunları ortaya çıkabilir.
* Proses önceliklerinin daha dengeli ve dinamik bir şekilde belirlenmesi, esnek kaynak yönetimi sağlayabilir.
* Bellek yönetiminde dinamik tahsis ve serbest bırakma stratejileri, sistem performansını iyileştirebilir.

Sonuç olarak, dört seviyeli öncelikli proses görevlendiricisi, işletim sistemlerinin verimliliğini artırma potansiyeline sahiptir. Ancak, etkinliği, kaynakların ve belleğin dengeli bir şekilde yönetilmesine bağlıdır. Bu nedenle, öncelikli görevlendiricinin avantajları, belirli durumlarda karşılaşılan dezavantajları aşabilir ve genel sistem performansını iyileştirebilir.

أعلى النموذج