



T.C.
SAKARYA ÜNİVERSİTESİ

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

ÖDEV 2

Grup Elemanları:

G181210077 – Berke Altıntaş

G181210023 - Muhammed Melih Çelik

SAKARYA

Nisan, 2020

ÖDEV

Berke Altıntaş , Muhammed Melih Çelik

^a G181210077 2/C

^b G181210023 2/C

Özet

Programa verilen random isimler.txt dosyasından isim çekerek tc , imei , telefon ve yaşa sahip bir kişi nesnesini Kisiler.txt'ye yazdırmak. Bu süreçte tüm değerler kendi yazdığımız rastgele fonksiyonu aracılığıyla sağlanmalı.

© 2020 Sakarya Üniversitesi.

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

Anahtar Kelimeler: dosya okuma , lfsr , system.time , pointer.

ÖDEV HAKKINDA VE DENEYİMLEDİKLERİMİZ

Ödevin java kısmını zorlanmadan yaptık. Önce javada mevcut olan rand() fonksiyonunu araştırarak ve source kodlarını okuyarak System time'ı kullandığını gördük. Bunu kendi rastgele fonksiyonumuza uyarlamaya çalıştık. System.currentmilis ve System.nanotime gibi sistem zamanını kullanarak en değişken sonucu veren rakama ulaştık ve bunu kendi rastgele fonksiyonumuzda kullandık. Ödevin java kısmında hiçbir yerini eksik bırakmadık. Tam olarak bizden istenilen görevi ödev tamamiyle yapıyor. Ödevin 2. Kısmı yani C dilinde yazma kısmı öncelikle lablarda yapılan nesne yönelimli programlamayı yazdığımız dosyalara uyarladık ardından java'da kullandığımız sistem zamanını kullanma yöntemini C dilinde çalıştırmayı denedik fakat yapamadık. Sistem zamanını ve rand() fonksiyonunu kullanmadan nasıl rastgele sayı üreteceğimizi internetten araştırdık ve LFSR algoritmasına rastladık önce denemeler yaptık ve yeterince farklı sonuçlar ürettiğini düşündük. Ardından kodun analizini yaptık ve bitwise operatörlerini öğrendik. LFSR algoritmasının kodunu iyice anladıktan sonra kendi ödevimizde kullanmaya karar verdik. Bu algoritmayı kullandıktan sonra imei ve tc kontrollerini yaparken char bir değişkene direk int dönüşümü yapıldığında char içinde rakam olsa bile ascii değerini döndürdüğünü anlayamadık bununla biraz süre kaybettik araştırmalar ve denemeler üzerine bu sorunu fark ederek kendimiz asciiToInt isimli bir fonksiyon yazarak bu sorunun önüne geçtik. Ardından char* ile string bir değer döndürünce FILE yapısından bir değişken oluşturunca isimSoyIsim değişkeninin bozulduğunu fark ettik büyük uğraşlar sonucu bunun önüne geçemedik isimSoyIsim Kisi yapısına düzgün geliyordu fakat dosyaya yazmak için dosya açıldıktan sonra isimSoyIsim bozuluyordu ve değişik değerleri dosyaya yazmaya çalışıyordu. Bunun önüne rastgelede oluşturulduğu gibi dosyaya yazmaya karar vererek hallettik. Fakat yinede Kisi yapısının her türlü değişkenine atama yapılıyor bunu ekrana çıktılar olarak kontrol ettik. Ödevin 2. Kısmı yani C kısmı da istenilen şeyi tamamiyle yapıyor.

Programların çalışma süreleri 1000 kişiyi java yaklaşık 4 saniyede C ise yaklaşık 3 saniyede üretiyor. Üretilen kişi sayısı katlandıkça programın çalışma süreside katlanıyor.

NOT

Ödevin her iki kısmında ödevde Kisiler.txt dosyasının nerede temizleneceği belirtilmediği için seçenek 1 seçilmesi durumunda mevcut Kisiler.txt dosyası temizlendi. Yani parça parça kişi oluşturulamaz oluşturulacak kişi sayısı tek 1 seçeneği seçilmesi durumunda direk yazılması gerekiyor. Programların çalışma süreleri 1000 kişide java için yaklaşık 4 saniye C için yaklaşık 3 saniye sürüyor.