

BAĞLI LİSTE İLE KELİME SAYMA

Emre SEVİNDİK, Emir AVCI
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

emre.sevindik99@gmail.com, emr9229@gmail.com

Problem Tanımı

Text içerisindeki kelimeleri sayarak bağlı listeye adet olarak büyükten küçüğe doğru eleman ekleme problemini C dilinde yaptık.

Yapılan Araştırmalar

C'de, dosya giriş çıkış işlemlerinin temelini, bilgisayarı oluşturan aygıtlar için mantıksal bir arabirim görevi yapan, **Akış** adı verilen bir kavram oluşturur. Genel bir ifade ile, akış bir dosyaya ulaşmak için kullanabileceğiniz mantıksal bir arabirimdir.

Akış, **stdio.h** başlık dosyası içinde tanımlı **FILE** adlı yapı veri türünden bildirimi yapılan bir işaretçi değişkeni ile kullanılan bir dosya veya fiziksel bir cihazdır.

Bir akışın dosya (fiziksel aygıtlar dahil) ile bağlantısını sağlamak için açma işlemi, dosyadan bağlantısını kesmek için kapama işlemi uygulanır. Dosya açıldıktan sonra, dosyadan programınıza bilgi aktarabilir veya dosya içeriğini değiştirebilirsiniz. Dosyayı açmak için fopen fonksiyonunu kullandık ve dosya.txt adından bir text dosyası oluşturduk. Eğer dosya boşsa fopen fonksiyonu NULL değer döndürecek ve ekrana "Dosya açılmadı." şeklinde bir hata yazdıracaktır.

Eğer dosya açılırsa fgets fonksiyonu NULL dönmediği sürece dosyanın içindeki cümle okunacaktır. Char tipinde tanımladığımız tokena, strtok yardımıyla cümledeki boşluk virgöl ve alt satıra geçme işlemlerinden önceki kelimeler atılacaktır. Yani token, strtok fonksiyonu yardımı ile text dosyasındaki kelimeleri tutacaktır. Text dosyasındaki ilk kelime token değişkeninde tutulur. Bu sayede token NULL değer döndürmez ve while döngüsü içinde kelime_ekle() fonksiyonu

yardımları ile kelime, bağlı listenin ilk düğümüne eklenir.

Bağlı liste, düğüm ismi verilen bir elemanda veri saklar. Yeni veri ekleneceğinde bir düğüm oluşturulur ve listeye bağlanır. Eleman ekleneceğinde bir düğüm oluşturulur ve listeye bağlanır. Eleman silineceğinde ise düğümlerden biri silinir, kopan bağ tekrar sağlanır.

Düğüm elemanlarına ulaşmak için liste üzerinde gezilir. Biz bu projede Tek Yönlü Bağlı Liste yapısını kullandık. Tek Yönlü Bağlı Liste yapısını açıklamak gerekirse, düğüm içerisinde iki adet değişken bulunur. Bunlardan biri veri için biri kelimenin sayısı için ayrılmıştır. Diğer değişken ise işaretçidir, **bir sonraki düğümün adresini tutar**. Bağlı liste dinamik bir veri yapısıdır ve programın çalışması sırasında büyüyüp küçülebilir. Dezavantajları ise, sahip olduğu işaretçiler sebebiyle dizilere göre daha çok hafıza gerektirir ve tekli bağlı listede geriye dönüş oldukça külfetlidir.

Biz bu projede bağlı liste yapısında tutulması için struct içinde bir str, adet ve pointer olarak sonraki değişkenlerini tanımladık. Böylece kelimeler sayıldıkça bağlı listeye eklenecek ve sonraki değişkeni ile diğer kelimeler ile bağlantıları olacaktır.

Bağlantılı Liste Yapısı

Kelime	Adet	
kasım	4	

 →

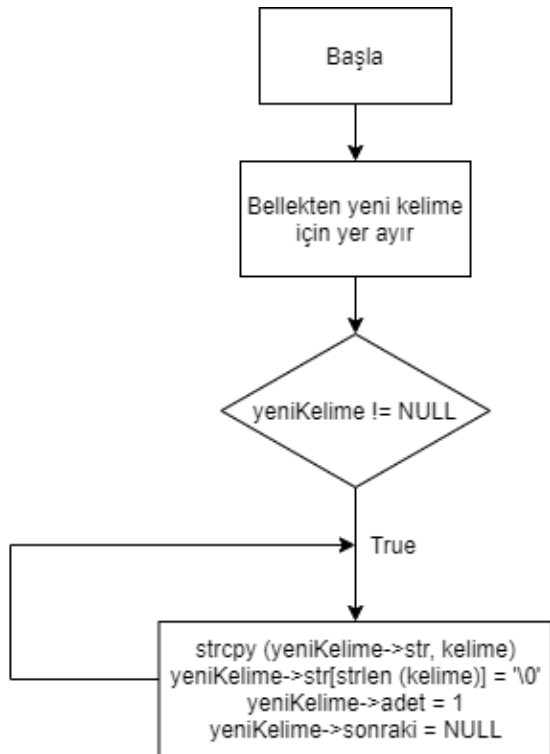
Kelime	Adet	
en	3	

Projeyi geliştirirken kelimeler tipinde kelime_olustur fonksiyonu tanımladık. Bu fonksiyon rekürsiftir ve yeni kelime oluşturmak için bellekten malloc fonksiyonu yardımıyla yer ayırır. Kelimenin str'sine fonksiyona gönderilen kelimeyi strcpy yardımı ile ekler, adetini 1 yapar ve sonraki kuyruğuna

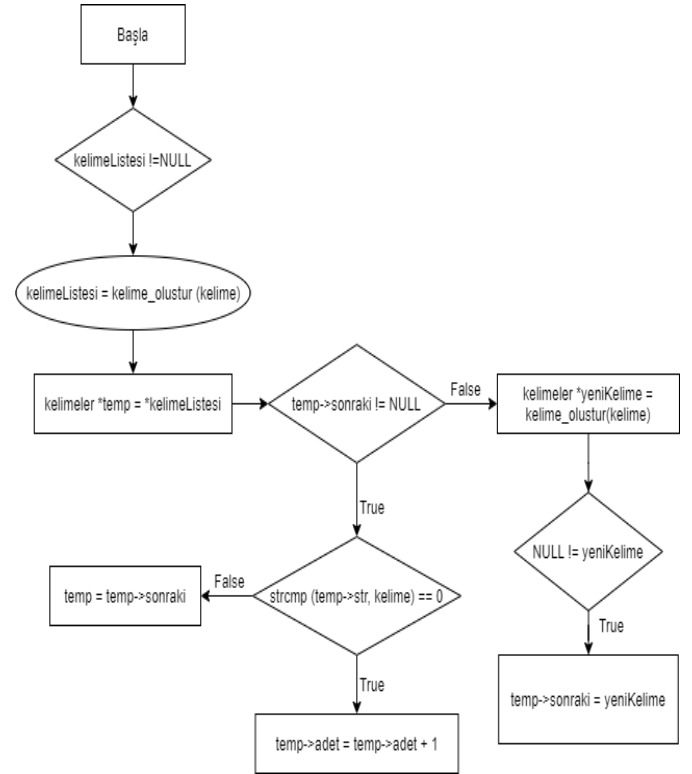
NULL değeri atar. Daha sonra kelimeyi listeye eklenmesi için kelimeyi döndürür. Kelime_ekle fonksiyonu yardımıyla da oluşturulan bu yeni kelime listeye eklenir. Eğer listede bu kelime varsa ekleme yapılmaz ve adeti bir arttırılır. Karşılaştığımız zorluklardan biri başa,sona ve araya ekleme fonksiyonlarıyla listeyi sıralayamamızdı. Bu fonksiyonlar yardımı ile listedeki kelimeleri adetlerine göre sıralayamadık.

Tasarım

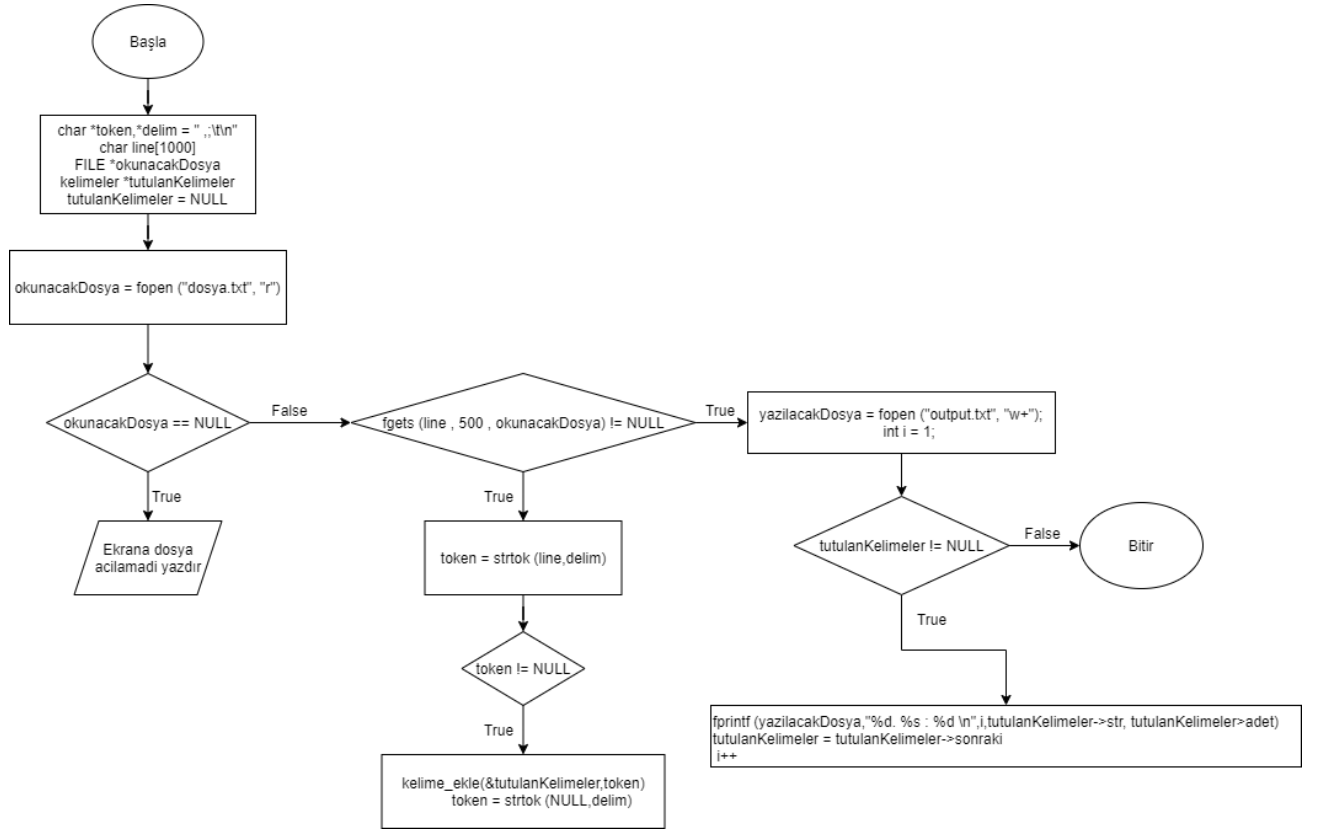
1.Akış şeması



kelimeler *kelime_olustur(char *kelime)



kelimeler *kelime_ekle (kelimeler
**kelimeListesi, char *kelime)



main()

2.Yazılım Mimarisi

```
1  fonksiyon kelime_olustur(kelime)
2  yeniKelime ← NULL
3
4  yeniKelime olustur
5
6  yeniKelime :  NULL değilse
7
8  yeniKelime->str ← bellekten yer al
9  kelimeyi kopyala
10 adetini 1 yap
11     sonraki kelime ← NULL
12 yeniKelime döndür
13 fonksiyon
kelime_ekle(kelimeListesi, kelime)
14  kelimeListesi NULL değilse
15  kelime olustur
16
17 kelimeListesi : NULL değilse
18 eğer kelime ← kelimeListesi->str
19 kelime adeti ← +1
20 değilse
21 sonraki kelimeye geç
22
23 dosya aç
24     eğer dosya ← NULL
25         Dosya açılmadı
26     değilse
27         dosya okunduğu
sürece
28 kelimeleri listeye ekle
29     yazılacak dosyayı aç
30     liste ← NULL değilse
31         kelime ve adeti
dosyaya yaz
32     sonraki düğüme geç
33     liste ← NULL
34     bitir.
```

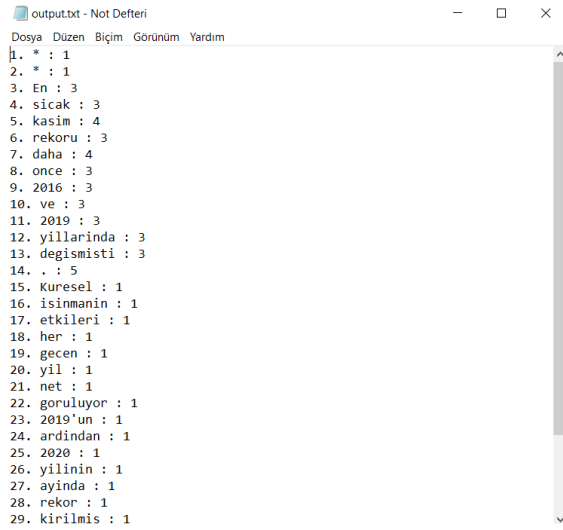
Genel Yapı

Proje tek yönlü bağlı liste yapısı kullanılarak yapılmıştır. Struct yapısında oluşturduğumuz kuyrukta char tipinde kelimenin kendisi, int tipinde adeti ve pointer olarak sonraki değişkeni vardır. Kelime_olustur fonksiyonu içinde her yeni kelime için malloc ile bellekten yer ayrılır. Bu kelime NULL olmadığı sürece fonksiyona gelen kelime, strcpy (yeniKelime->str, kelime) fonksiyonu ile bağlı listeye eklenir. Adeti 1 olarak belirlenir ve sonraki değeri NULL olarak atanır. Bu her kelime için böyledir. Daha sonra listeye eklenmesi için oluşturulan kelime return ile döndürülür.

Daha sonra kelime_ekle fonksiyonu sayesinde kelimeleri listeye ekleriz. Fonksiyona gönderilen liste boş olduğu sürece kelime_olustur fonksiyonu sayesinde listeye kelime eklenir. Daha sonra bu liste temp değişkenine eşitlenir. Temp->sonraki NULL olmadığı sürece döngüye girer ve eğer listedeki kelime yeni eklenen kelimeye eşitse o kelimenin adeti 1 artar ve listeye yeniden eklenmez.

Main içerisinde ise okunacak dosya açılır. Delim değişkeni belirlenir ve strtok fonksiyonu sayesinde kelimeler ayrıştırılır. Ayrıştırılan her kelime kelime_ekle fonksiyonu ile listeye eklenir. Eğer kelime mevcutsa bu fonksiyon

içinde adet arttırımı yapılır. Daha sonra sonuç için yazılacak dosya açılır. Kelime listesi NULL dönmedikçe fprintf fonksiyonu ile kelimeler adetleriyle birlikte dosyaya yazdırılır. Daha sonra bağlı liste yapısı sayesinde sonraki kelimeye geçilir ve sonraki kelime dosyaya yazdırılır. Bu döngü bittiğinde program sona erer.



```
1. * : 1
2. * : 1
3. En : 3
4. sıcak : 3
5. kasım : 4
6. rekoru : 3
7. daha : 4
8. önce : 3
9. 2016 : 3
10. ve : 3
11. 2019 : 3
12. yıllarında : 3
13. değişimisti : 3
14. . : 5
15. Kuresel : 1
16. isinmanın : 1
17. etkileri : 1
18. her : 1
19. geçen : 1
20. yıl : 1
21. net : 1
22. görülüyor : 1
23. 2019'un : 1
24. ardından : 1
25. 2020 : 1
26. yılının : 1
27. ayında : 1
28. rekor : 1
29. kırılmış : 1
```

Dosyaya yazdırılan kelimeler

Referanslar

Web Site

- https://www.bilgiunlugum.net/prog/cprog/c_dosya
- <https://medium.com/@tolgahan.cepel/do%C4%9Frusal-veri-yap%C4%B1lar%C4%B1-2-ba%C4%9Fl%C4%B1-liste-linked-list-8e5d3d84c41f>
- <http://ebubekirsezer.com/bagli-listelinked-list/>
- https://www.geeksforgeeks.org/strtok-strtok_r-functions-c-examples/