T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

DOSYADAN OKUNAN SAYILARIN BASAMAKLARINA GÖRE TOPLAMINI SERİ ve PARALEL PROGRAMLAMA İLE HESAPLAMA

Grup Elemanları:

G171210303 – Furkan KOÇ

G151210123 – Muzaffer Haşim Gezgin

SAKARYA

Mart, 2017

Programlama Dillerinin Prensipleri Dersi

DOSYADAN OKUNAN SAYILARIN BASAMAKLARINA GÖRE TOPLAMINI SERİ ve PARALEL PROGRAMLAMA İLE HESAPLAMA

Furkan KOÇ[a], Muzaffer Haşim GEZGİN[b]

[a] *G171210303 --- 2A GRUBU*

[b] *G151210123 --- 2A GRUBU*

Özet

Bu programda amaç Sayilar.txt dosyasından okunan 4 basamakları sayıların basamaklarının ayrı ayrı toplamlarını bulup yazdıran programı yazmak. Öncelikle "okuma" class'ının içinde Try-Catch yapısı ve okuma metotlarını kullanarak .txt dosyamızı okuyoruz ve basamaklarına ayırdığımız rakamları ilgili ArrayListlerimizin içine atıyoruz. Bu okuma işleminden sonra "hesap" class'ının içinde her basamağın ArrayListini ayrı ayrı alarak toplamlarını bulup yazdırma işlemi yapıyoruz. En son olarak "Odev_4" deki main yapımızın içinden programı seri ve paralel olarak çalıştırıp hesaplama sürelerini buluyoruz ve yazdırıyoruz.

Anahtar Kelimeler: ArrayList, Dosya Okuma, Thread, Runnable, ExecutorService, Substring

1. GELİŞTİRİLEN YAZILIM

1. Okuma.java

İlk olarak bize gereken kütüphaneleri projemize import ediyoruz. Gerekli değişkenleri ve ArrayListlerimizi tanımladıktan sonra bir Try-Catch yapısı kullanarak okuma işlemimizi gerçekleştiriyoruz. Bu Try-Catch yapısının içine dosyadan okumak için olan metotları yazdıktan sonra While döngüsü açarak okuduğumuz satırları teker teker basamaklarına ayırıyoruz ve her basamağın sayısını kendi ArrayListinin içine atıyoruz.

2. Hesap.java

Bu class'ımız öncelikle Runnable metodundan kalıtım alarak oluşturuyoruz. Gerekli ArrayList'imizi ve değişkenlerimizi tanımladıktan sonra Hesap fonksiyonunun içine dışarıdan aldığımız ArrayList'imizi atıyoruz. İçeri attığımız ArrayListi kendi oluşturduğumuz arraylist'in içine atıyoruz. Daha sonrasında run() metodunun

içinde arraylistimizin uzunluğu kadar bir for açıp arraylistimizin içindeki rakamları teker teker topluyoruz. Eğer bu işlem 4 kere yapıldıysa toplamımızı ekrana yazdırıyoruz. 4 kere olmasının sebebi ise seri olarak hesapladıktan sonra paralel hesaplama yaparken tekrar yazdırma işlemi yapmak istemediğimiz için.

3. Odev_4.java

İlk olarak okuma yapabilmek için .txt dosyamızın adıyla Okuma class'ını çağırıyoruz ve okumayı yapıyoruz. Sonrasında 1 threadli bir havuz oluşturuyoruz. Hesaplamaya başladığımız anda nanoTime() fonksiyonu ile başladığımız anki süreyi alıyoruz ve 1 thread ile her sayı basamağını ayrı ayrı toplama işlemine başlıyoruz ve hepsi bittikten sonra yazdırma işlemi yapılıyor. Bu işlem bittikten sonra havuzun hala çalışmadığına emin olup işlemin bittiği anki süreyi de nanoTime() fonksiyonu ile alıyoruz. İlk aldığımız süreden ikinci aldığımız süreyi çıkartıp seri hesaplamanın ne kadar sürdüğünü buluyoruz ve yazdırıyoruz. Ondan sonra paralel hesaplamaya geçiyoruz ve bu yöntemin de tek farkı thread sayısını 1 den fazla sayıya çıkartmak oluyor. Yine aynı şekilde başlangıç süresini alıp işlemleri yapıp bitiş süresini alıyoruz ve paralel hesaplama süresini bulup ekrana yazdırıyoruz.

2. ÇIKTILAR

Yazmış olduğumuz program seri hesaplama ve paralel hesaplama yaptığı için paralel hesaplama yaparken thread sayısını ne kadar çok girersek ona göre hesaplama süresi değişecektir. Mesela thread sayısını 3 girdiğimizde hesaplamayı daha kısa bir sürede yapacağı gibi eğer thread sayısını çok fazla girersek örneğin 100 gibi bir değer girersek hesaplama süresi daha da uzayacaktır.

3. SONUÇ

Bu program bize paralel programlamanın önemini gösterdi ve eğer çok ağır işlemler yapacak olan programlarda paralel programlama yöntemini kullanırsak çok daha kısa sürede hesaplamalar yapabileceğimizi ve yazmış olduğumuz programın çok daha verimli bir şekilde çalışacağını öğrendik.

Referanslar

[1] https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-string-metodlari

[2] https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/threadler

[3] https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri

[4] https://www.callicoder.com/java-multithreading-thread-and-runnable-tutorial/