



SAKARYA
ÜNİVERSİTESİ

İŞLETİM SİSTEMİ ÖDEVİ

43. GRUP

GİTHUB LİNK : https://github.com/mervegezginci/Grup43_isletimProje.git

FATMA GÜNER - B201210373 1-B

MEDİNA NİLAY BEZİRCİ - B201210035 1-C

FATMA NUR YILMAZ -B201210067 1-B

MERVE GEZGİNCİ - B201210085 1-B

1) PROGRAMIN GENEL YAPISI:

KUYRUK SINIFI:

Bu projedeki "Queue" sınıfı, Java dilinde "LinkedList" sınıfını kalıtım yoluyla kullanılarak gerçekleştirilmiştir. Bu sınıf, "Queue" interface'ini implemente ederek, kuyruk yapısını temsil eder. Kuyruk yapısı, elemanların önce girilen ilk eleman önce çıkarılacağı yapısını temsil eder. Bu sınıf içerisinde, kuyruğa eleman eklemek için "add()" metodu, kuyruktan eleman çıkarmak için "poll()" metodu ve kuyruğun boş olup olmadığını kontrol etmek için "isEmpty()" metodu bulunur. Bu sınıf, projedeki geri beslemeli sıralayıcıda kuyruklara dayalı bir yapı kullanılmasını sağlar.

GÖREVLENDİRİCİ SINIFI:

Bu projedeki "Scheduler" sınıfı, gerçek zamanlı proseslerin ve kullanıcı proseslerinin çalıştırılmasını yöneten bir yapıdır. Bu sınıf, projedeki "Queue" sınıfını kullanarak iki adet kuyruk tutar(gerçek zamanlı prosesler için bir kuyruk ve kullanıcı prosesleri için bir kuyruk.)

MAIN SINIFI:

Bu projedeki "main" sınıfı, projenin çalıştırılmasını sağlayan ana yapıdır. Bu sınıf içerisinde, projedeki "Scheduler" sınıfının metodu çağrılarak proseslerin çalıştırılması sağlanır.

"main" sınıfı içerisinde, projedeki "Process" sınıfından oluşan proseslerin listesi oluşturulur ve bu prosesler "Scheduler" sınıfının metoduna gönderilir. Bu metod çalıştırıldıktan sonra, projedeki geri beslemeli sıralayıcı çalışmaya başlar ve proseslerin çalıştırılmasını yönetir.

COLOR SINIFI:

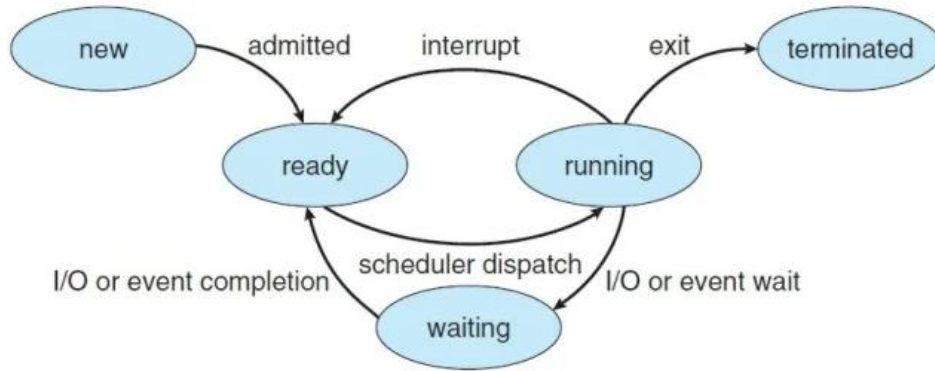
Bu projedeki "Color" sınıfı, projenin çalışma sırasında konsol ekranında renkli yazı çıktısı verilmesini sağlar. Bu sınıf, projedeki "Process" sınıfının içerisinde kullanılarak, her bir prosesin yazı çıktısının farklı bir renkte olmasını sağlar.

PROSES SINIFI:

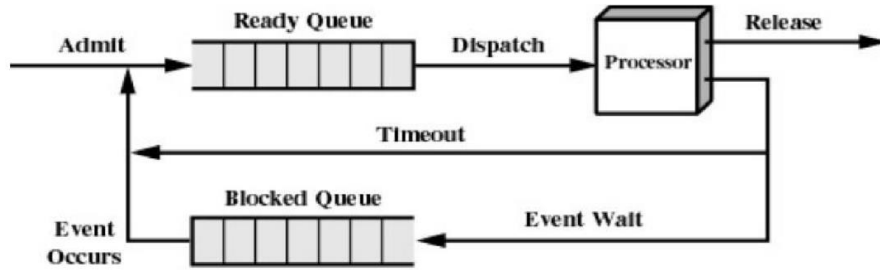
Bu projedeki "Process" sınıfı, geri beslemeli sıralayıcıda çalıştırılacak prosesleri temsil eden bir yapıdır. Bu sınıf, projedeki geri beslemeli sıralayıcıda çalıştırılacak proseslerin özelliklerini tutar. Özellikler arasında prosesin varış zamanı, öncelik derecesi ve çalışma süresi bulunur. Bu sınıf, projedeki geri beslemeli sıralayıcıda çalıştırılacak proseslerin özelliklerini tutar ve projedeki "Scheduler" sınıfı tarafından yönetilir.

2) İŞLETİM SİSTEMİ ALGORİTMALARI

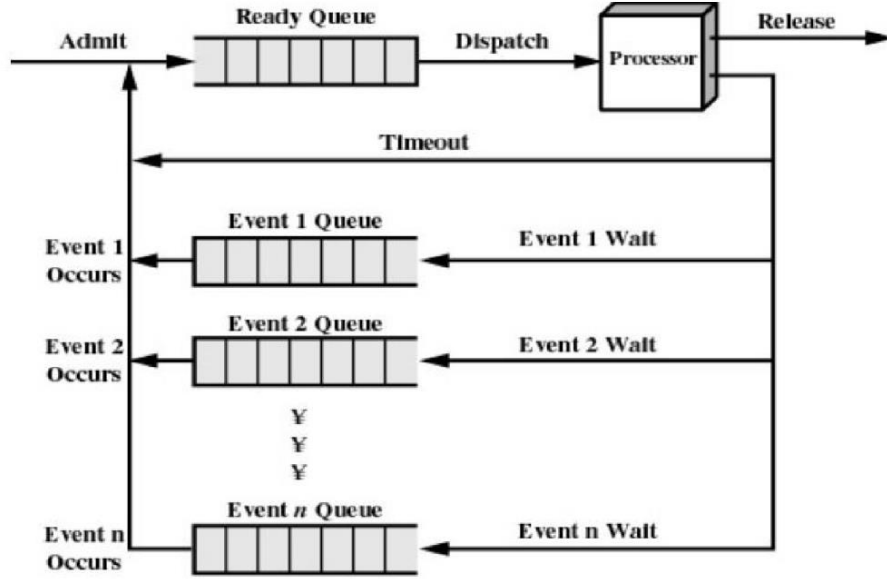
GENEL PROSES ALGORİTMASI



TEKLİ KUYRUK ALGORİTMASI:



ÇOKLU KUYRUK ALGORİTMASI :



GÖREVLENDİRİCİNİN GEREKLİLİĞİ

Görevlendirici, sınırlı kaynakları olan bir çoklu programlama sisteminde çalışan prosesleri yöneten bir mekanizmadır. Bu görevlendirici, dört seviyeli öncelikli bir yapıya sahiptir.

Bu görevlendirici, sınırlı kaynakları olan bir sistemde proseslerin çalıştırılma şeklini yönetir ve bu proseslerin önceliklerine göre kuyruklara yerleştirilerek işleme alınmasını sağlar. Bu sayede, sistemdeki kaynaklar daha etkili çalışır.

3) GERÇEK SİSTEMLERLE KARŞILAŞTIRMA VE OLASI İYİLEŞTİRME

Geri beslemeli sıralayıcılar, gerçek sistemlerde çok az kullanılır. Bu tür sıralayıcıların çalışma şekilleri düşük verimlidir ve işlemlerin sırasının kullanıcı tarafından belirlenmesi zaman alıcıdır. Bu nedenle, gerçek sistemlerde genellikle daha verimli ve otomatik olarak çalışan sıralama algoritmaları kullanılır.

Bu projedeki geri beslemeli sıralayıcı, gerçek sistemlerde kullanılan sıralama algoritmalarından farklı olarak, proseslerin sırasını kullanıcı tarafından belirlenir. Bu, proseslerin sıralanması için kullanıcının etkisinin olmasını ve daha yavaş bir sıralama işleminin gerçekleştirilmesine neden olur. Bu nedenle, gerçek sistemlerde kullanılan sıralama algoritmaları ile karşılaştırıldığında, geri beslemeli sıralayıcının verimliliği daha düşüktür.

Olası iyileştirme olarak sisteme gelen processlerin sayısı ve önceliği değerlendirmeye alınıp hangi yöntemin kullanılması daha verimli olur sorusuna cevap veren bir algoritma tasarlayıp bu yöntemler arasında sistemin seçim yapması daha etkili bir çalışma sağlar.

BELLEK KULLANIMI:

Belleğin hangi parçalarının kullanımda olduğunu, hangi parçalarının (bitişken alanların) kullanılmadığını izlemek, Süreçlere (işlemlere) bellek tahsis etmek (allocate), tahsis edilen belleği geri almak (deallocate), Bellek ile disk arasındaki yer değiştirme (swap) işlemlerini gerçekleştirmektir.

