

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Ders : İşletim Sistemleri
Dönem : 2022-2023 Güz Dönemi

Ad Soyad : Muhammad Najmuddin Farid
Öğrenci No : B201210556
Eposta : muhammad.farid@ogr.sakarya.edu.tr
Öğretim : 1B

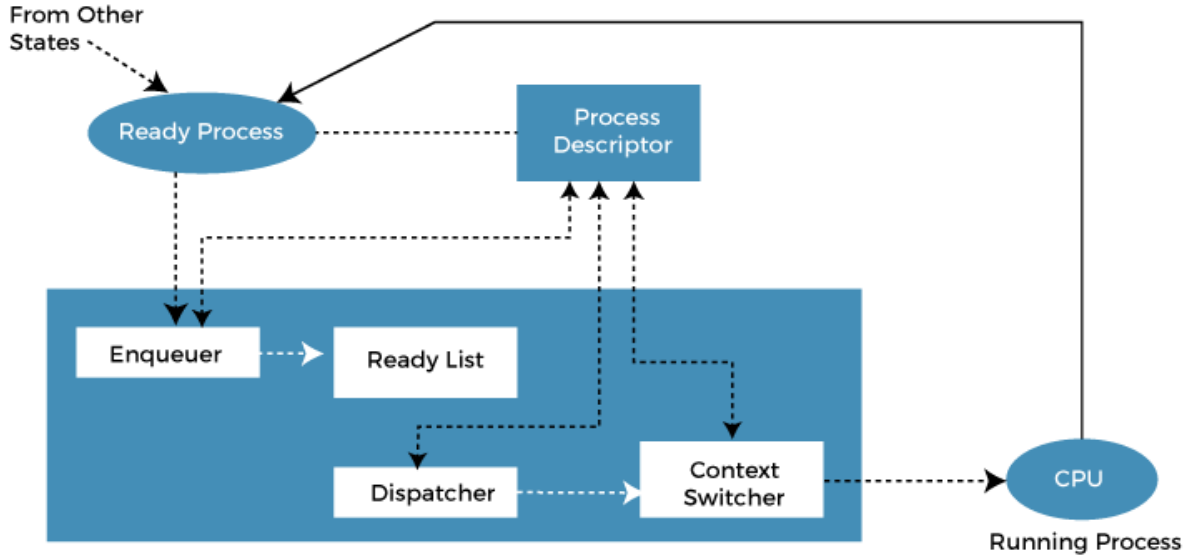
Ad Soyad : Resul Daşpınar
Öğrenci No : B201210092
Eposta : resul.daspinar@ogr.sakarya.rdu.tr
Öğretim : 1-C

Github Link : <https://github.com/resul66/Operation-project>

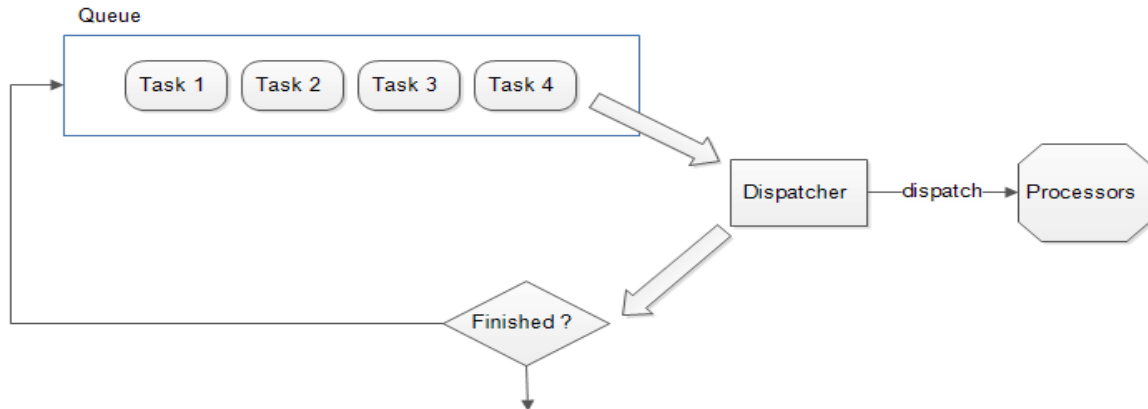
Tanıtım

Dispatcher, yalnızca zamanlayıcıdan sonra devreye giren özel bir program türüdür. Bir zamanlayıcı, süreçlerini seçmeyi tamamladığında, dağıtıcı, amaçlanan süreci istenen kuyruğa/duruma götürür. Dağıtıcı temel olarak CPU üzerinde tam kontrole sahip bir süreç sağlayan bir modüldür (kısa vadeli planlayıcı sonunda onu seçtikten sonra). Bu işlev aşağıdakileri gerektirir:

- Bağlam değiştirme
- Kullanıcı moduna geçiş
- Belirli bir programı yeniden başlatmak için kullanıcı programının uygun konumuna atlama.



Dört işlemin, Task 1, Task 2, Task 3 ve Task 4'ün hazır kuyruğunda olduğunu varsayalım. Varış süreleri sırasıyla T1, T2, T3 ve T4'tür. Ve tüm bu süreçte veya görevde bir İlk Giren İlk Çıkar (FIFO) yada First Come First Served (FCFS) programlama algoritması kullanılır.



Önce Task 1 işlemi geldi, bu nedenle zamanlayıcı yürütülecek ilk işlem olduğuna karar verecek ve Dağıtıcı, Task 1'i hazır kuyruğundan çıkaracak ve CPU'ya verecektir.

Daha sonra programlayıcı, yürütülmesi gereken bir sonraki işlem olarak Task 2 işlemini belirleyecektir, böylece Dağıtıcı yeni bir işlem için kuyruğa döndüğünde, Task 2 işlemini alacak ve CPU'ya verecektir. Bu süreç Task 3 ve ardından Task 4 için aynı şekilde devam eder.

Kod Açıklama

A. FCFS/FIFO algoritması

İlk gelen ilk çıkar (FIFO), aynı zamanda ilk gelen ilk hizmet alır (FCFS) olarak da bilinir, en basit zamanlama algoritmasıdır. FIFO, işlemleri hazır kuyruğuna geldikleri sıraya göre sıralar. Bunda, önce gelen işlem önce yürütülür ve bir sonraki işlem ancak bir önceki tam olarak yürütüldükten sonra başlar.

B. Round Robin Modu

Round Robin oluşturulan projede prosesleri kaydırmak için kullanılır. Bir program kullanarak Round Robin'de aşağıdaki zamanlar hesaplanır

- Tamamlanma Zamanı : İşlemin yürütülmesini tamamladığı zaman.
- Dönüş Süresi : Süre Tamamlanma zamanı ile varış zamanı arasındaki fark. $Dönüş\ Zamanı = Tamamlanma\ Zamanı - Varış\ Zamanı$
- Bekleme Süresi : Zaman Geri dönüş süresi ile patlama süresi arasındaki fark.
- Bekleme Süresi = Dönüş Süresi – Patlama Süresi

C. Prosesler

Yürütülmekte olan programa proses denir. Projede yapılan durumlar hazır(ready), Çalışıyor(running), Bekliyor(waiting) ve öldürüldü(killed). Proseslerin temsil eden altı unsur vardır. Bu elemanlar, proses kimliği, varış zamanı, öncelik, süre süresi, durum ve renktir.

Prosesin durumunu bildirmek için dört ana işlev vardır. StartProcess işlemin çalışma durumunu bildirir, TerminatedProcess işlemin öldürülen durumunu bildirir, ve HangedProcess işlemin bekleme durumunu bildirir.

D.Program anlatım

Program Öncelikle txt dosyasını okuyor daha sonra ilk adımda kuyruk boyutu belirleniyor.Color sınıfında Renk paletleri oluşturuldu. Daha sonra ikinci okuma gerçekleşiyor.ikinci okumada kuyruk oluşturulup kuyruğa önceliklerine göre eleman atanıyor.Proje de belirlenene göre öncelik “0” ise realtimequeue “1,2,3” ise usertimequeue oluşturuldu.Projede 20 sn kontrolü yapıldı süresi 20 sn den fazla olanlar 20. Saniyeden sonra prosesleri durduruldu..Prosesler önceliklerine göre çalışmaya başlıyor ve sonlanıp bitiyor.

```
Problems Javadoc Declaration Console X
<terminated> main (5) [Java Application] C:\Program Files\Eclipse Adoptium\jdk-17.0.5-hotspot\bin\javaw.exe (8 Oca 2023 15:31:04 – 15:31:15) [pid: 8792]
proses başladı
0 sn poses yürütülüyor (id: 0 öncelik: 0 kalan süre: 4)
1 sn poses yürütülüyor (id: 0 öncelik: 0 kalan süre: 3)
2 sn poses yürütülüyor (id: 0 öncelik: 0 kalan süre: 2)
3 sn poses yürütülüyor (id: 0 öncelik: 0 kalan süre: 1)
proses sonlandı
proses başladı
0 sn poses yürütülüyor (id: 1 öncelik: 0 kalan süre: 4)
1 sn poses yürütülüyor (id: 1 öncelik: 0 kalan süre: 3)
2 sn poses yürütülüyor (id: 1 öncelik: 0 kalan süre: 2)
3 sn poses yürütülüyor (id: 1 öncelik: 0 kalan süre: 1)
proses sonlandı
proses başladı
0 sn poses yürütülüyor (id: 2 öncelik: 1 kalan süre: 5)
proses askıda
proses başladı
0 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 20)
1 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 19)
2 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 18)
3 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 17)
4 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 16)
5 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 15)
6 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 14)
7 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 13)
8 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 12)
9 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 11)
10 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 10)
11 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 9)
12 sn poses yürütülüyor (id: 4 öncelik: 0 kalan süre: 8)
```

Linkler

Github Linki : <https://github.com/resul66/Operation-project>