

# Dokuwiki Kocaeli Üniversitesi

Betül Bodur

*Bilgisayar Mühendisliği Bölümü*

*Kocaeli Üniversitesi*

210201069

betlbodur86@gmail.com

## I. ÖZET

Bu doküman Programlama Laboratuvarı 1. projesi olan "Dokuwiki Kocaeli Üniversitesi" için oluşturulan algoritmalar açıklanmıştır. Dokümanda problemin tanımı verilip bu problemler hakkında araştırmalar ve yöntemler sunulmuştur. Araştırmalar ve yöntemlerde bahsedilen fonksiyonların ayrıntılı açıklaması bulunmaktadır.

***Index Terms*—TXT , program , klasör , dosya , argüman**

## II. PROBLEM TANIMI

Bu projede bizden istenen bir veri tabanı kullanmadan üniversitede bulunan bölümleri ve bölümlerde okutulan derslerin içeriklerini gösteren bir uygulama oluşturulmasıdır. Bu uygulamada kullanıcı , dosya içeriğinde arama yapabilmeli , etiket ve yetim etiket denilen formattaki kelimeleri bulup bir output.txt dosyasına yazdırabilmeli , etiketlerin adlarını değiştirebilmeli ve bunları bir menüden seçerek yapabilmelidir.

## III. ARAŞTIRMALAR VE YÖNTEMLER

Projede öncelikle klasörlere ve text dosyalarına ulaşılması gerekmektedir. Bu sorunu çözebilmek için "dirent.h" kütüphanesi internet üzerinde araştırılmıştır[8]. Bunun sonucunda programın çalıştığı yerdeki dosyaları görüntüleme işlemi başarıyla sonuçlanmıştır. Ancak programın çalıştığı yerdeki diğer klasöre erişilemediği için sorun hala tam olarak çözülememiştir. Sorunu çözebilmek için dosya yolunun sürekli değişip bir klasördeki ".txt" uzantılı dosyalarını ve klasörleri ayırması gerekmektedir. Bu sorun ise şöyle çözülmüştür ; programın çalıştığı yerdeki klasörleri dirent.h kütüphanesinde iç içe struct yapısından 'DIR' adlı nesne bulunmaktadır. Kütüphane main() fonksiyonuna dahil edildiğinde, bir pointer nesnesi oluşturulmuştur ('struct dirent \*de'). Bu kütüphanenin içinde tanımlanmış DIR nesnesi ile hangi konuma gidileceğini , d-name() ile de aradığımız klasör veya TXT dosyasının adı alınabilmektedir. Alınan klasör ve TXT adları string dizisine atılmıştır. İç içe for döngüsüyle ; programın çalıştığı klasördeki tüm klasör adlarına ve bir

klasörün içindeki tüm TXT dosyalarına erişilmiştir.

Programda Türkçe dil desteği istenmiştir. Bu sorunu IDE üzerinden çözmek için locale.h kütüphanesi dahil edilmiştir. Bu kütüphanenin setlocale(LC-ALL,'Turkish') fonksiyonu kullanılıp , kullanılan IDE olan Code::Blocks'un default encodingi olan 'UTF-8'i Türkçe karakterleri destekleyen encoding olan 'windows-1254' ile değiştirilmiştir. Ancak programın menü kısmından fonksiyona yönlendirilen kod bu fonksiyon içinde veriyi bir yerden alıp kayıt etme veya kullanıcıdan veri alma işlemleri yaparken scanf(),sprintf() gibi fonksiyonlarda çalışmamıştır. Bu çalışmama durumu programın ilerleyişini olumsuz etkilemiştir. Bundan dolayı yeni çözüm yöntemi[5] olarak windows işletim sisteminde 'Ayarlar- Zaman ve Dil - Dil - Yönetici Dil Ayarları - Sistem Yerel Ayarını Değiştir - Beta:Dünya çapında dil desteği için Unicode UTF-8 kullan' çözüm yolunu kullanarak makineyi yeniden başlattığımızda sorun çözülmüştür. Ancak bu sefer locale.h kütüphanesinde bulunan setlocale(LC-ALL,'Turkish') fonksiyonu işlevini kaybetmiştir. Bu yüzden projenin başında dahil edilen locale.h kütüphanesi kaldırılıp sorun tamamen çözüme ulaşmıştır.

Programın çalıştığı yerdeki klasörlere ve ".txt" uzantılı dosyalara erişildiğinde dosyaların içindeki verileri alıp değerlendirebilmek için dosyayı satır satır okunmasına karar verilmiştir. Ve her satırda bulunan veri alındığında , kullanıcının aradığı kelimeyi bulup kullanıcıya aradığı kelime hakkında satır bilgisini ve dosya adını verebilmiştir.

Kullanıcı bir TXT dosyasının ismini değiştirmek istediğinde klasör adını , değiştirmek istediği txt adını ve yeni TXT adını girmek zorundadır. Bu veriler kullanıcıdan menüde alınmaktadır. Veriler alındıktan sonra fonksiyona gönderilmektedir. Fonksiyonun içinde adı değiştirilmek istenen TXT dosyasının yolu ve yeni TXT dosyasının yolu oluşturulmaktadır. Bu iki yol stdio.h kütüphanesindeki rename() fonksiyonuna gönderilerek

dosyanın adı değiştirilmektedir.

Program ; kullanıcının aradığı kelimeyi satır satır ararken , bu satırları bir dizi içinde tutmaktadır . Bu tüm text dosyalarının bulunduğu dizideki etiket formatındaki kelimeleri bulunup bir diziye atılmıştır . Aynı şekilde tüm txt dosyalarının adı bir diziye atılmıştır . Etiket formatlı kelimelerin bulunduğu dizide txt dosyalarının adının bulunduğu diziyi tek tek arıyoruz ve her txt adlı dizide eşleşme olduğunda sayac bir artırılmaktadır . Ve bu fonksiyon en sonunda Output.TXT dosyasına yazılmaktadır.

#### IV. GELİŞTİRME ORTAMI

Bu proje Windows işletim sisteminde , Code::Blocks version 20.03 üzerinde geliştirilmiştir . 'gcc 8.1.0' ile derlenmiştir.

#### V. KOD BİLGİSİ

Programda oluşabilecek herhangi bir hata sonrasında , kodu düzeltmek uğruna kodun tamamını karıştırmamak için her görev ayrı bir fonksiyona bölünmüştür . Bu fonksiyonların görevleri şu şekilde özetlenmiştir:

##### **void main():**

Kodun başlangıç fonksiyonudur . Bu kısımda global pointer dizileri ve menü bulunmaktadır . Kullanıcının kendi isteğiyle programdan çıkabilmesi ve uygulamada kesinti olmadan işlem yapabilmesi için while() döngüsü bulunmaktadır. While() döngüsünün içinde kullanıcıdan yapmak istediği seçeneği scanf() fonksiyonu ile aldıktan sonra if() , else if() ve else koşullu ifadeleriyle kullanıcıdan alınan işlem uygulanmaktadır.



Fig. 1. Menü Görüntüsü

##### **void dosyalar():**

Programın çalıştığı klasördeki , klasörlerin adının alınmasını

sağlamaktadır.Argüman olarak klasörlerin adını tutması için dizinin adresini ve kaç tane klasör olduğunu bulmamız için integer bir adres almaktadır.

##### **void textler():**

Hangi klasör adı argüman olarak geldiyse o klasördeki TXT uzantılı dosyaların adlarını almayı sağlamaktadır. Argüman olarak TXT uzantılı dosyaların adını tutması için dizinin adresini , hangi klasördeki TXT uzantılı dosyaların bulunacağını belirten klasörün adresi ve kaç tane TXT uzantılı dosya olduğunu bulmamız için integer bir adres almaktadır.

##### **yetim-etiket-d():**

Kullanıcının girdiği yetim etiket adında bir TXT dosyası açıp içine yazmaktadır . Yani bu fonksiyon yetim etiketler istenen etikete çevirmek için kullanılmaktadır . Argüman olarak yetim etiket adını taşıyan dizi ve yeni açılan TXT dosyasının içindeki ders kodunu sürekli artırmak için bir integer pointer almaktadır.

##### **void kelime-arama():**

Kullanıcının aramak istediği kelimenin hangi klasör ? Hangi text dosyasında ? Kaçınca indexte ? Ve etiket formatında mı? Sorularına cevap veren bir fonksiyondur.Argüman olarak satır satır okuduğu TXT uzantılı dosyalarından , bir satırı almak için bir string dizisinin adresini , kullanıcının aradığı kelimenin bulunduğu dizinin adresini ve kelimenin txt içinde olup olmadığını kontrol edecek bir integer adres almaktadır . Kullanıcının aradığı kelimeyi ararken bu kelimenin etiket olup olmadığını ve eğer bir kelimenin içinde geçiyorsa bu bilgileri if() , else if () ve else koşullu ifadeleri kullanılarak verilmektedir.Kelime arama fonksiyonun akış şeması aşağıdaki gibidir.

##### **void c-open():**

Klasörlerin içindeki TXT dosyalarının içinde gezilip , onların içinde aranan kelimeyi bulmamızı sağlamaktadır. Argüman olarak klasörlerin adını tutması , txtlerin adını tutması , txtlerde bulunan tüm yazıları tutması ve arama yapmak istediğimiz kelimeyi tutması için dizilerin adresini almaktadır. Klasörlerin adını alması için dosyalar() fonksiyonu bulunmaktadır . İç içe olmak üzere iki for() döngüsü kullanılmıştır . En dışarıdaki for() döngüsü klasörler adlarının bulunduğu dizinin içinde gezinmeyi sağlamaktadır . İçerideki for() döngüsü ise TXT uzantılı dosyaların adlarının tutulduğu dizinin içinde gezinmeyi sağlamaktadır . Bu iki for() döngüsü içinde kelime aramamız gereken dosyaların yolunu oluşturmamızı sağlayacak fonksiyonlar bulunmaktadır.

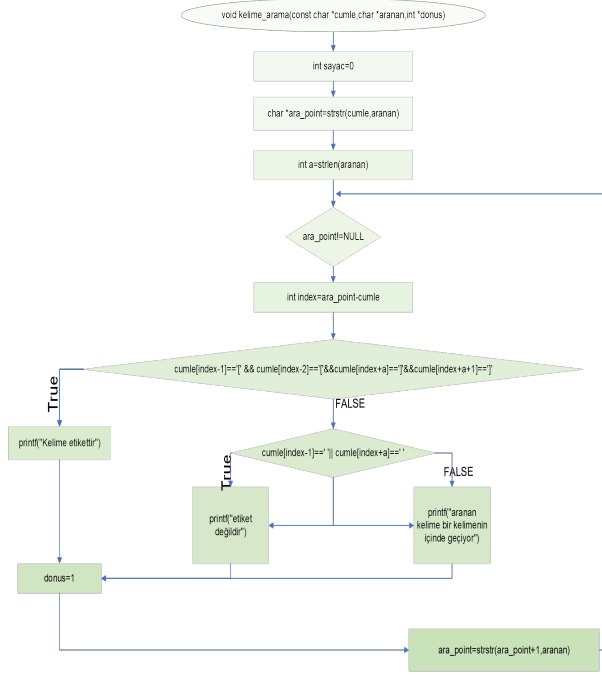


Fig. 2. Kelime Arama Fonksiyonu Akış Şeması

Bu fonksiyonlar sprintf() ve strcat() fonksiyonlarıdır . Dosya yolu oluşturulduktan sonra fopen() ile dosyanın açılması gerekmektedir bunun kontrolü için if ve else kullanılmıştır . Aşağıdaki şemadaki gibidir.

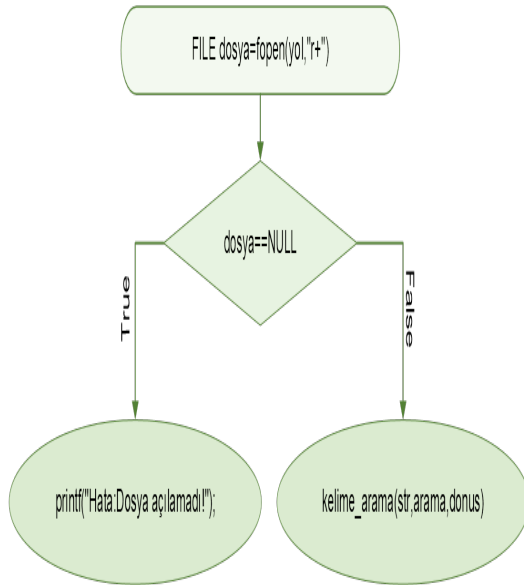


Fig. 3. Dosya'nın Açılıp Açılmaması Durumu Algoritması

Eğer dosya yolu girilmiş dosya açılırsa kelime-arama() fonksiyonu çalışmaktadır . Aranacak kelime ile ilgili bilgileri

kelime-arama() algoritmasından gelecek bilgiler ile burada çıktı olarak vermektedir.

### void dosya-adi-degistirme():

Kullanıcı istediği bir text dosyasının adını değiştirmesini sağlamaktadır . Argüman olarak kullanıcıdan alınan değişecek text dosyasının adını , değişecek text dosyasının bulunduğu klasör adını ve yeni text dosyasının adını almaktadır . stdio.h kütüphanesinde bulunan rename() fonksiyonunu kullanarak mevcut text dosyasının adı değiştirilmektedir.

## VI. YALANCI KOD

- 1-Başla.
- 2-Menüde çıkan seçeneklerden birini seç.
- 3-Kelime araması yapmak için '1' tuşuna bas.
- 4- Aranacak kelimeyi gir.
- 5-'q'tuşuna bas.
- 6-Menüde çıkan seçeneklerden birini seç.
- 7-Etiket adı değiştirmek için '2' tuşuna bas.
- 8-Değiştirmek istediğin txt dosyasının bulunduğu klasörün adını gir.
- 9-Değiştirmek istediğin txt dosyasının adını gir.
- 10-Değiştirmek istediğin txt dosyasının yeni adını gir.
- 11-Menüde çıkan seçeneklerden birini seç.
- 12-Çıktı dosyasını görüntülemek ve yenilemek için '3' tuşuna bas.
- 13-'q'tuşuna bas.
- 14-Menüde çıkan seçeneklerden birini seç.
- 15-Programdan çıkış yapmak için '6' tuşuna bas.
- 16-Bitiir.

## VII. İSTATİSTİK

Program kodu toplam 484 satırdan oluşmaktadır.

Kullanılan kütüphaneler ve kullanım amaçları:

### stdio.h

Kullanıcıdan bilgi alma ve kullanıcıya bilgi vermek

### stdlib.h

Dinamik bellek tahsisi yapabilmek için kullanılmıştır.

### string.h

Bellekte yer açmak için ,belli bir noktadan başlayıp belli bir boyutta kopyalama işlemi yapmak,bir diziyi diğerine kopyalamak ,string içinde sting arama ve string içinde char karakter arama yapabilmek için kullanılmıştır.

### dirent.h

Klasörleri ve txt uzantılı dosyaları bulmak için kullanılmıştır.

## VIII. SONUÇLAR

Sonuç olarak kullanıcı menüden seçtiği işlemler ile kelime araması yaptı , aranan kelime etiket mi değil mi yoksa sadece bir kelimenin içinde mi geçtiğini öğrendi.Txt dosyasının adını değiştirdi.

## REFERENCES

- [1] <https://www.bilgigunlugum.net/prog/cprog>
- [2] Yorulmaz M.,Yorulmaz S., “Programlamayı C ile Öğreniyorum”, Palme Yayınevi, 2018.
- [3] <https://www.tutorialspoint.com>
- [4] <https://www.programiz.com/c-programming/c-structure-function>
- [5] <https://stackoverflow.com/questions/56419639/what-does-beta-use-unicode-utf-8-for-worldwide-language-support-actually-do>
- [6] <https://www.geeksforgeeks.org/extract-substrings-between-any-pair-of-delimiters/>
- [7] <https://codeforwin.org/2018/03/c-program-rename-a-file-using-rename-function.html>
- [8] <https://www.ibm.com/docs/tr/aix/7.3?topic=files-direnth-file>