

PROGRAMLAMA LABORATUVARI PROJE I

DOKUWİKİ KOCAELİ ÜNİVERSİTESİ

Tayyib OKUR
200202060

Kocaeli Üniversitesi Bilgisayar Mühendisliği İÖ
ultratayyib@gmail.com

Yusuf GÖKSU
210202027

Kocaeli Üniversitesi Bilgisayar Mühendisliği İÖ
goxuyusuf@gmail.com

Index Terms—Dosyalar, dirent.h, linux, string

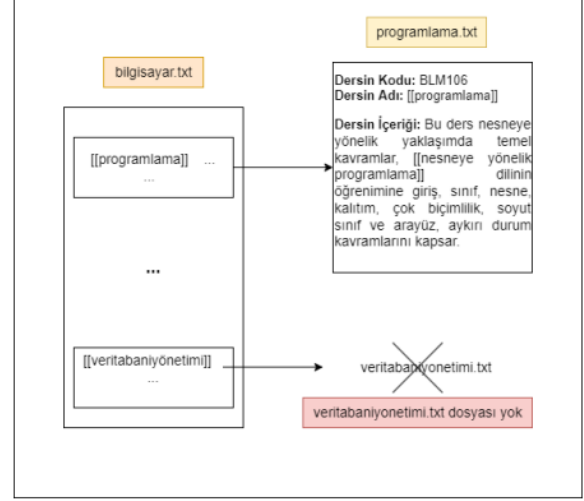
I. ÖZET

Bu rapor belgesi 2022-2023 Güz dönemi BLM209 programlama laboratuvarı I 1. projeyi açıklamaya yönelik bilgiler içermektedir. Projenin ismi Dokuwiki Kocaeli Üniversitesidir. Programlama Lab. I 1. proje dosyaya yazma, dosya güncelleme ve dosya okuma becerilerini güçlendirmeye yöneliktir. Projenin hedefiyse bir veri tabanı kullanmadan üniversitede bulunan bölümleri ve bölümlerde okutulan derslerin içeriklerini gösteren bir uygulama oluşturulması hedeflenmektedir. Bu raporda proje tanıtımı, araştırmalar ve yöntemi, tasarım, deneysel sonuçlar, sonuç ve kaynakça kısımları bulunmaktadır. Bunlara ek olarak kodun akış şeması ve yalancı kodu bulunmaktadır.

II. PROJE TANITIM

2022-2023 Güz dönemi Programlama Lab. I 1. proje dosyaya yazma, dosya güncelleme ve dosya okuma becerilerini geliştirmeye yöneliktir. Bu işlemler de arama, güncelleme ve dosyaya yazma başlıkları altında yapılmaktadır. Bunun dışında menü (arayüz) oluşturulması istenmektedir. Projede içi içe klasörlerden oluşan bir üniversite adlı ana klasör bulunmaktadır. Üniversite adlı klasörün altında en az iki klasör vardır ve bunların altında çeşitli uzantıda olan dosyalar bulunmaktadır (.txt, .jpg, .thmx, .docx). Bizden istenen bu etiketler ile çeşitli işlemler (yazma,güncelleme) yapmaktır.

Her bir dersin bir kodu, adı ve içeriği olmalıdır. Tüm dersler bu şablona uygun olmalıdır.



Şekil 1: Örnek şablon

Örneğin bir ders şablonu Şekil 1'deki programlama.txt şeklinde olmalıdır. Arama adı altında kullanıcı, kelime ve etiket araması yapabilmelidir. Eğer aranan kelime bir etiketse etiket olduğu belirtilmelidir. Kelimenin hangi dosyada ve hangi satırda geçtiği belirtilmelidir. Ayrıca arama kısmında yetim etiketler ve istenen etiketler ekrana yazdırılmalıdır. (istenen etiket: dosyası var fakat etiket yok), (yetim etiket: etiket var fakat dosyası yok). Güncelleme kısmında ise etiket ismi değişikliği bütün dosyalarda etkili olmak koşulu ile veya yetim etiket dosyası oluşturma işlemi yapılabilir. Eğer yetim etiketin dosyası oluşturulursa bu etiket yetim etiket listesinden çıkartılmalıdır. Dosyaya yazma kısmında ise output.txt adında yeni bir dosya oluşturarak tüm etiketler, etiketlerin sayısı ve yetim etiketler yazdırılmalıdır. Eğer kullanıcı output.txt 'de değişiklik yaparsa ve tekrardan dosyaya yazma işlemini uygularsa bize güncellenmiş liste olarak tekrardan tüm etiketler, bunların adetleri ve yetim etiketlerin isimleri o dosyaya yazdırılmış olmalıdır.

III. ARAŞTIRMALAR VE YÖNTEM

A. ARAŞTIRMALAR

Projeyi ilk okumaya başladığımızda kafamızda birtakım fikirler oluşmaya başladı. İsmi bilmediğimiz bir klasörün nasıl açılacağı konusunda bir fikrimiz yoktu ancak internet üzerinde yaptığımız detaylı araştırmalar sonucu C'de bunun

en sağlam çözümünün `dirent.h` kütüphanesini kullanmak olduğunu öğrendik. Bu kütüphaneyi ilk defa gördüğümüz için iyice araştırmaya koyulduk ve `opendir`, `readdir` ve `dirent`'in struct yapısını kullanmayı öğrendik. `Readdir` fonksiyonunun dosyalardaki gibi statik bir şekilde çalıştığını anladık. Dosyalarda statik çalışma nedir diyecek olursak; fonksiyonun statik olması kendi kendisinin kullanıldıktan kaldığı yerden devam etmesidir. Mesela `fgets` fonksiyonunu bir sefer kullandık, eğer ki `fgets` fonksiyonu `txt` dosyamızın 1. satırının tamamını aldığını varsayarsak, eğer bir defa daha çağırırsam `fgets` fonksiyonunu bu sefer okumaya kaldığı yerden devam edecek ve böylelikle 2. satırdan okumaya devam edecek. Bu böyle böyle sürerek en sonunda `EOF`'yi görene kadar ya da `fgets` fonksiyonunun istediğimiz String dizisine `NULL` döndürdüğünde anlıyoruz ki artık okunacak bir satır kalmamış bir sonraki `txt` dosyasına geçmeliyiz ya da işimiz eğer ki dosyayla bitmediyse `rewind()` fonksiyonuyla o daha demin bahsetmiş olduğum statik olarak dosyanın sonuna gelen hayali imlecimiz, artık `txt` dosyamızın en başını gösterecek yani en başa gelmiş olacak. `Rewind()` fonksiyonu `fseek(dosya-ismi, 0, SEEK-SET)` ile aynı mantıkla çalışmakta ancak biz `rewind()` fonksiyonunu kullanmayı tercih ettik. Dosyaların isimlerine böylece erişim sağlayabildik. (`DİPNOT` : '-' 'ler alttan '-' olmalı latex izin vermediği için öyle yazdık)

B. YÖNTEM

Yönteme gelecek olursak: İlk öncelikle `dirent.h` kütüphanesini kullanarak Üniversite adlı klasörümün altındaki klasörlerimin isimlerini bir string dizisinde tuttum. Böylece elimde açabileceğim bütün klasör isimlerini elde ettik. Onların içerisini de açarak içerindeki ".txt" uzantılı dosyaları alarak başka bir stringte tuttuk. Daha sonra o `txt` dosyalarını tek tek açarak içerindeki etiket formatında olan etiketleri yine bir string dizimide tuttum. Bu elde ettiğim etiketleri tekrar etmeyecek bir forma dönüştürdük. Bunları etiket-tekrarsız adlı stringimize attık. Bir arayüz oluşturmak için menü fonksiyonu oluşturduk. Bu fonksiyonu döngüyle switch-case yapısına bağlı bir şekle soktuk.

Eğer kullanıcı '1' tuşuna basarsa dosyaların içeriğini gösteren fonksiyona, `txt` dosya adlarını alan stringi ve `txt` dosya sayısını yolladık. Bu fonksiyon bize tüm `txt` dosyalarını ekrana yazdırıyor ve bir seçim istiyor, bu seçime göre seçilen dosya açılıyor ve içeriği `fgets()` ile yeni oluşturduğumuz boş bir string dizimize atarak ondan sonra o string dizimizin içeriğini ekrana yazdırıyoruz. Böyle böyle bütün satırları ekrana yazdırıyoruz. Ancak burada bizim yazış tarzımızdan dolayı bir sorun ile karşılaştık. Bu sorun eğer ki son satır boşsa `fgets()` fonksiyonu ile boş satırı oluşturduğumuz string dizimize atmıyordu. Atmamasından dolayı bizim string dizimizin içinde en son kalan cümle bir defa daha ekrana yazdırılıyordu. Bunu önlemek için her ekrana string dizimizi yazdırdıktan sonra dizimizin içeriğine `strcpy` ile `NULL` attık. `Memset` ile de kolayca boşaltılabilirdi ancak biz uğraşmak istemedik. Aklımıza ilk gelen yöntemi yaptık bir sorun ile karşılaşmadığımız için böyle bir yol tercihinde bulunduk.

Eğer kullanıcı '2' tuşuna basarsa ise `ara()` fonksiyonu çağırılıyor. Aranan kelimeyi kullanıcıdan aldık. Daha sonra tekrarsız etiketlerimizden , etiketlerdeki parantezleri atıp parantezsiz bir şekilde (etiket-isimleri) stringimizde elde ediyoruz. Etiket isimleri ile dosya isimlerini karşılaştırarak yetim etiketleri elde ediyoruz. Daha sonra dosyaların isimlerine karşılık gelen bir etiket yoksa onu bulup istenen etiketlere atıyoruz. Sonra elde ettiğimiz bütün etiketleri (istenen etiket hariç) aranan kelime ile karşılaştırarak aranan kelimenin etiket olup olmadığını ekrana yazdırıyoruz. Dosyalarımızı tek tek açıp satır satır okuyup girilen kelimenin geçtiği satır/satırları ve hangi dosyada geçtiyse ekrana yazdırıyoruz. En son da yetim etiket ve istenen etiketlerin hepsini güncel bir şekilde ekrana yazdırıyoruz.

'3' tuşuna kullanıcı tarafından menüde basılırsa güncelle adlı fonksiyonumuz devreye giriyor. Yine etiket isimlerini, parantezlerini atarak stringimizi elde ediyoruz. yetim etiketleri elde ediyoruz. Kullanıcıya yetim etiket dosyası oluşturma ve etiket güncelleme seçenekleri sunuluyor. Eğer ki kullanıcı etiket değiştirmek isterse '1' tuşuna basıyor ve değiştirmek istediği etiketin adını ve yeni etiketin isminin ne olacağını giriyor. Burada şöyle bir sorun ile karşılaştık. Bu sorun üst üste iki defa `gets()` fonksiyonu kullandığımız için 1. kullandığımız `fgets()` fonksiyonu kendi kendine ekarte oluyordu ve sadece ikinci stringi alıyordu. Sadece gelirse sadece 2. `fgets()` fonksiyonu çalışıyordu. Bunu çözmek için Windows'ta çalışırken `fflush()` adlı fonksiyonu kullandık ve sorun çözüldü ancak linux'ta `fflush()` fonksiyonu çalışmıyordu bu sebepten mütevellit `fflush()` fonksiyonunun kaldırıp onun yerine `getchar()` fonksiyonunu kullandık. İnternette araştırdığımızda gördük ki bu sorunu çözmek için 5 satırlık bir void türünden `cleanstdn` adlı bir fonksiyon yazmamız gerekiyordu. Yazabilirdik ancak yapmadık onun yerine bir sefer olmak suretiyle `getchar()` fonksiyonunu kullandık ve sorunumuz hallolmuş oldu. Kaldığımız yerden devam edelim eğer ki kullanıcı etiket değiştirmek isterse '1' tuşuna basıyor ve değiştirmek istediği etiketin adını ve yeni etiketin isminin ne olacağını giriyor. Biz de bütün etiket listemizle o değiştirilmek istenen etiketin adını karşılaştırıyoruz böyle bir etiket var mı diye kontrol ettiriyoruz. Eğer varsa ve kullanıcı tarafından girilen kelime bir etiketse, ekrana; kelime etikettir ve şu adlı `txt` dosyasında şu numaralı satırlarında geçmektedir diye ekrana yazdırıyoruz. Etiket değiştirme işlemini başka bir geçici `tmp.txt` adlı bir dosya oluşturarak orada değişiklik işlemlerini yaparak değiştirilmek istenen dosyaya direkt aynısını yapıyoruz. Yetim etiketlerin dosyalarını oluştururken de 200'den başlamak suretiyle ders kodlarını oluşturuyoruz ve onun bir artırılmış halini yani 201'i derskodutut adlı bir `txt` dosyasına yazıyoruz. Bu `txt` bize eğer program kapatılıp açılırsa diye en son oluşturulan yetim etiketin ders kodunu tutmamız için bir kova gibi çalışıyor ve biz her seferinde o dosyayı açarak -yani derskodunu tuttuğumuz `txt`'den bahsediyorum- satır satır okuyarak en büyük sayımızı yani en sondaki sayımızı derskodlu adlı integer değişkenimize atıyoruz böylelikle elimizde en son

güncel olan 200'den sonraki yazdırmak istediğimiz yetim etiketin ders kodunu elde etmiş oluyoruz. Böylelikle ders kodlarının hem en son halini tutmuş oluyoruz hem de programın kapatılması ihtimaline karşılık yaşanabilecek sorunun önüne geçmiş oluyoruz. Tüm bunların akabinde dersin adını etiket formatında yazıyoruz son olarak da bize söylenildiği şekilde "Ders içeriği" kısmını boş bırakıyoruz. Çünkü oluşturduğumuz yetim etiketin ismini bilmiyoruz, her oluşturduğumuz yetim etiketin ders içeriğini boş bırakmak en mantıklısı.

Kullanıcı eğer '4' rakamını girerse dosyaya-yaz isimli fonksiyonumuz çalışmaktadır. Üniversite adlı ana klasörümüzde output.txt adlı dosyamızı read yani okuma modunda açmaya çalışıyoruz çünkü bu dosya önceden var olmuş olabilir, var olmamış da olabilir. Bunu yapmamızın sebebi ise pdf'te verilen "Güncel liste" halini yazdırmak istememiz. Eğer ki dosya read modunda açılmadıysa bu demek oluyor ki dosya yoktur ve içinde hiçbir şey yazmıyordur. Bu da demek oluyor ki dosyamızı oluşturmamız lazım ve "Güncel Liste" yazmamamız lazım. Bu kontrolü yapmasaydık her seferinde "Güncel Liste" yazdıracaktı ve pdf ile çelişki içerisinde bulunacaktık. Eğer output.txt dosyası var ise "a" addition yani ekleme modunda açıyoruz ve en son yazılan output.txt nin üzerine eklemelerimizi "Güncel Liste" şeklinde yapıyoruz. Output.txt 'yi nasıl oluşturduğumuzu ya da girdiğimizi anladık, peki bunun içine ne yazıyoruz? bunun içine elimizdeki bütün etiketlerin ismini ve karşılıklarına kaç adet olduklarını yazıyoruz. Sonra da bunun altına yetim etiketlerin ismini alt alta yazdırıyoruz. Eğer ki kullanıcı bu 4 seçenekten farklı bir rakam-sayı ya da bir şey girerse "Hatalı Seçim" yazısı ekrana bastırılıyor.

IV. AKIŞ ŞEMALARI VE TASARIM

A. Akış şemaları eklerdedir:

Genel Akış [1] (genel.pdf)
İstenen Etiket [2] (istenen.pdf)
Yetim Etiket [3] (yetim.pdf)

B. Tasarım

Tasarımımız temelinde menü yatmaktadır. Bu menüler yukarıda belirttiğimiz üzere dallara ayrılmaktadır. İlk başta tüm etiketler sırasıyla ekrana yazdırılır ve 1-Dosyaya yazdır, 2-Ara fonksiyonu, 3-Güncelleme, 4- Dosyaya yaz olmak üzere menümüz oluşmaktadır. 1 tuşuna basarsa ekrana bütün txt dosyaları geliyor sıralı bir halde. Yanındaki sayıyı tuşlayarak o açmak istediğimiz txt açılıyor ve ekrana yazdırılıyor içinde ne var ise. 2 tuşuna basarsak ara fonksiyonu devreye giriyor ve aramak istediğiniz kelimeyi giriniz diye yazdırılıyor ekrana. Eğer ki girilen kelime etiket ise etiket formatında olmasa dahi kelime etikettir diye ekrana yazdırılıyor ve hangi satır satırlarda geçtiği ekrana bastırılıyor. 3 tuşuna basarsak güncelle fonksiyonumuz 2' ye ayrılıyor ve etiket ismi mi güncellemek istiyorsunuz yoksa yetim etiketin txt dosyasını mı oluşturmak istiyorsunuz diye ekrana yazdırılıyor. 4 e basarsak dosyaya yazma işlemi yani output.txt yoksa

oluşuyor varsa üzerine ekleyerek tüm etiketlerin listesi ve yetim etiketler yazdırılıyor.

```
Tüm Etiketler
1. Etiket [[Matematik_I]]
2. Etiket [[Programlama_I]]
3. Etiket [[Matematik_II]]
4. Etiket [[Programlama_II]]
5. Etiket [[Görsel Programlama]]
6. Etiket [[VeritabanıYönetimi]]
7. Etiket [[Ahmet ERDİL]]
8. Etiket [[Alev MUTLU]]
9. Etiket [[Matematik_I]]
10. Etiket [[Yusuf GÖKSU]]
11. Etiket [[Programlama_I]]
12. Etiket [[ProgramlamaLab]]

1-Dosya içeriği göster
2-Arama
3-Güncelleme
4-Dosyaya yaz
0-Çıkış
Lütfen [1-4] bir seçim yapın
1
Dosya adları:
1-Bilgisayar Mühendisliği.txt
2-Endüstri Mühendisliği.txt
3-Kimya Mühendisliği.txt
4-Yazılım Mühendisliği.txt
5-Çevre Mühendisliği.txt
6-Ahmet ERDİL.txt
7-Alev MUTLU.txt
8-Matematik I.txt
9-Matematik I.txt
10-Matematik II.txt
11-NesneyeYönelikProgramlama.txt
12-Programlama_I.txt
13-Programlama_II.txt
14-Tarih.txt
15-Yusuf GÖKSU.txt
16-Burak Inner.txt

Lütfen içini gormek istediğiniz dosyayı seçiniz (1-16):
```

(Ubuntu üzerinden programın çalışması)

V. DENEYSEL SONUÇLAR

```
1-Dosya içeriği göster
2-Arama
3-Güncelleme
4-Dosyaya yaz
0-Çıkış
Lütfen [1-4] bir seçim yapın
2
Aranan kelimeyi girin: Alev MUTLU_
```

(Arama fonksiyonumuzda kelime araması)

```
Aranan kelimeyi girin: Alev MUTLU
Aranan kelime etikettir
Doküman/Universite/Dersler/Alev MUTLU.txt dosyasında:
kelimenin geçtiği cümle ---->>> Dersin Adı : [[Alev MUTLU]]

Doküman/Universite/Dersler/NesneyeYönelikProgramlama.txt dosyasında:
kelimenin geçtiği cümle ---->>> Dr. Öğr. Üyesi [[Alev MUTLU]] alev.mutlu@kocaeli.edu.tr 02623803577

Doküman/Universite/Dersler/Programlama_I.txt dosyasında:
kelimenin geçtiği cümle ---->>> Dr. Öğr. Üyesi [[Alev MUTLU]] alev.mutlu@kocaeli.edu.tr 02623803577
```

(Aranan kelimenin etiket olma durumu, hangi txt dosyasında hangi satırlarda geçtiği gösterilmiştir.)

```

Dokuwi/Universite/Dersler/Tarih.txt dosyasında;
kelimenin geçtiği cümle --->> Öğretim Elemanları : Prof. Dr. [[Ahmet ERDİL]]

*****İSTENEN ETİKETLER***

Bilgisayar Mühendisliği
Endüstri Mühendisliği
Kimya Mühendisliği
Yazılım Mühendisliği
Çevre Mühendisliği
MesneYönelikProgramlama
Tarih
Burak İnnar

*****YETİM ETİKETLER*****

Görsel Programlama
VeritabanıYönetimi
Ahmet ERDİL
Matematik I
Ahmet SAYAR
Programlama I
Alev MUTLU
ProgramlamaLab

```

(Arama fonksiyonunda bizden istenilen, yetim ve istenen etiketlerin yazdırılması işlemi)

TABLE I
GEÇERLİ ETİKET TABLOSU

| Etiket Doğruluk Tablosu | | | |
|-------------------------|-----------------------|--------------|-----------------|
| 1. etiket | 2. etiket olma koşulu | 1 İçin Örnek | 2 İçin Örnek |
| [[kelime]] | [[kelime1 kelime2]] | [[Tayyib]] | [[Yusuf Göksu]] |

VI. SONUÇLAR

Bu projemizde C’de dosyaların nasıl açılacağı, nasıl işleneceği konusunda önemli bilgiler öğrendik. Geçen sene derslerde gördüğümüz string fonksiyonlarını bu projemizde çokça kullandık bu sebepten fonksiyonlara iyice hakim olduk. Türkçe karakterlerin Windows’ta sıkıntı yaratmasından ötürü diğer işletim sistemlerine yönelme fikri doğdu. Biz hem Ubuntu üzerinden hem de Kali Linux üzerinden kodumuzu çalıştırdık ve güzel sonuçlar aldık. Türkçe karakter sorununuz tamamen düzeldi. Böylelikle Linux işletim sisteminin kurulumunu öğrenmiş olduk. Linux’ta temel komutları (cd, gcc, sudo, apt, apt-get, passwd vb.) öğrenmiş olduk. Aslında windows terminalinde yani cmd’de sorun olduğunu öğrendik. Bu sorunu çözmek için terminalin dilini değiştirebiliriz ancak değiştirirsek ileride daha fazla hata ile karşılaşmayı göze almamak için başka bir işletim sistemi kullanmak daha sağlıklı geldi. Kullandığımız terminali de değiştirebildik örneğin MSYS kullanarak da farklı işletim sistemi kurmadan sorununuzu halledebildik ancak her halükarda linux komutlarını öğrenmek durumunda kalacaktık ve öğrendik.

GELİŞTİRME ORTAMI

Kodumuzu C üzerinden yazdık. İlk başta kodumuzu yazmaya Codeblocks üzerinden başladık ancak Codeblocks’ta dosyaları açarken Türkçe karakter sorunu yaşadığımız için Devc++’ye geçiş yaptık. Kodumuzun tamamını orada yazdık lakin bu sefer dosyalarda Türkçe karakterleri yazdırırken cmd’de sorunla karşılaştık bu sebepten Linux’ta sunum yapmaya

karar verdik. Linux işletim sisteminde ise Ubuntu ve Kali Linux’u tercih ettik, ikisinde de sorunsuz çalışmaktadır.(Ancak Winodws’ta dosya izinlerine erişirken 2 tane ters ‘/’ kullanıyoruz lakin linux’ta sadece ‘/’ kullanmak durumundayız.) Kullandığımız kütüphaneler:

- stdio.h
- stdlib.h
- string.h
- dirent.h

KAYNAKÇA

Dirent.h kullanımı için kullanılan site [1].
Dirent.h kullanımı için gerekli dir açma fonksiyonu [2]
Dirent.h kullanımı için gerekli dir okuma fonksiyonu [3]
Çeşitli C’ de string üzerinde işlem yapmak için kullandığımız fonksiyonların açıklamasını yapan internet sitesi [4]
Strstr,strcmp,memcpy,strcpy,strncpy,atoi fonksiyonları için [5]

REFERENCES

- [1] <https://man7.org/linux/man-pages/man0/dirent.h.0p.html>
- [2] <https://pubs.opengroup.org/onlinepubs/009604599/functions/ opendir.html>
- [3] <https://pubs.opengroup.org/onlinepubs/7908799/xsh/readdir.html>
- [4] <https://www.bilgigunlugum.net/>
- [5] Muhammet Y. , Seher Y.(2018) PROGRAMLAMAYI C İLE ÖĞRENİYORUM, (7.basım), Ankara, Palme Yayınevi.