

Word With Linked List

Ferhat AÇIKALIN
Berkay Yasin ÇİFTÇİ

180202065@kocaeli.edu.tr
190202056@kocaeli.edu.tr

1.1. Problem Tanımı:

Programlama Laboratuvarı 1 projesi olarak bizden “ Bağlı Liste ile Kelime Sayma” adında bir proje yapmamız istenmiştir.. Bu projede bir txt dosyasından okunan metindeki kelimeleri ve karakterleri her seferinde yeni bir kelimeye göre metin içerisinde arama yaparak ve metin içerisindeki kelimeler sayılarak oluşturacağımız bağlı listeye adet olarak büyükten küçüğe olacak şekilde eleman ekleyerek sıralanması istenmiştir. Bağlı liste oluşturduktan sonra bağlı listenin başına , sonuna, araya ekleme yapılması istenmiştir. Bu projede C programlama dilini kullanılmamız istenmiştir.

1.2. Yapılan Araştırmalar:

Proje geliştirirken karşılaşılan sorunlardan birkaçı aşağıda belirtilmiştir ve bu sorunları nasıl aştığımız aşağıda detaylı olarak anlatılmıştır.

Birinci olarak karşılaşılan sorun bağlı listeye başına, sonuna, arasına ekleme yapılması istenmiştir. Biz başına ve sona eklemeyi daha önceden biliyorduk ama araya eklemeyi bilmiyorduk , bunun için internette birkaç web sitesinden yaralandık. Gerekli bilgiler edinildikten

sonra başa ,sona ve araya ekleme işlemleri gerçekleştirebiliyor olduk .

İkinci olarak karşılaşılan sorun ise txt dosyasının içindeki metindeki kelimeleri nasıl okuyacağımız hakkındaydı. Bu sorunumuzu çözmek için kısa bir araştırma yaptıktan sonra fopen() fonksiyonu kullanarak dosya okuma işlemi yapacağımızı öğrendik.

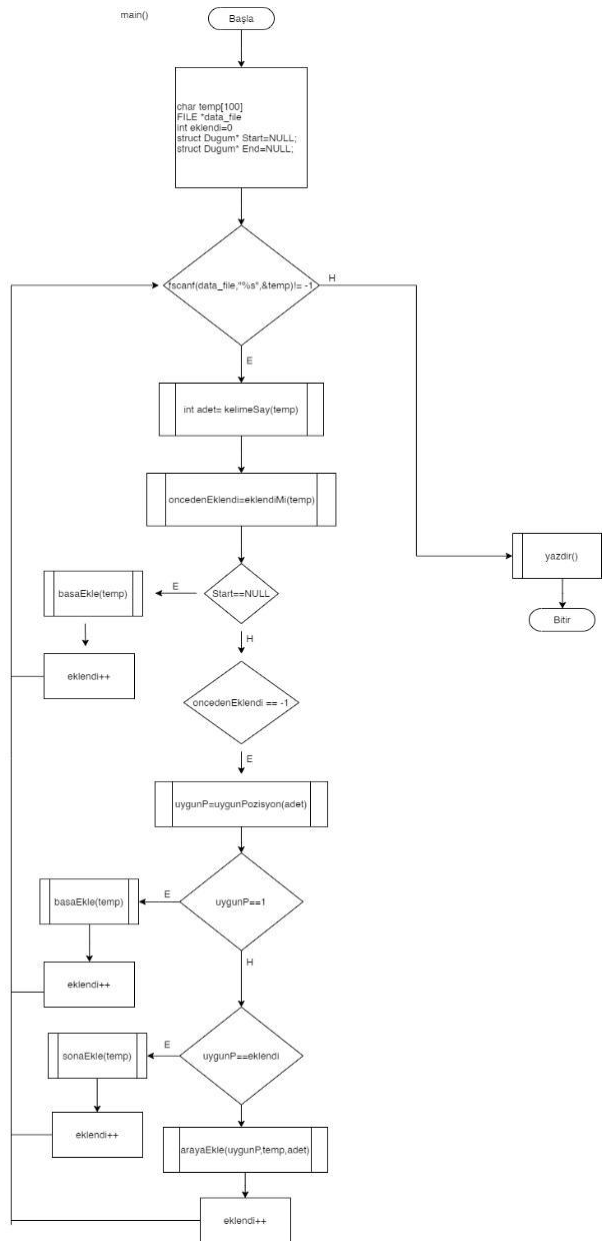
Üçüncü olarak karşılaşılan sorun ise okunan kelimeler ve karakterler adet sayısını nasıl hesaplayacağımız hakkındaydı. Bunun içinde bir değişken tutarak metin içerisinde arama yaptırdıktan sonra belirlediğimiz değişkeni artırıyor ve böylelikle karakter ve kelimelerin adetini de nasıl tutulacağını öğrendik.

Diğer bir sorun ise, kullandığımız işletim sistemi, ide ve terminalin encodinglerinin farklı olması nedeniyle Türkçe karakterleri ekrana düzgün yazdıramıyorduk. Code::Blocks idesinin default encodingi UTF-8 , metin dosyanın encodingi UTF-8 fakat sorun şu ki, Windows komut istemi karakter seti olarak windows-1254 kullanmakta bu yüzden encoding sorunu meydana geliyordu. Program doğru çalışsa da bunu kullanıcıya doğru aktarma

konusunda sıkıntı yaşıyorduk. Bu sorunu çözmek için araştırma yaptık ve locale.h kütüphanesini kullanarak programın karakter setini değiştirebileceğimizi öğrendik.^[1] setlocale(LC_ALL, "Turkish") şeklinde türkçe karakter uyumluluğu kazandırdık ve artık encoding windows-1254 oldu. Daha sonra data.txt dosyasını windows-1254 encoding olacak şekilde tekrar oluşturduk.

1.3. Tasarım:

1.3.1. Akış Şeması:



1.3.2. Yazılım Mimarisi:

Programda gerekli görevler için ayrı ayrı fonksiyonlar tanımlandı. Dosyadan okunan veriler bağlı listede tutuldu. Kelimeler sayıldıktan sonra uygun konuma eklenmesi için bağlı liste içinde dolaşarak uygun konum bulundu. Tanımlanan fonksiyonlar ve görevleri aşağıda özetlenmiştir.

Void BasaEkle:

Bağlı listede başa eleman eklemek için kullandığımız return etmeyen bir fonksiyonumuzdur.

yeni_dugum adında ve Dugum tipinde Yeni düğüm için bellekten alan ayrılır. strcpy(); fonksiyonu ile karakter dizisini datası yeni_dugum e kopyalanır. Kullanıcıdan alınan parametreyi dugum tipinde oluşturduğumuz yeni düğüm için adet değeri olarak atanır. Yeni düğümün next değeri start elemanı olarak atandıktan sonra yeni düğüm start elemanı olarak güncellenir. Ve böylelikle başa ekleme işlemimiz son bulmuş olur.

Void SonaEkle:

Bağlı listede sona eleman eklemek için kullandığımız return etmeyen bir fonksiyonumuzdur. Bu fonksiyonumuzda traverse işlemi uygulanmıştır. Traverse işlemi aradığımız düğümü bulmak için ya da düğümdeki değere eşit mi olduğunu bulmak için kullanılır. Bu kullandığımız Traverse işlemini kısaca anlatmak gerekirse: ilk düğüm geçici düğümüne eşitledikten sonra işaret edilen düğümü buluyor ve geçici düğümü işaret edilen düğümüne eşitliyor. Traverse şartı ise düğümün pointer değişkeninin de null değere eşit olmalıdır.

yeni_dugum adında ve Dugum tipinde Yeni düğüm için bellekten alan ayrılır. strcpy(); fonksiyonu ile karakter dizisini

datası yeni_dugum e kopyalanır. Kullanıcıdan alınan parametreyi dugum tipinde oluşturduğumuz yeni düğüm için adet değeri olarak atanır. Sona ekleme yapılacağı için oluşturduğumuz düğüm hiçbir düğümü yani null değerine eşit olacaktır. Eğer bağlı listede hiç eleman yoksa yeni oluşturduğumuz düğüm start düğümü olarak atanır. Eğer varsa Traverse işlemi uygulanır. End adlı oluşturduğumuz düğüme start adlı düğüm atılır. Sonra End null oluncaya kadar döngü oluşturulur. Ve bu döngü ile bize sondaki düğümü buldurur. Ardından End düğümünün next değerine yeni oluşturduğumuz yeni_dugum düğümü atılır. Yani sona bir tane eleman eklenmiştir. Ve böylelikle sona ekleme işlemimiz son bulmuştur.

Void ArayaEkle:

Bağlı listede araya eleman eklemek için kullandığımız return etmeyen bir fonksiyonumuzdur. Bu fonksiyonumuzda alınan parametrelerden ilki pozisyonu diğer alınan parametre ise eklenecek değeri ifade etmektedir. Verilen ifadelere göre traverse işlemi yapılır.

temp değişkeni adında nulla eşit olan bir düğüm oluşturulur. yeni_dugum adlı bir düğüm oluşturulur. Ve bu yeni_dugum adındaki düğüm için bellekten yer ayrılır. strcpy(); fonksiyonu ile karakter dizisini datası yeni_dugum e kopyalanır. Kullanıcıdan aldığımız adeti yeni_dugum verisi olarak atanır. temp ' e başlangıç düğümü olan Start ı atadık. for döngüsü ile i değeri = 2 den pozisyon değerine eşit oluncaya kadar temp=temp->nex işlemi yapılır eğer temp değişkeni null ise break ile döngü kırılır. Sonra temp - > next değeri =null olmayıncaya kadar temp -> next değeri yeni_dugum-> next e atanır. yeni_dugum ise temp - > next olacak

şekilde atama yapılarak araya ekleme işlemi tamamlanmış olur .

Void Yazdir:

Oluşturduğumuz bağlı listedeki elemanları yazdırmak için kullandığımız return etmeyen bir fonksiyonumuzdur.

Start değerini bozmamak için yazdir adlı düğüme atanır. i değeri 1 olarak belirlenir ve bize kaç farklı karakter olduğu göstermek için döngü içinde kullanılır. Yine Travers işlemi yaprak yazdırma işlemi gerçekleştirilir. Sonra yazdir null oluncaya kadar döngü oluşturulur. Ve yazdir adlı düğümün verisi ve adet ekrana yazdırılır. Yazdir adlı düğüme yazdir next atanır ve i değeri bir artırılır. Bu döngü sondaki elemanı yazdırmadığı için döngüden çıktıktan sonra son elemanı yazdırır.

Int kelimeSay:

Argüman olarak aldığı kelimenin metin içerisinde kaç defa geçtiğini return eder. Dosyadaki kelimeler while döngüsü ile tek tek strcmp() fonksiyonu ile fonksiyonun aldığı parametre ile karşılaştırılır. Eşitlik durumunda kelime_count değişkeni bir arttırılır. Fonksiyon kelime_count değişkenini return eder.

Int eklendiMi:

Argüman olarak aldığı kelimenin bağlı listeye daha önce eklenip eklenmediği kontrol eder. Start düğümünden başlanarak bağlı liste içinde dolaşarak her düğümdeki data ile fonksiyonun aldığı argüman karşılaştırılır. Eşleşme durumunda döngü sona erer. Döngü her döngüsünde i değişkeni bir arttırır. While döngüsü bağlı

listenin sonuna geldiyse eklenmedi demektir. Bu durumda fonksiyon -1 değerini döndürür. Aksi durumda i değişkenini döndürür. Bu i değişkeni bulunduğu konumu da belirtir.

int uygunPozisyon:

Kelimenin bağlı listeye büyükten küçüğe doğru eklenmesi durumunda hangi sıraya eklenmesi gerektiğini döndürür. Fonksiyon kelimenin metin içerisinde kaç defa geçtiğini parametre olarak alır. Bağlı liste içerisinde adet>=(alınan parametre) olduğu sürece ilerlenir ve her ilerlemede i değişkeni bir arttırılır. Son olarak i değişkeni return edilir.

1.4. Genel Yapı:

Kodun çalışma biçimini genel olarak özetlersek;

İlk olarak, Dugum adında struct yapısı tanımlanır bu yapı int adet, char data[100] ve struct Dugum *next pointerını içerir. Adet değişkeni, data da tutulan kelimenin metin içerisinde kaç defa geçtiği tutar. *next pointerı bir sonraki düğümün adresini tutar. Bu düğüm üzerinden sonraki düğüme geçilir.

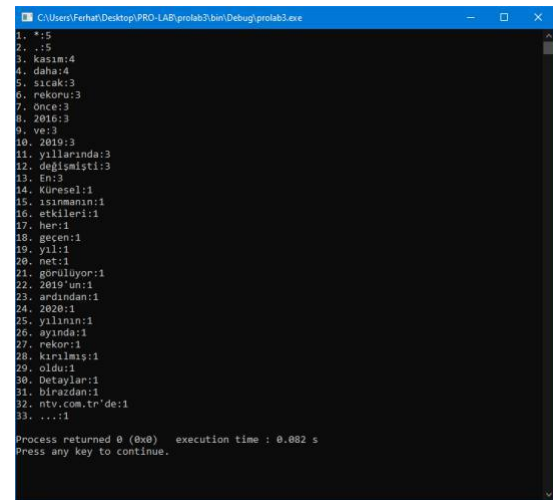
Daha sonra Start ve End düğümleri oluşturulur ve NULL olarak tanımlanır. Start ilk düğüm, End son düğümü belirtir.

Bu işlemlerden sonra main() fonksiyonu çalışmaya başlar. İlk olarak karakter seti kullanılan işletim sistemine uygun hale getirilir.

Daha sonra data.txt dosyasından metin kelime kelime okunur. Her kelimenin kaç defa metinde geçtiği hesaplanır. Ve bağlı listeye ekleme işlemine geçilir. Eğer Start==NULL ise bağlı listenin boş olduğu ve en başa eklenmesi gerektiği için basaEkle fonksiyonu ile en başa eklenir.

Diğer durumda kelimenin bağlı listeye önceden eklenip eklenmediği kontrol edilir, eğer daha önce eklenmediyse uygunPozisyon fonksiyonu ile eklenmesi gereken sıra bulunur. Eğer ilk sıraya eklenmesi gerekiyorsa basaEkle fonksiyonu ile ilk sıraya, son sıraya eklenmesi gerekiyorsa sonaEkle fonksiyonu ile sona ve diğer durumlarda da arayaEkle fonksiyonu ile araya eklenir.

Son olarak yazdir fonksiyonu ile bağlı liste içindeki veriler ekrana yazdırılır.



```
C:\Users\Ferhat\Desktop\PRO-LAB\prolab3\bin\Debug\prolab3.exe
1. 15
2. 15
3. Kasım:4
4. daha:4
5. sırtak:3
6. rekoru:3
7. önce:3
8. 2016:3
9. vez:3
10. 2019:3
11. yıllarında:3
12. değişimisti:3
13. fm:3
14. Küresel:1
15. isinmanın:1
16. etkiler:1
17. her:1
18. geçen:1
19. yıl:1
20. net:1
21. geliyor:1
22. 2019'un:1
23. ardından:1
24. 2020:1
25. yılının:1
26. ayında:1
27. rekor:1
28. karışık:1
29. oldu:1
30. Detaylar:1
31. birazdan:1
32. ntv.com.tr'de:1
33. ...1
Process returned 0 (0x0)   execution time : 0.082 s
Press any key to continue.
```

1.5. Referanslar:

Kitap, çok yazarlı:

Paul Deitel , Harvey Deitel (2015). C İle Programlama. Palme Yayınevi

Web Site:

1.<https://docs.microsoft.com/tr-tr/cpp/c-runtime-library/reference/setlocale-wsetlocale?view=msvc-160> (Access date: 30.12.2020)

<https://codeforwin.org/> (Access date: 01.01.2021)