

# SAKARYA ÜNİVERSİTESİ

Bilgisayar ve Bilişim Bilimleri  
Fakültesi

Bilgisayar Mühendisliği Bölümü

İşletim Sistemleri Dersi

Proje Ödevi

**Ad Soyad:** Doğukan Berk Kaya

**Numara:** G181210041

**Grup:** 2.Öğretim-C

**Tarih:** 08.01.2023

# RAPOR

**Konu:** Ödevde bir text dosyasından okuduğu processleri dört seviyeli öncelikli görevlendirici kullanan bir sistemde çalıştırıp süreci düzenleyen bir java programı yazılması bekleniyor.

Uygulama öncelikle dosyadan input veriyi okuyarak başlıyor. Bunun için ufak bir dosya okuma sınıfım var. Gerekli atama ve oluşturmalarından sonra aslında bütün uygulama sürekli bir saat göre çalışan bir mekanizma olarak çalışıyor. Her döngü başlangıcında inputtan o saniyeye denk gelen varış zamanlı processler okunuyor ve seviyelerine göre önce yüksek ve düşük seviyeliler sonra da düşük seviyeliler içinde 3 farklı seviye olacak şekilde kuyruklara ekleniyorlar. Processler oluşturulurken, processlerin ekrana basma sürecine birbirinden ayırt edilebilmeleri için kendilerine birer renk ataması yapılıyor. Ekranda renkli gösterme süreci beni ilk zorlayan kısım oldu. ANSI renk kodları kullanan bir renk seçici ekledim. Her process oluşumunda pid'sine göre renklerden birini alıyor ve ekrana çıkartılırken bu kod başta kullanılıyor, daha sonra da ansi rengi standart haline resetleniyor. Burada yaşadığım bir sorun, eclipsede ya da Windows terminalinde bu kodların çalışmıyor oluşu oldu. Eclipse için bir plugin kurdum ama (sanıyorum ki uygulama ubuntu'da derlenecek) bir Linux terminalde deneyemedim.

```
Console x Problems Debug Shell
<terminated> Program (5) [Java Application] C:\Program Files\Java\jdk-18\bin\javaw.exe (Jan 8, 2023, 4
0.0 sn process basladi(id:0000 oncelik:1 kalan sure:2 sn)
1.0 sn process askida(id:0000 oncelik:1 kalan sure:1 sn)
1.0 sn process basladi(id:0001 oncelik:0 kalan sure:1 sn)
2.0 sn process sonlandi(id:0001 oncelik:0 kalan sure:0 sn)
2.0 sn process basladi(id:0003 oncelik:0 kalan sure:3 sn)
3.0 sn process yurutuluyor(id:0003 oncelik:0 kalan sure:2 sn)
3.0 sn process yurutuluyor(id:0003 oncelik:0 kalan sure:2 sn)
4.0 sn process yurutuluyor(id:0003 oncelik:0 kalan sure:1 sn)
4.0 sn process yurutuluyor(id:0003 oncelik:0 kalan sure:1 sn)
5.0 sn process sonlandi(id:0003 oncelik:0 kalan sure:0 sn)
5.0 sn process basladi(id:0006 oncelik:0 kalan sure:4 sn)
6.0 sn process yurutuluyor(id:0006 oncelik:0 kalan sure:3 sn)
6.0 sn process yurutuluyor(id:0006 oncelik:0 kalan sure:3 sn)
7.0 sn process yurutuluyor(id:0006 oncelik:0 kalan sure:2 sn)
7.0 sn process yurutuluyor(id:0006 oncelik:0 kalan sure:2 sn)
8.0 sn process yurutuluyor(id:0006 oncelik:0 kalan sure:1 sn)
8.0 sn process yurutuluyor(id:0006 oncelik:0 kalan sure:1 sn)
9.0 sn process sonlandi(id:0006 oncelik:0 kalan sure:0 sn)
9.0 sn process basladi(id:0007 oncelik:0 kalan sure:4 sn)
10.0 sn process yurutuluyor(id:0007 oncelik:0 kalan sure:3 sn)
10.0 sn process yurutuluyor(id:0007 oncelik:0 kalan sure:3 sn)
11.0 sn process yurutuluyor(id:0007 oncelik:0 kalan sure:2 sn)
11.0 sn process yurutuluyor(id:0007 oncelik:0 kalan sure:2 sn)
```

Daha sonrasında uygulama kuyuklar içinde seçim yapıyor ve bu processı yazdığım process işleyici sınıfa gönderiyor. Öncelikle inputları alıp yüksek önceliklilere fcfs yapan en sonda da kalanları round robin yapıp tüketen bir şekilde iskelet oluşturup sonra araları doldurmayı deneyerek başladım. Bir diğer en çok takıldığım nokta bu oldu. Program bir noktada kuyruk önceliğini pcblerin içinden dışa doğru düşürmem gereken bir yere vardı ve bu noktada bütün mimarının baştan düşünülmesi gerekiyordu. Process işleyicinin ayrı olması gibi şeylerden bundan sonra eklendi. Yine beni en çok zorlayan konulardan biri processbuilder oldu. Yaklaşık 3-4 günümü aldı. Builder oluşturmayı, .start() ile process oluşturmayı, environmentı maplemeyi (her ne kadar bu konularda boğulsam da) ya da processin input output akışını yönlendirmeyi ya da çalıştıracağı uygulamayı vermeyi anladım. Ama tam olarak ödevde nasıl processin bilgisini bunda tutacağımı nasıl entegre edeceğimi çözemedim. Belki de ayrıca bunun için bir program yazıp içinde çalıştırılabilir diye düşündüm ama tek jar ile nasıl yapacağımı bunu nasıl kullanacağım konusunda pek deneyimim olmadığı için ve daha sonraları bu kadar düşünebildiğim için kullanamadım.

Döngüde yüksek öncelikli, bir iki ve üç seviyeli kuyuklar ve son olarak RoundRobin olarak çalışacak toplamda 5 kuyruk var. Seçim şu şekilde yapılıyor: Her döngüde öncelikle yüksek öncelikli kuyruktan bekleyen olup olmadığı kontrol ediliyor.

Eğer var ise önce yüksek öncelikli kuyruk First come first served olarak process işleyiciye gönderiliyor. Processlere total zamanları kadar bir q ataması yapılıyor ve q bitimine kadar çalıştırılıyorlar. Biten processler kuyruktan çıkarılıyor.

Eğer yüksek öncelikli kuyruk boş ise daha düşük öncelikli olan kullanıcı processlerine bakılıyor. Önce 1. Seviye, eğer boşsa sonra 2 sonra 3. Seviye kuyuklar kontrol ediliyor. Process olan kuyruktan baştaki process çekilip işleticiye gönderiliyor. Her process q kadar zaman tanınıyor. Bir q'yu giriş.txtdeki processlerin maksimum 4 process zamanlı olmasından dolayı zaman aşımalarını da gösterebilmek için 2q olarak belirledim. Ancak optimizasyon açısından da bakarsak genellikle ortalama q süresinin yarısından biraz daha fazlası tercih edilir. Program farklı q değerleri de alarak çalışabiliyor. Ve yine kuyruktan önce küçük process zamanlı processlerin alınıp işleme koyulması uygulamaların daha hızlı çalıştığı algısını yaratacaktır. Benzer bir örnek olarak Google Chrome ve Microsoft Edge ilk çıkış zamanı verilebilir. Overall performans açısından edge daha iyi olmasına rağmen Chrome'daki küçük ya da önce bakılacak işlerin önce yüklenmesi, edgede ise her şeyin tamamlandıktan sonra gösterilmesi gibi farklar Chrome'un daha hızlı görünmesi yanılgısına düşürmüştür insanları. Aynı şekilde özellikle mobil işletim sistemleri performanstan ödün veremeyeceği ve donanımdan tam yararlanmak istediği için bu tarz özellikleri kullanır.

Eğer process zamanı bitmeden q için atanan zaman tükenirse process zaman aşımına uğrar ve önceliği düşürülerek seviye1 seviye2ye, seviye 2 3'e, 3 de RoundRobin yapacak son seviye kuyruğa olacak şekilde bir alt kuyruğa sondan eklenir.

Her saniye okuma yapılırken kontrol edilir, eğer process işlenirken boş olan yüksek öncelikli kuyruğa bir ekleme yapılır, yeni bir process gelirse düşük seviyeli olan process askıya alınır ve yüksek öncelikli process kuyruğunu bitene kadar işlenmesiyle döngülere devam edilir. Daha sonrasında düşük seviyeli kuyuklar aynı sistemle tekrar çalışıp askıdaki processı process işleticiye gönderir.

Her bütün bitirilememiş ve 3 seviyeyi de düşmüş processler en son round robin kuyruğunda toplanır ve burada q 1 olacak şekilde bitene kadar hepsi işleticiye gönderilip bitirilirler. Her q sonunda bitmemiş olanlar round robin kuyruğunun sonuna eklenip sırasını bekler.

Processler işlenirken ve işlem sonrasında bilgilerini ekrana basıyorlar. Ve döngü en son tüm kuyruklar boş kaldığında tümüyle sonlanır program sona erer.

Program farklı daha yüksek process zamanlı inputların olduğu dosyalarla da test edildi.

Odevi Yapanlar: Doğukan Berk Kaya G181210041

GitHub Linki: [https://github.com/dogukankaya1/G181210041-](https://github.com/dogukankaya1/G181210041-IsletimSistemleriProjeOdevi)  
IsletimSistemleriProjeOdevi