

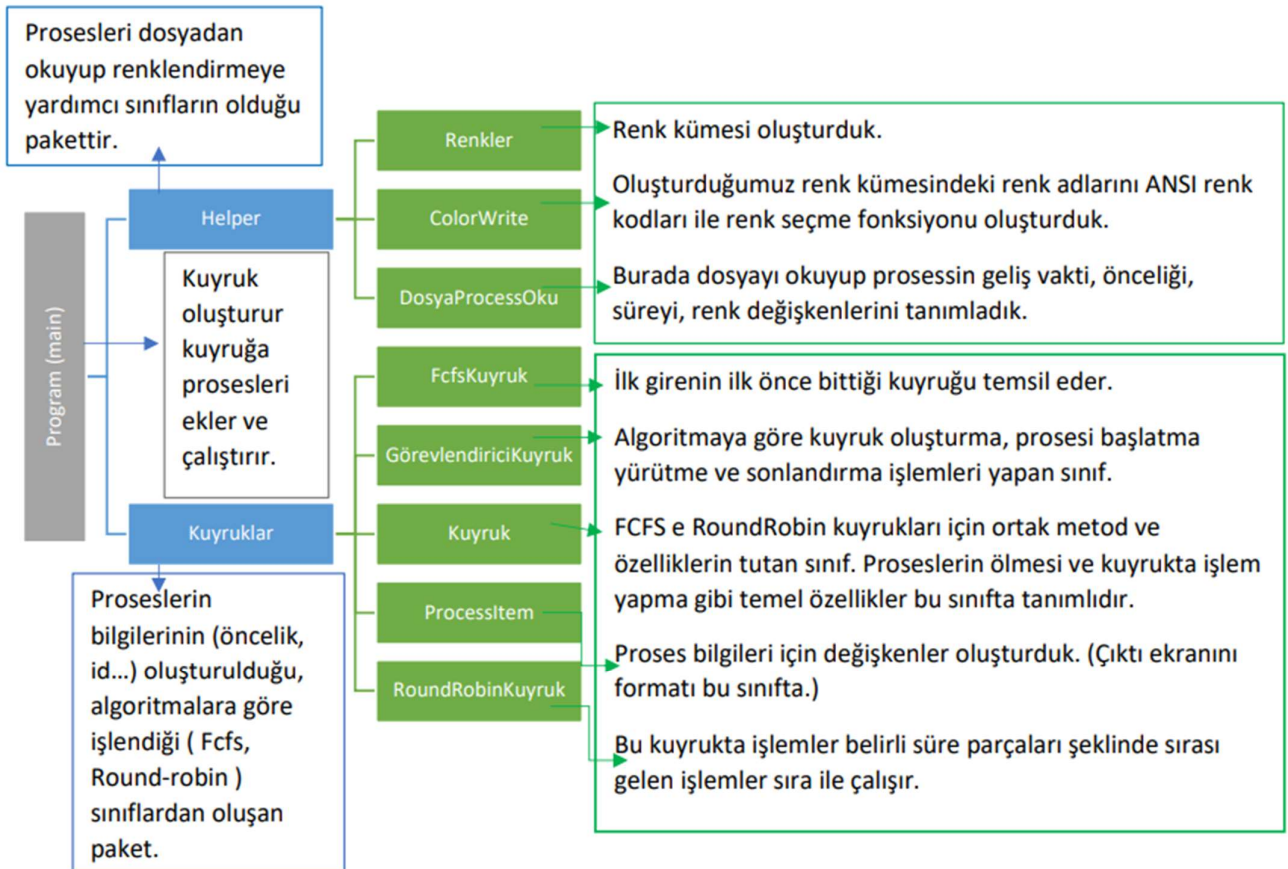
İŞLETİM SİSTEMLERİ PROJESİ

Grup No: 22

Grup Üyeler:[HOZAİFA ALKHEDER ,GÖKDENİZ DEMİR, İLAYDA NUR DAMAR ,MAHMUD EMRE BAŞPEHLİVAN, ASLI KIZILASLAN]

ÖĞRENCİ NO:[G211210572 , G201210026, G211210068, G211210020, G211210074]

Github linki: https://github.com/ilydanrdmar/isletim_sistemleriproje/tree/main



İŞLEVLER

Program proses işlemlerini gerçekleştiren ve çıktı olarak sunan 3 paketten oluşuyor. **Helper** paketinde dosya okuma ve çıktıyı renklendirme gibi yardımcı işlemleri sağlıyor. **Kuyruklar** paketi kuyruk oluşturma, proses bilgilerini alma, işlemeye yarayan sınıflardan oluşur. **Main** paketi program dosyasının bulunduğu ana işlemi gerçekleştirir. Kuyruk oluşturur ve prosesleri kuyruğa ekler.

Helper paketi 3 sınıftan oluşur.

Renkler sınıfında renklerin şeması oluşturduk. Bu proseslerin renklendirmede yardımcı olacak.

ColorWrite sınıfında renk şemasını ANSI renk kodlarıyla renklere atadık ve renk seçme fonksiyonu oluşturduk. Sonrasın da Çıktıyı ekrana yazdıran bir fonksiyon oluşturduk.

DosyaProcessOku sınıfımızda dosyayı okur, geliş zamanı, öncelik, süre olarak prosesi ekler ve rastgele bir renk verir.

Kuyruklar paketi ise 5 sınıftan oluşur.

ProcessItem

Proseslerin bilgilerin tutulduğu değişkenler ve proses bilgilerini ekrana yazdıran fonksiyon vardır. Aynı zamanda biten prosesleri kontrol eden fonksiyon, ölü prosesleri ölen prosesler listesine ekleyen fonksiyonun bulunduğu sınıftır.

FCFSKuyruk

İlk giren işlemin ilk önce bittiği kuyruğu temsil eden sınıftır. Kuyruğa yeni proses ekleme ana Kuyruk sınıfından devalınmıştır. Kuyruğa giren ve zamanı gelen işlemler bitene veya ana sistem tarafından öldürüne kadar sırası gelen işlem başlayıp bitmektedir. FCFS kuyrukta işlem bölünmesi yoktur. Başlayan işlem bitene devam etmektedir.

RoundRobinKuyruk

Bu kuyruk da işlemler, belirli süre parçaları şeklinde sırası gelen işlemler, sıra ile çalışmaktadır. Özetler isek işlem parça süresi $Q=1$ sn olan Round-Robin Kuyrukta 0. Saniyede P1, P2, P3 6 saniyelik 3 işlem gelir ise 0. Saniye P1, 1.Saniye P2 2.Saniye P3,3.Saniye tüm sırası gelenler çalıştığı için yine P1 işlemi ile devam edecektir.

GörevlendiriciKuyruk

Bu kuyrukta 4 adet kuyruk bulunmaktadır. Bu kuyruklarda öncelik değerleri bulunmaktadır."0." Derece öncelik FCFS kuyrukla bulunur. "1. 2. 3." derece için ise toplamda 3 tane Round-Robin Kuyruk bulunmaktadır. İşlem yapar çalışacak proses önceliğe göre iç sınıflardan bir tanesinden seçilir eğer 0. Derece olan FCFS kuyruktan proses seçerek tamamlanana kadar çalıştırılır. Diğer derecelerde ise, kuyrukta çalışması gereken işlemler, sıra ile kuyrukta çalışması gereken prosesler çalışacaktır.

Kuyruk

Nesne yönelimli programlama mantığının da FCFS ve Round-Robin kuyrukları için ortak metod ve özelliklerin tutulduğu sınıftır. Kuyruğa proses ekleme, bekleyen proseslerin ölmesi ve kuyrukta işlem yapılacak proseslerin saklanması gibi temel özellikler bu sınıf içinde tanımlıdır.

Main

Program ise kuyruğu oluşturup çalıştırıyor. Eğer argüman eksikse argümanı eklemenizi istiyor. (giriş.txt verilen örnek argüman)

GÖREVLENDİRİCİ KABUK HAKKINDA

Kullandığımız FCFS yöntemi prosesler sisteme geliş sırasına göre kesintisiz çalışırlar. Süresi uzun olan proseslerden dolayı kısa prosesler gereksiz askıda kalırlar. Etkileşimli prosesler için etkili bir yöntem değildir. Bu yüzden genelde diğer yöntemlerle birlikte kullanılır. Yoğun prosesler için uygun olmadığı boş ve gereksiz beslemelerden anlaşıyor. Bizde bu projede olduğu gibi Round-Robin ile kullandık. Round-Robin ise kesintili bir yöntemdir. Zaman dilimlerinden oluşur ($q=1 \dots$). Etkileşimli prosesler varsa daha etkilidir. Zaman dilimi önemi şu şekildedir, büyük zaman dilimi seçilirse FCFS gibi olacaktır. Küçük seçilirse zaman kaybına ve işlemin etkin olmasını engelleyebilir.

Örneğin, bu kuyruk fazla beklemeden en hızlı şekilde çalışıyor ama öncelik azalınca proses bekleme süresi artıyor. Örnekte toplam 20 sn olduğu için çok düşük öncelikli işlem çalışmadan ölüyor.

Linux'ta ise kesintili ve öncelik tabanlı yönelim vardır. İki Adet önceliği vardır (active-expired). Zaman aralığında kalan zaman süresi kadar görev çalışabilir (active) eğer zaman kalmamışsa (expired), diğer görevler kendi zaman aralıklarını kullanana kadar çalışamaz. Linux etkileşimli prosesler için yanıt süresi zayıftır. FCFS kuyruk yapısına benzer bu yönüyle. Kaynak dağıtımına daha çok dikkat eder.

Windows'ta öncelik tabanlı kesintili bir iş sıralama yaklaşımı vardır. En yüksek öncelikli iş parçacığı bir sonra çalışır. Gerçek zamanlı iş parçacıkları diğerlerini kesebilir. Öncelik sınıfları vardır. İş parçacıklarının öncelikleri bu sınıflara göre belirlenir. Eğer çalışabilir bir iş parçacığı yoksa, boşta iş parçacığı çalışır. Optimizasyon ve iyileştirmelere daha çok dikkat eder.