



**SAKARYA**  
ÜNİVERSİTESİ

## **İşletim Sistemleri Dersi Projesi**

**B191210015-Pınar KARIŞMAZ-1A**

**G201210057-Muhammet Barış YILMAZ-2A**

**B201210042-Yusuf BİÇER-1A**

**B201210069-Süleyman Barış DALLI-1A**

### **Github Link**

**<https://github.com/pinarkarismaz/PBYS/tree/master>**

## Projedeki Sınıflar

### Process Sınıfı

Txt. Dosyasından okunan her satırın sahip olacağı nitelikleri ve fonksiyonları belirler.

```
public Process(int id,int VarisZamani, int Oncelik, int ProcessZamani, int timeout) {  
    this.VarisZamani = VarisZamani;  
    this.Oncelik = Oncelik;  
    this.ProcessZamani = ProcessZamani;  
    this.id=id;  
    this.timeout = timeout;  
    this.ProcessIsWorking = 0;  
}
```

### Dispatcher Sınıfı

Görevlendirme sınıfıdır. Oluşturulmuş kuyrukların hangi sıra ile çalışacağını , askıya alınan ve zaman aşımına uğrayan prosesleri belirler. Tüm prosesler önce ana bir kuyruğa alınır oradan da öncelikli kuyruklara taşınır. Öncelik sıralarına ve geliş durumuna göre çalıştırılır.

```
else if(processQueue.peek().getProcessZamani() == 0)  
{  
    TimeoutKontrol(processQueue);  
    System.out.println("Saniye: " + clock.getTime() + " PROSES BİTTİ"+" ID:"+processQueue.peek().getId()+" ONCELİK:"  
    processQueue.remove();  
    sayac=0;  
    if(priority0.peek()!=null)  
    {...}  
    else if((priority0.peek()!=null))  
    {  
        basla(priority0);  
    }  
}
```

Her biten proses ardından proses biter bitmez sıradaki prosesin başlaması için basla() fonksiyonu içinde kendisini çağırır.

```
private void TimeoutKontrol(Queue<Process> processQueue) {  
    if (processQueue.peek().getOncelik() == 0){  
        if (priority1.peek() != null){...}  
    }  
  
    else if (processQueue.peek().getOncelik() == 1){...}  
    else if (processQueue.peek().getOncelik() == 2) {...}  
}
```

TimeoutKontrol() öncelik durumlarının kontrol edilip proseslerin askıya alınıp alınmayacağını ve zaman aşımı durumunu kontrol eder.

## Clock Sınıfı

Zaman aşımı ve proses çalışma durumları için oluşturulmuştur.

```
private int timer;

1 usage  ⬆ Baris Yilmaz
Clock() { timer = 0; }

2 usages  ⬆ Baris Yilmaz +1
public void TimeAntsin() throws InterruptedException {
    TimeUnit.SECONDS.sleep( timeout: 1);
    timer++;
}

13 usages  ⬆ Baris Yilmaz
public int getTime() { return timer; }

no usages  ⬆ Baris Yilmaz
public void setTime(int timer) { this.timer = timer; }
```

## Data Sınıfı

Dosyadan okunan verinin özelliklerini belirler.

```
public int VarisZamani;

3 usages
public int id;
3 usages
public int Oncelik;
3 usages
public int ProcessZamani;

no usages  ⬆ Baris Yilmaz
public int getVarisZamani() { return VarisZamani; }
no usages  ⬆ Baris Yilmaz
public int getOncelik() { return Oncelik; }
no usages  ⬆ Baris Yilmaz
public int getProcessZamani() { return ProcessZamani; }
no usages  ⬆ pinarkarismaz
public int getId(){return id;}
```

## Renklendirme İşlemi

```
ArrayList<String> renkler = new ArrayList();
renkler.add("\u001B[35m");
renkler.add("\u001B[30m");
renkler.add("\u001B[31m");
renkler.add("\u001B[32m");
renkler.add("\u001B[33m");
renkler.add("\u001B[34m");
renkler.add("\u001B[35m");
renkler.add("\u001B[36m");
renkler.add("\u001B[37m");
Random rnd = new Random();
```

## Main Sınıfı

Dosyadan okuma işlemini yapar ve proses nesnelerini oluşturur. Sonra görenlendirici sınıfını çağırır.

```
//List<String> data = new ArrayList<>();
Data data = new Data();

while ((line = bufferedReader.readLine()) != null) {
    line = line.replaceAll(regex: " ", replacement: "");

    String[] values = line.split(specialCharacter);
    data.id=idver;
    idver++;
    data.VarisZamani = Integer.parseInt(values[0]);
    data.Oncelik = Integer.parseInt(values[1]);
    data.ProcessZamani = Integer.parseInt(values[2]);

    Process process = new Process(data.id,data.VarisZamani,data.Oncelik,data.ProcessZamani, timeout: 0);
    sure = process.getProcessZamani() +sure;
    processQueue.add(process);
}

Dispatcher dispatcher = new Dispatcher();
dispatcher.OncelikliKuyrugeTasi(processQueue,sure);
```

## Örnek Çıktılar

```
Saniye: 0 PROSES BASLADI ID:0 ONCELİK:1 KALAN SURE:5
Saniye: 1 PROSES ASKIDA ID:0 ONCELİK:1 KALAN SURE:4
Saniye: 1 PROSES BAŞLADI ID:1 ONCELİK:0 KALAN SURE:1
Saniye: 2 PROSES BITTI ID:1 ONCELİK:0 KALAN SURE:0
Saniye: 2 PROSES BASLADI ID:3 ONCELİK:0 KALAN SURE:3
Saniye: 3 PROSES CALISIYOR ID:3 ONCELİK:0 KALAN SURE:2
Saniye: 4 PROSES CALISIYOR ID:3 ONCELİK:0 KALAN SURE:1
Saniye: 5 PROSES BITTI ID:3 ONCELİK:0 KALAN SURE:0
Saniye: 5 PROSES BASLADI ID:6 ONCELİK:0 KALAN SURE:4
Saniye: 6 PROSES CALISIYOR ID:6 ONCELİK:0 KALAN SURE:3
```

```
Saniye: 15 PROSES BITTI ID:8 ONCELİK:0 KALAN SURE:0
Saniye: 15 PROSES BASLADI ID:10 ONCELİK:0 KALAN SURE:3
Saniye: 16 PROSES CALISIYOR ID:10 ONCELİK:0 KALAN SURE:2
Saniye: 17 PROSES CALISIYOR ID:10 ONCELİK:0 KALAN SURE:1
Saniye: 18 PROSES BITTI ID:10 ONCELİK:0 KALAN SURE:0
Saniye: 18 PROSES BASLADI ID:16 ONCELİK:0 KALAN SURE:4
Saniye: 19 PROSES CALISIYOR ID:16 ONCELİK:0 KALAN SURE:3
Saniye: 20 PROSES CALISIYOR ID:16 ONCELİK:0 KALAN SURE:2
Saniye: 21 PROSES ZAMAN ASIMI ID:0 ONCELİK:1 KALAN SURE:4
Saniye: 21 PROSES CALISIYOR ID:16 ONCELİK:0 KALAN SURE:1
Saniye: 22 PROSES BITTI ID:16 ONCELİK:0 KALAN SURE:0
Saniye: 22 PROSES BASLADI ID:17 ONCELİK:0 KALAN SURE:4
Saniye: 23 PROSES CALISIYOR ID:17 ONCELİK:0 KALAN SURE:3
Saniye: 24 PROSES CALISIYOR ID:17 ONCELİK:0 KALAN SURE:2
```