



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

JAVA İLE EŞ ZAMANLI FAKTÖRİYEL HESABI

Grup Elemanları:

B151210053 - Erbil NAS

B151210113 - Umut TOSUN

SAKARYA

Nisan, 2017

Programlama Dillerinin Prensipleri Dersi

JAVA İLE EŞ ZAMANLI FAKTÖRİYEL HESABI

Erbil Nas^a, Umut Tosun^b

^a B151210053 - 1A

^b B151210113 - 1A

Özet

Java programlama dilini kullanarak büyük sayıları eş zamanlı faktöriyel hesabı yöntemiyle hesapladık. Faktöriyelin hem paralel, hem de seri olarak hesaplandığındaki sonucu ekrana milisaniye cinsinden yazdırdık. 100000! gibi büyük sayıların seri hesaplanmasının uzun sürerken paralel olarak bu faktöriyel hesaplandığında, bu sürenin 5'te 1 oranında düştüğünü gözlemledik.

© 2017 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Faktöriyel, thread, eş zamanlı hesaplama, Java

1. GELİŞTİRİLEN YAZILIM

Java'yla kodladığımız bu program, 1000000! gibi büyük sayıların faktöriyelini hesaplarken bize yardımcı oluyor. Seri hesaplama dediğimiz recursive yöntemle bu işlemi yapmak, paralel hesaplamaya göre oldukça uzun sürüyor, yaklaşık olarak beşte biri kadar (oranlar sayının büyüklüğüne göre değişiyor). Biz de bu ödevde, paralel hesaplamanın seri hesaplamaya göre daha hızlı olduğunu gösterdik. Ödev dosyasında istenildiği gibi sabit bir thread havuzu kullandık ve mantıklı thread sayıları verdik.

Programın işleyişi kullanıcının faktöriyelini hesaplamak istediği sayıyı girmesiyle başlıyor. Girdiği sayıya göre iki türden faktöriyel hesaplanıyor: Paralel hesaplama ve seri hesaplama. Önce ekrana paralel hesaplanışın süresi, sonrasında ise seri hesaplanışının süresi yazılıyor. Ardından istenilen faktöriyelin sonucu sonuc.txt dosyasına metin olarak kaydediliyor (eğer sistemde dosya yoksa program kendi oluşturuyor).

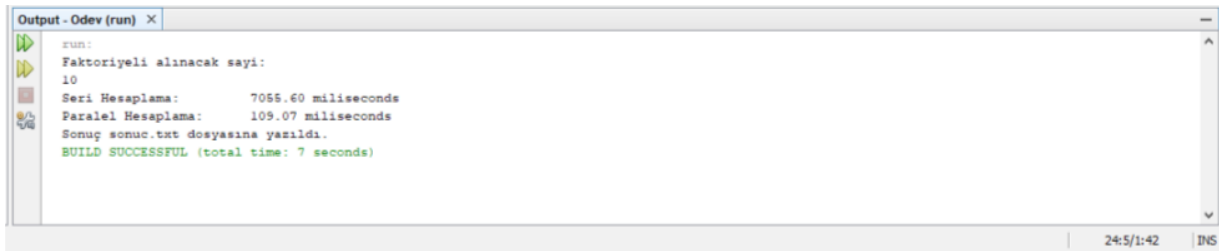
Bu programı yazarken Math kütüphanesinin BigInteger class'ı bizim için hayati bir önem taşıyor.

// BigInteger'ın kullanımı

```
import java.math.*;
```

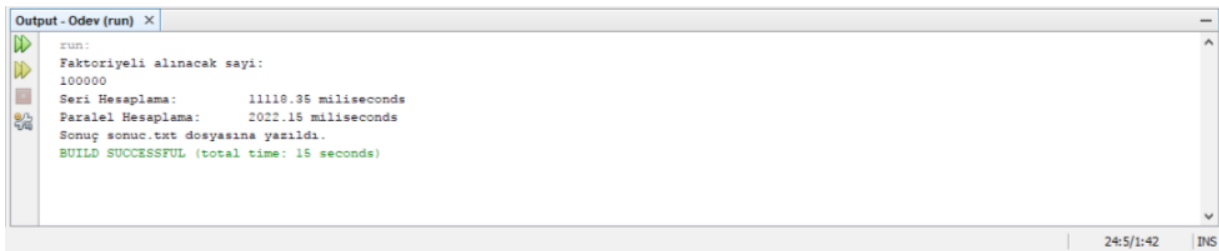
```
BigInteger sonuc = BigInteger.valueOf(1);
```

2. ÇIKTILAR



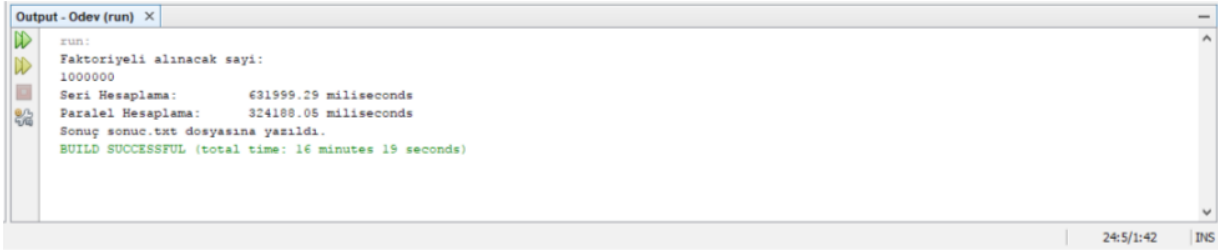
```
run:
Faktöriyel alınacak sayı:
10
Seri Hesaplama:      7055.60 milliseconds
Paralel Hesaplama:   109.07 milliseconds
Sonuç sonuc.txt dosyasına yazıldı.
BUILD SUCCESSFUL (total time: 7 seconds)
```

10!'in seri ve paralel hesaplanma süresi



```
run:
Faktöriyel alınacak sayı:
100000
Seri Hesaplama:      11118.35 milliseconds
Paralel Hesaplama:   2022.15 milliseconds
Sonuç sonuc.txt dosyasına yazıldı.
BUILD SUCCESSFUL (total time: 15 seconds)
```

100,000!'in seri ve paralel hesaplanma süresi



```
run:
Faktoriyelini alınacak sayı:
1000000
Seri Hesaplama:      631999.29 milliseconds
Paralel Hesaplama:   324188.05 milliseconds
Sonuç sonuc.txt dosyasına yazıldı.
BUILD SUCCESSFUL (total time: 16 minutes 19 seconds)
```

1,000,000!'in seri ve paralel hesaplama süresi

3. SONUÇ

Elde edilen sonuçlara göre thread kullanarak yapılan paralel hesaplama sayesinde, büyük bir sayının faktöriyelini bulmakta çok hızlı olabileceğimizi öğrendik. Sadece faktöriyel hesaplamada değil, diğer yüksek işlem gücü isteyen matematiksel problemlerde de eş zamanlı hesaplama yöntemini kullanarak çok daha hızlı ve verimli bir şekilde sonuca ulaşabiliriz.

Referanslar

- [1] <http://www.dreamincode.net/forums/topic/216487-java-program-using-threads-to-find-factorial-and-fibonacci-number/>
- [2] <http://stackoverflow.com/questions/38102071/calculate-e-with-threads-in-java>
- [3] <http://stackoverflow.com/questions/36367650/factorial-using-multithreading>
- [4] <http://stackoverflow.com/questions/10469552/factorial-function-parallel-processing>