



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
İŞLETİM SİSTEMLERİ PROJE RAPORU

B201210033- Yusuf ÖZASLAN

B201210351- Eray BAYSAL

B201210097- Mazhar AKYOLDAŞ

B191210555- Tea SHKURTİ

B201210089- Yusuf GÜNCE

SAKARYA

Ocak, 2023

İşletim Sistemleri Dersi

Özet

Anahtar Kelimeler:

GELİŞTİRİLEN YAZILIM

Ödevde kullandığım yardımcı kaynaklar

Gelen Procces'lerin Sıralanması Ve Kuyruklara Alınması

Bu kısımda gelen process'leri öncelik değerlerine göre öncelikli kuyruklara yerleştiriyoruz.

```

307 public void split_sort(Queue processList) {
308     // _allProccesse'de bulunan prosesler öncelikli kuyruklara eklenir(_priority değişkenine göre)
309     for(int i=0;i<processList.getProcessList().size() ; i++)
310     {
311         int tempPriority=processList.getProcessList().get(i).get_priority();
312         if( tempPriority == 0) {
313             processQueue0.addProcess(processList.getProcessList().get(i));
314         }
315         else if( tempPriority == 1) {
316             processQueue1.addProcess(processList.getProcessList().get(i));
317         }
318         else if( tempPriority == 2) {
319             processQueue2.addProcess(processList.getProcessList().get(i));
320         }
321         else if( tempPriority == 3) {
322             processQueue3.addProcess(processList.getProcessList().get(i));
323         }
324         else {
325             System.out.println("...Gecersiz priority degiskeni...");
326         }
327     }
328
329     //Yukarıda priortiy'e gore sıraladık ardından kendi iclerinde arrivalTime'a gore sıralamak için bunları kullandık ---->
330     /*-->*/if(!processQueue0.isEmpty()) {processQueue0=processQueue0.sort(processQueue0);}
331     /*-->*/if(!processQueue1.isEmpty()) {processQueue1=processQueue1.sort(processQueue1);}
332     /*-->*/if(!processQueue2.isEmpty()) {processQueue2=processQueue2.sort(processQueue2);}
333     /*-->*/if(!processQueue3.isEmpty()) {processQueue3=processQueue3.sort(processQueue3);}
334 }

```

Ardından process'ler karışık sırayla geldiği için ArrivalTime değişkenlerine göre sıraladık. Bu bizi process'leri her seferinde aramaktan kurtardı. Time değişkenine göre ilk elemana bakılıyor eğer ilk elemanın ArrivalTime'ı Time değerimizden büyükse devamına bakmaya gerek kalmamaktadır. Sıralama fonksiyonunun içine bakacak olursak ; Gelen kuyruğu SelectionShort algoritmasıyla beraber ArrivalTime'a göre sıraladık.

```

27 public Queue sort(Queue queue) {
28     // Varış zamanına göre sıralanır
29     int n = queue.getProcessList().size();
30     int tmpArrival_j;
31     int tmpArrival_min;
32     myProcess temp;
33     //SelectionShort ile sıraladık
34     for (int i = 0; i < n-1; i++)
35     {
36         int min_idx = i;
37         for (int j = i+1; j < n; j++) {
38             tmpArrival_j=queue.getProcessList().get(j).get_arrivalTime();
39             tmpArrival_min=queue.getProcessList().get(min_idx).get_arrivalTime();
40
41             if ( tmpArrival_j < tmpArrival_min) {
42                 min_idx = j;
43             }
44         }
45         temp = queue.getProcessList().get(min_idx);
46         queue = queue.setProcessList(queue, min_idx, queue.getProcessList().get(i));
47         queue = queue.setProcessList(queue, i, temp);
48     }
49     return queue;
50 }

```

Proses mesajlarının renkli çıkabilmesi için ColoredSystemOutPrintLn sınıfındaki ANSI escape sequence'ları kullanılmıştır

```

1 package packet;
2
3 public class ColoredSystemOutPrintLn {
4
5     //renk kodları
6     public static final String ANSI_RESET = "\u001B[0m";
7
8     public static final String ANSI_RED = "\u001B[31m";
9     public static final String ANSI_GREEN = "\u001B[32m";
10    public static final String ANSI_YELLOW = "\u001B[33m";
11    public static final String ANSI_BLUE = "\u001B[34m";
12    public static final String ANSI_PURPLE = "\u001B[35m";
13    public static final String ANSI_CYAN = "\u001B[36m";
14    public static final String ANSI_WHITE = "\u001B[37m";
15
16    public static final String ANSI_BRIGHT_BLACK = "\u001B[90m";
17    public static final String ANSI_BRIGHT_RED = "\u001B[91m";
18    public static final String ANSI_BRIGHT_GREEN = "\u001B[92m";
19    public static final String ANSI_BRIGHT_YELLOW = "\u001B[93m";
20    public static final String ANSI_BRIGHT_BLUE = "\u001B[94m";
21    public static final String ANSI_BRIGHT_PURPLE = "\u001B[95m";
22    public static final String ANSI_BRIGHT_CYAN = "\u001B[96m";
23    public static final String ANSI_BRIGHT_WHITE = "\u001B[97m";
24
25    public static final String[] COLORS = {
26        ANSI_RED, ANSI_GREEN, ANSI_YELLOW,
27        ANSI_BLUE, ANSI_PURPLE, ANSI_CYAN, ANSI_WHITE,
28        ANSI_BRIGHT_BLACK, ANSI_BRIGHT_RED, ANSI_BRIGHT_GREEN, ANSI_BRIGHT_YELLOW,
29        ANSI_BRIGHT_BLUE, ANSI_BRIGHT_PURPLE, ANSI_BRIGHT_CYAN, ANSI_BRIGHT_WHITE
30    };
31 }
32

```

Her yeni proses oluşturulduğunda COLORS[] array'indeki ANSI kodlarından biri o prosese özel bir colorId property'si şeklinde atanır

```

//her proses oluşturulduğunda farklı bir renk atanır
colorNum++;
if (colorNum >= COLORS.length) { colorNum = 0; }
_colorId = colorNum;

```

Bu colorId'ler, proses mesajlarının stringlerinde başa gelir

```

int colorId = Integer.parseInt(args[3]);
System.out.println(COLORS[colorId] + "(id: " + args[0] + " öncelik: " + args[1] + " kalan süre: " + args[2] + " sn)" + ANSI_RESET);

```

```

public void executeMessage() {
    System.out.print(COLORS[get_colorId()] + Dispatcher.timer + " sn proses başladı " + ANSI_RESET);
}
public void runningMessage() {
    System.out.print(COLORS[get_colorId()] + Dispatcher.timer + " sn proses yürütülüyor " + ANSI_RESET);
}
public void suspendedMessage() {
    System.out.print(COLORS[get_colorId()] + Dispatcher.timer + " sn proses askıda " + ANSI_RESET);
}
public void endMessage() {
    System.out.print(COLORS[get_colorId()] + Dispatcher.timer + " sn proses sonlandı " + ANSI_RESET);
}
public void overTimeMessage() {
    System.out.print(COLORS[get_colorId()] + Dispatcher.timer + " sn proses zaman asimi " + ANSI_RESET);
}

```

Proses mesajları ekrana renkli bir şekilde çıkar

```

0 sn proses basladi (id:0 oncelik:1 kalan sure:2 sn)
1 sn proses askida (id:0 oncelik:2 kalan sure:1 sn)
1 sn proses basladi (id:1 oncelik:0 kalan sure:1 sn)
2 sn proses sonlandi (id:1 oncelik:0 kalan sure:0 sn)
2 sn proses basladi (id:3 oncelik:0 kalan sure:3 sn)
3 sn proses yurutuluyor (id:3 oncelik:0 kalan sure:2 sn)
4 sn proses yurutuluyor (id:3 oncelik:0 kalan sure:1 sn)
5 sn proses sonlandi (id:3 oncelik:0 kalan sure:0 sn)
5 sn proses basladi (id:6 oncelik:0 kalan sure:4 sn)
6 sn proses yurutuluyor (id:6 oncelik:0 kalan sure:3 sn)
7 sn proses yurutuluyor (id:6 oncelik:0 kalan sure:2 sn)
8 sn proses yurutuluyor (id:6 oncelik:0 kalan sure:1 sn)
9 sn proses sonlandi (id:6 oncelik:0 kalan sure:0 sn)
9 sn proses basladi (id:7 oncelik:0 kalan sure:4 sn)
10 sn proses yurutuluyor (id:7 oncelik:0 kalan sure:3 sn)
11 sn proses yurutuluyor (id:7 oncelik:0 kalan sure:2 sn)
12 sn proses yurutuluyor (id:7 oncelik:0 kalan sure:1 sn)
13 sn proses sonlandi (id:7 oncelik:0 kalan sure:0 sn)
13 sn proses basladi (id:8 oncelik:0 kalan sure:2 sn)
14 sn proses yurutuluyor (id:8 oncelik:0 kalan sure:1 sn)
15 sn proses sonlandi (id:8 oncelik:0 kalan sure:0 sn)
15 sn proses basladi (id:10 oncelik:0 kalan sure:3 sn)
16 sn proses yurutuluyor (id:10 oncelik:0 kalan sure:2 sn)
17 sn proses yurutuluyor (id:10 oncelik:0 kalan sure:1 sn)
18 sn proses sonlandi (id:10 oncelik:0 kalan sure:0 sn)
18 sn proses basladi (id:16 oncelik:0 kalan sure:4 sn)
19 sn proses yurutuluyor (id:16 oncelik:0 kalan sure:3 sn)
20 sn proses yurutuluyor (id:16 oncelik:0 kalan sure:2 sn)
21 sn proses yurutuluyor (id:16 oncelik:0 kalan sure:1 sn)
21 sn proses zaman asimi (id:0 oncelik:2 kalan sure:1 sn)
21 sn proses zaman asimi (id:4 oncelik:2 kalan sure:2 sn)
21 sn proses zaman asimi (id:2 oncelik:3 kalan sure:2 sn)
22 sn proses sonlandi (id:16 oncelik:0 kalan sure:0 sn)
22 sn proses basladi (id:17 oncelik:0 kalan sure:4 sn)
22 sn proses zaman asimi (id:5 oncelik:2 kalan sure:3 sn)
23 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:3 sn)
24 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:2 sn)
24 sn proses zaman asimi (id:9 oncelik:2 kalan sure:4 sn)
25 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:1 sn)
25 sn proses zaman asimi (id:11 oncelik:3 kalan sure:2 sn)
26 sn proses sonlandi (id:17 oncelik:0 kalan sure:0 sn)

```