T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

İŞLETİM SİSTEMLERİ PROJESİ

GRUP 27

G211210031 - METEHAN DÜNDAR

G201210033 - SENA NUR ERDEM

G211210040 – ASUDE İLHAN

G211210016 – FATİH UYVAR TÜRKEL

G201210073 - MUHAMMED SEFA ÖZDEMİR

Github link: https://github.com/fatihuturkel/grup27.git

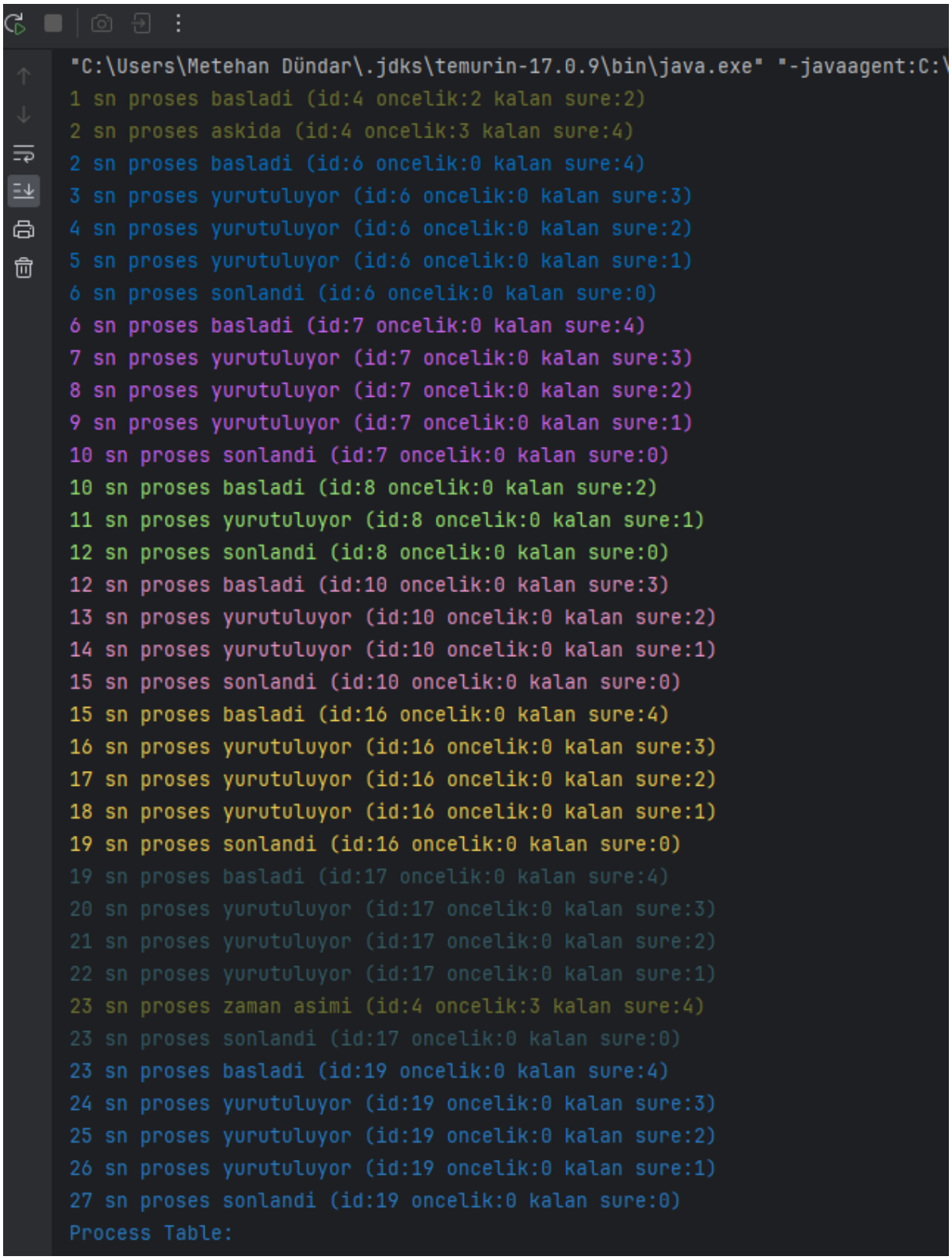
GÖREVLENDİRİCİ TASARIM PROJESİ

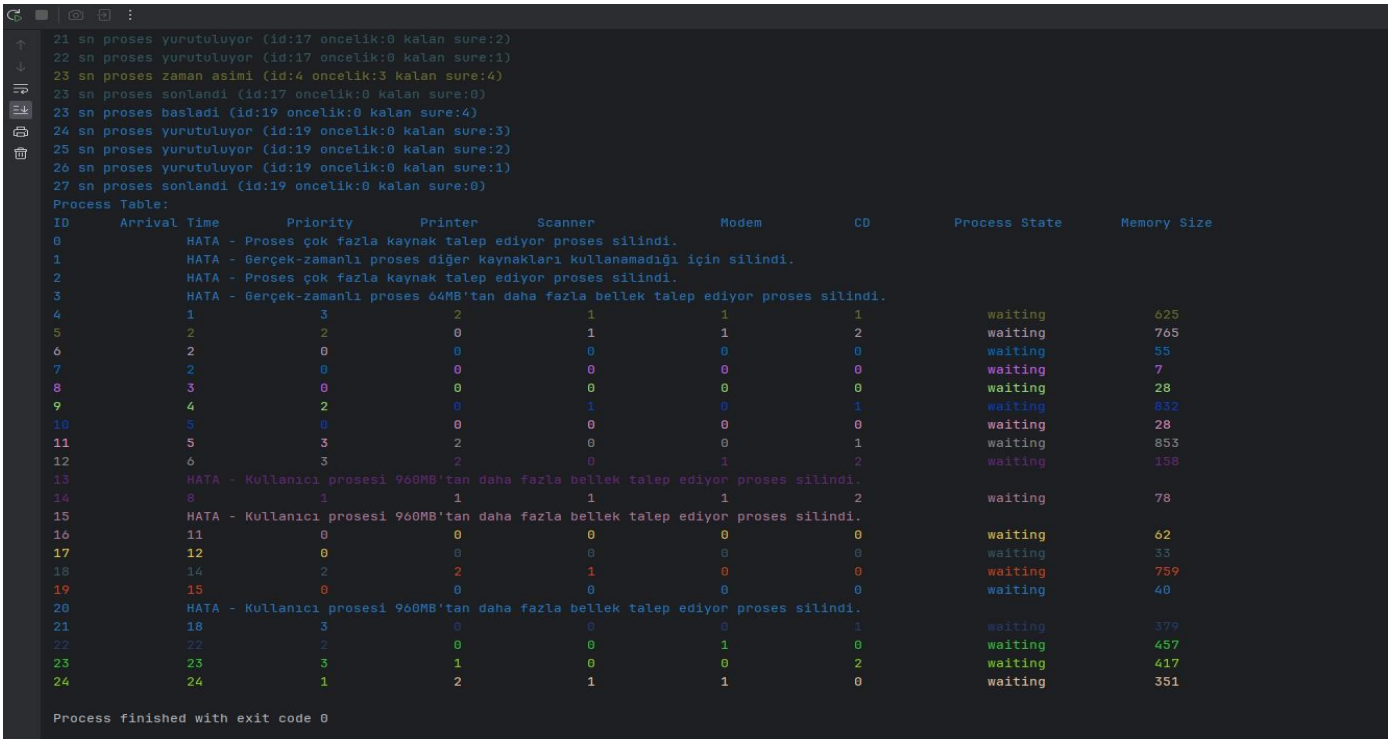
Proje, sınırlı kullanılabilir kaynakların kısıtlamaları altında çalışan, dört seviyeli öncelikli bir proses görevlendiricisine sahip çoklu programlama sistemi tasarlamaktır. Sistemde, gerçek zamanlı işlemler ve normal kullanıcı prosesleri için ayrı kuyruklar bulunmaktadır. Görevlendirici, proseslerin işlenişinde farklı öncelik düzeylerini ve bellek ile diğer kaynakların tahsisini yönetir.

Gerçek zamanlı prosesler, FCFS(İlk gelen ilk çalışır) algoritmasının bulunduğu Gerçek Zamanlı Proses Kuyruğuna giriş yaparlar. Bu kuyruktaki prosesler daha düşük öncelikle çalışan diğer proseslerden daha yüksek önceliğe sahiptir. Bu prosesler tamamlanıncaya kadar yürütülür. Kesme yapılamaz. Normal kullanıcı prosesleri, üç seviyeli bir geri beslemeli kuyrukta çalışır. Bu kuyrukların temel zamanlama kuantumu “bir” saniyedir.

PROGRAMIN GENEL YAPISI

Görevlendirici projede bulunan .txt belgesinden verileri alır. Burada varış zamanı, öncelik, proses zamanı formatında virgüllerle ayrılmış verileri kullanır. Öncelikle projemiz main classından başlar. Main classında oluşturduğumuz FileReader nesnesine .txt belgesinin yolu verilir. Dosya açılır ve satırlar bitene kadar tüm satırlar okunarak Process yapıcı fonksiyona parametre olarak verilir. Process sınıfının içinde bu gelen satır virgül karakterine göre parse edilir. Prosesin önceliği, çalışma zamanı, kalan süre olmak üzere değerler atanır. Daha sonra oluşan bu process Queue sınıfındaki AddProcessToJob metoduna gelen tüm prosesleri listeye alması açısından parametre olarak atar. Listeye tüm veriler eklendikten sonra SplitQueue komutu ile tüm veriler uygun kuyruğa konulur. Her satır PrcessBuildera parametre olarak atanır. Dosya bağlantısı kapatılır. Kuyruktaki veri kadar RunProcess metodu çalıştırılır. Öncelikle tüm Gerçek Zamanlı Prosesler tamamlanır. Daha sonra prosesler önceliklerine göre kalan 3 kuyrukta çalışmalarını tamamlarlar. Süresi biten veya expired olan proses listeden kaldırılır. 1. ve 2. Kuyrukta bitmeyen processler updateAllProcess fonksiyonuyla yeni kuyrukta yerini alır. Proses çalışmaya başlayınca, kesme olunca, devam ettikçe, zaman aşımı oldukça, bitince gerekli bilgiler ekrana yazdırılır. Her prosesin kendine ait random renk değeri olduğu için rahatça ayırt etmek mümkün.


"C:\Users\Metehan Döndar\.jdks\temurin-17.0.9\bin\java.exe" "-javaagent:C:\
1 sn proses basladi (id:4 oncelik:2 kalan sure:2)
2 sn proses askida (id:4 oncelik:3 kalan sure:4)
2 sn proses basladi (id:6 oncelik:0 kalan sure:4)
3 sn proses yurutuluyor (id:6 oncelik:0 kalan sure:3)
4 sn proses yurutuluyor (id:6 oncelik:0 kalan sure:2)
5 sn proses yurutuluyor (id:6 oncelik:0 kalan sure:1)
6 sn proses sonlandi (id:6 oncelik:0 kalan sure:0)
6 sn proses basladi (id:7 oncelik:0 kalan sure:4)
7 sn proses yurutuluyor (id:7 oncelik:0 kalan sure:3)
8 sn proses yurutuluyor (id:7 oncelik:0 kalan sure:2)
9 sn proses yurutuluyor (id:7 oncelik:0 kalan sure:1)
10 sn proses sonlandi (id:7 oncelik:0 kalan sure:0)
10 sn proses basladi (id:8 oncelik:0 kalan sure:2)
11 sn proses yurutuluyor (id:8 oncelik:0 kalan sure:1)
12 sn proses sonlandi (id:8 oncelik:0 kalan sure:0)
12 sn proses basladi (id:10 oncelik:0 kalan sure:3)
13 sn proses yurutuluyor (id:10 oncelik:0 kalan sure:2)
14 sn proses yurutuluyor (id:10 oncelik:0 kalan sure:1)
15 sn proses sonlandi (id:10 oncelik:0 kalan sure:0)
15 sn proses basladi (id:16 oncelik:0 kalan sure:4)
16 sn proses yurutuluyor (id:16 oncelik:0 kalan sure:3)
17 sn proses yurutuluyor (id:16 oncelik:0 kalan sure:2)
18 sn proses yurutuluyor (id:16 oncelik:0 kalan sure:1)
19 sn proses sonlandi (id:16 oncelik:0 kalan sure:0)
19 sn proses basladi (id:17 oncelik:0 kalan sure:4)
20 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:3)
21 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:2)
22 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:1)
23 sn proses zaman asimi (id:4 oncelik:3 kalan sure:4)
23 sn proses sonlandi (id:17 oncelik:0 kalan sure:0)
23 sn proses basladi (id:19 oncelik:0 kalan sure:4)
24 sn proses yurutuluyor (id:19 oncelik:0 kalan sure:3)
25 sn proses yurutuluyor (id:19 oncelik:0 kalan sure:2)
26 sn proses yurutuluyor (id:19 oncelik:0 kalan sure:1)
27 sn proses sonlandi (id:19 oncelik:0 kalan sure:0)
Process Table:



```
21 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:2)
22 sn proses yurutuluyor (id:17 oncelik:0 kalan sure:1)
23 sn proses zaman asimi (id:4 oncelik:3 kalan sure:4)
23 sn proses sonlandi (id:17 oncelik:0 kalan sure:0)
23 sn proses basladi (id:19 oncelik:0 kalan sure:4)
24 sn proses yurutuluyor (id:19 oncelik:0 kalan sure:3)
25 sn proses yurutuluyor (id:19 oncelik:0 kalan sure:2)
26 sn proses yurutuluyor (id:19 oncelik:0 kalan sure:1)
27 sn proses sonlandi (id:19 oncelik:0 kalan sure:0)
Process Table:
ID      Arrival Time        Priority        Printer      Scanner             Modem          CD         Process State      Memory Size
0               HATA - Proses çok fazla kaynak talep ediyor proses silindi.
1               HATA - Gerçek-zamanlı proses diğer kaynakları kullanamadığı için silindi.
2               HATA - Proses çok fazla kaynak talep ediyor proses silindi.
3               HATA - Gerçek-zamanlı proses 64MB'tan daha fazla bellek talep ediyor proses silindi.
4               1               3               2               1               1               1               waiting            625
5               2               2               0               1               1               2               waiting            765
6               2               0               0               0               0               0               waiting            55
7               2               0               0               0               0               0               waiting            7
8               3               0               0               0               0               0               waiting            28
9               4               2               0               1               0               1               waiting            832
10              5               0               0               0               0               0               waiting            28
11              5               3               2               0               0               1               waiting            853
12              6               3               2               0               1               2               waiting            158
13              HATA - Kullanıcı prosesi 960MB'tan daha fazla bellek talep ediyor proses silindi.
14              8               1               1               1               1               2               waiting            78
15              HATA - Kullanıcı prosesi 960MB'tan daha fazla bellek talep ediyor proses silindi.
16              11              0               0               0               0               0               waiting            62
17              12              0               0               0               0               0               waiting            33
18              14              2               2               1               0               0               waiting            759
19              15              0               0               0               0               0               waiting            40
20              HATA - Kullanıcı prosesi 960MB'tan daha fazla bellek talep ediyor proses silindi.
21              18              3               0               0               0               1               waiting            379
22              22              2               0               0               1               0               waiting            457
23              23              3               1               0               0               2               waiting            417
24              24              1               2               1               1               0               waiting            351

Process finished with exit code 0
```

BÖYLE BİR GÖREVLENDİRİCİ NEDEN KULLANILIR?

İşletim sistemi tasarımında kullanılan görevlendirici, işlemci zamanlama algoritmasına (CPU scheduling algorithm) göre beklemekte olan işlemlerden sıradakini alıp işlemciye yollayan programın ismidir. Buna göre bilgisayarda anlık olarak tek işlem çalışabilir ve bu işlem o anda çalışmakta olan diğer işlemler arasından seçilmiş bir işlemdir. Örneğin bilgisayarda 10 tane program açık olabilir ama CPU'da anlık olarak bir tanesi çalışır. Çalışan programların tamamı bekleme sırasında (ready queue) beklerler. Bir işlemci zamanlama algoritmasına göre bu işlemler sırayla CPU'ya gönderilerek çalıştırılırlar. İşte görevlendirici (dispatcher) bu işlemlerden sırası gelenin bekleme sırasından (ready queue) alınarak işlemciye gönderilmesi işlemini yerine getirir. Bu dispatcher yapısında kullandığımız önceliklerine göre ayrılmış 4 kuyruk yani realtime ve öncelik kuyrukları bize zaman ve düzen bakımından avantaj sağlayacktır. İyileştirme olarak kesmeye uğrayan process ler içinde ayrı bir kuyruk oluşturup orada tutabiliriz ve first come first search yerine remain time(kalan çalışma zamanı) az olan processleri çalıştırırsak programın hızında artış olacağını düşünüyorum.Kuantalama zamanını arttırarak bu şekilde daha optimal sonuçlara ulaşabiliriz.

BELLEK TAHSİSİ VE ALGORİTMA

Statik Bellek Tahsisi: Kodda, remaningMemoryForRealTime ve remaningMemoryForUser değişkenleriyle gerçek zamanlı görevler ve kullanıcı görevleri için ayrılmış statik bellek bölümleri tanımlanmıştır. Bu, statik

bölümleme (fixed partitioning) yaklaşımına benzer, burada toplam bellek önceden tanımlı bloklara ayrılmıştır.

Kaynak Kontrolü: Kodda, printer, scanner, modem ve CD gibi kaynakların kullanımı, sınırlı sayıdaki bu kaynakların her biri için ayrı değişkenlerle kontrol edilmektedir. Her prosesin bu kaynaklardan ne kadar kullanacağı belirlenmiş ve bu kaynakların kullanılabilirliği her proses eklendiğinde kontrol edilmektedir.

Kuyruklama Mekanizması: Prosesler, önceliklerine göre farklı kuyruklara (realtimequeue, userqueue_one, userqueue_two, userqueue_three) eklenir. Bu, çok seviyeli kuyruk (multi-level queue) yaklaşımını yansıtır ve her kuyruk farklı öncelik seviyesindeki prosesleri içerir.

TASARIM SEÇİMİNİN GERÇEKLEMESİ

Öncelik Bazlı Tahsis: Gerçek zamanlı görevler ve kullanıcı görevleri arasında ayrım yapılması, sistemdeki önemli görevlerin öncelikli olarak ele alınmasını sağlar. Gerçek zamanlı görevler genellikle daha kritik olduğundan, bu yaklaşım etkin bir kaynak yönetimi sağlar.

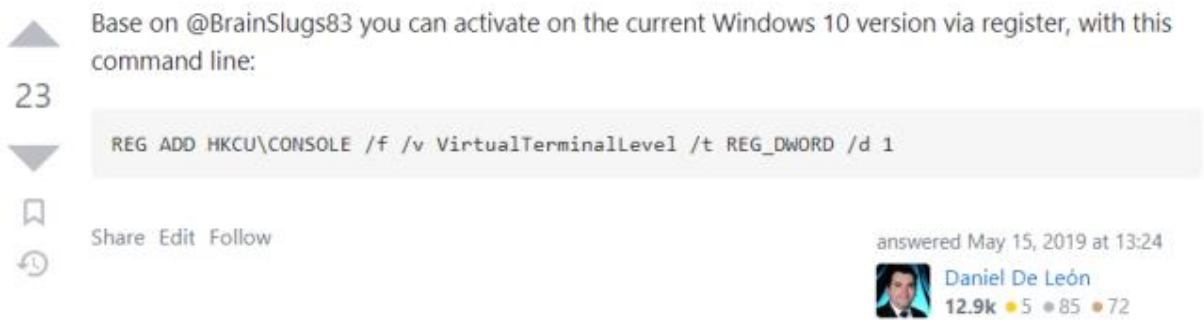
Kaynakların Verimli Kullanımı: Her prosesin ihtiyacı olan kaynak miktarının kontrol edilmesi, kaynakların gereksiz yere israf edilmesini önler ve daha fazla prosesin sistemde çalışabilmesine olanak tanır.

Esneklik ve Genişletilebilirlik: Kuyruklama mekanizması, yeni kuyruklar veya öncelik seviyeleri eklenebilir yapısıyla sistem genişletilebilir. Bu yaklaşım, farklı türdeki prosesler için farklı bellek ve kaynak gereksinimlerini daha iyi yönetebilir.

Sadelik ve Anlaşılabilirlik: Bellek ve kaynak tahsisinde basit kontrol mekanizmaları kullanılması, kodun anlaşılabilirliğini ve bakımını kolaylaştırır.

Sonuç:

Bu tasarım, bir işletim sistemi benzetiminde temel bellek ve kaynak yönetimi gereksinimlerini karşılamak için uygun bir yaklaşım sunar. Özellikle, öncelik bazlı işlem ve kaynak kontrolü, sistemdeki farklı türdeki görevlerin etkin bir şekilde yönetilmesini sağlar. Ancak, gerçek bir işletim sisteminin karmaşıklığı ve dinamikliği göz önünde bulundurulduğunda, bu yaklaşım daha karmaşık ve dinamik bellek tahsis stratejileri gerektirebilir. Örneğin, bellek taleplerinin dinamik olarak değiştiği durumlar için dinamik bölümleme veya sayfa belleği gibi daha gelişmiş yöntemler tercih edilebilir.



BAZI WINDOWS SÜRÜMLERİNDE CMD'DE RENKLER GÖZÜKMEMEKTEDİR!!!!

REG ADD HKCU\CONSOLE /f /v VirtualTerminalLevel /t REG_DWORD /d 1

Bu kod konsola girilmelidir.