

Programlama Laboratuvarı I

PROJE#3 BAĞLI LİSTE İLE KELİME SAYMA

Proje Teslim Tarihi : 05.01.2021

Problemin Tanımı

Bu projede bir metin içerisinde her seferinde yeni bir kelimeye göre arama yapılacaktır. Sayma işlemi sonrasında adet miktarına göre büyükten küçüğe olacak şekilde bağlı listeye ekleme işlemi yapılacaktır. Bu bağlı listede kelime ve bu kelimeye karşılık gelen adet miktarı bulunmalıdır. Bu ekleme işlemi için başa, sona ve araya ekleme metotları kullanılmalıdır. Proje içerisinde diziler kullanılmayacaktır.

Yapılan Araştırmalar

Projede dizilerin kullanımı yasak olduğundan ilk karşılaşılan problem metin içerisinde okunan kelimenin nasıl tutulacağıydı. Bunun için bir adet char pointer yapısı oluşturuldu ve bu yapıya 100 karakterlik bir alan ayrıldı. Böylelikle bu değişken ile 100 karaktere kadar tüm kelimelerin tutulma işlemi gerçekleştirildi.

Bir diğer problem ise metinden okuma yaptığımız kelimenin adet miktarını bulmak amacıyla metin dosyanın sonuna kadar okuma yaptığımız için bu kelimenin konumunu kaybetmemiz ve kendisinden sonra gelecek

kelimelerin sayma işlemini yapamamamızdı. Bu problemin çözümü için sayma işlemi öncesinde metin içerisinde bulunduğumuz yeri **ftell** fonksiyonu ile kaldığımız yer bir değişken ile tutulmuş sayma işleminin akabinde **fseek** fonksiyonu ile kaldığımız yere geri dönlmüştür.

Genel Yapı ve Yöntem

Proje dosyasında metin isminde bir txt dosyası oluşturulur ve metin bu txt dosyasına yazılır. Dosya işlemleri yapılabilmek için **FILE** tipinde bir değişken oluşturulur ve **fopen** fonksiyonu okuma modundan oluşturularak dosyaya erişim sağlanır. Bağlı liste oluşturmak için **struct** tipinde "node" ismine sahip, kelimeleri tutmak için **char**, adet miktarlarını tutmak için **integer** tipinde iki adet değişken oluşturulur. Bağlı listeyi bir sonraki düğüme taşımak için **struct node** tipinde bir değişken daha oluşturulur. Dosya içerisinde okunacak kelimeleri tutmak amacıyla **char pointer** tipinde "kelime" isminde bir değişken oluşturulur, bu değişkene 100 karakterlik alan açılır. Dosyada okunacak kelime kalmadığında geriye sıfır değeri döndürecek bir değişken ile döngü oluşturulur.

Döngü içerisinde ilk olarak **kelimeBul** fonksiyonu ile dosya işaretçisinin bulunduğu konumdan kelime okuma işlemi yapılır ve okunan kelime, kelime ismindeki değişkene aktarılır. Eğer kelime bağlı liste içerisinde bulunuyorsa sonraki kelimenin okuma işlemi yine bu fonksiyon içerisinde gerçekleştirilir.

Bu işlemten sonra dosya işaretçinin kaldığı yer "**dosyaisaretcisi**" isminde bir değişkende tutulur.

Dosya okuma işleminin ardından eğer dosyanın sonuna gelinmişse **donguKontrol** metodu ile geriye 0 değeri döndürülür ve döngünün bir kez daha dönmemesinin önüne geçilir. Eğer son eleman bağlı liste içerisinde bulunuyorsa geriye -1

değerini döndürür ve döngünün içinden break komutu ile doğrudan çıkışı sağlar.

kelimeSay metodunda “tut” isminde char pointer tipinde 100 karakterlik alana sahip bir değişken ve adet miktarını bulmak için integer tipinde “say” isminde değişken oluşturulur. Dosyanın sonuna kadar tüm kelimeler sırasıyla tut değişkenine aktarılıp kelime değişkeni ile kıyaslanır. Denk gelen her şart için say değişkeni bir arttırılır. Dosya sonuna gelindiğinde say değişkeni geri döndürülür ve kelimenin metinde kaç defa tekrarlandığı bulunmuş olur.

Akabinde ekle metodu ile elimizde bulunan kelimenin adet miktarına göre ekleme işlemi yapılır. Eğer adet miktarı ilk elemandan büyükse basaEkle fonksiyonu, adet miktarı son elemandan küçükse sonaEkle fonksiyonu ve bu iki şartıda sağlamayan durumlar için arayaEkle fonksiyonu çalışmaktadır.

basaEkle fonksiyonunda ilk etapta eklenecek düğüm için yer açılır. Düğümdeki kelime değişkenine 100 karakterlik alan verilir. Düğümdeki değişkene strcpy fonksiyonu ile kelime değişkeni aktarılır. Düğümdeki adet değişkenine ise öncesinde hesaplanmış adet bilgisi aktarılır. Eklenecek eleman bir sonraki düğüm olarak ilk elemanı gösterir. ilkEleman’da eklecek elemanı gösterir böylelikle eklenecek eleman düğümün en başındaki eleman olmuş olur.

sonaEkle fonksiyonunda ilk etapta eklenecek düğüm için yer açılır. Düğümdeki kelime değişkenine 100 karakterlik alan verilir. Düğümdeki değişkene strcpy fonksiyonu ile kelime değişkeni aktarılır. Düğümdeki adet değişkenine ise öncesinde hesaplanmış adet bilgisi aktarılır. Eklenecek eleman bir sonraki düğümü NULL olarak ayarlar. İterator sonEleman’a eşitlenir ve iteratorın sonraki değerine ekle işlemi yapılır. sonEleman ise artık iteratorın sonraki değerine eşit olmuş olur.

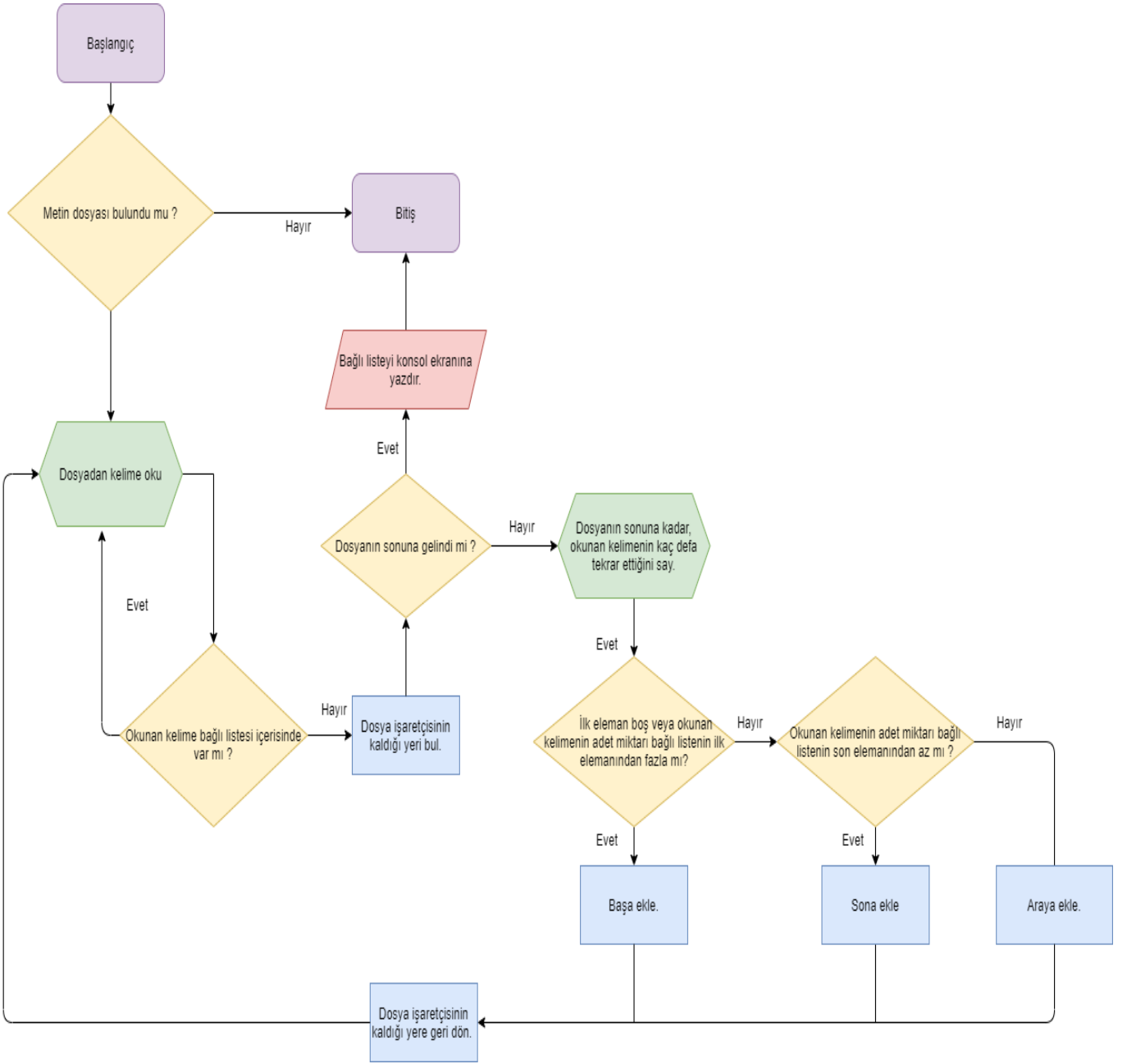
arayaEkle fonksiyonunda ilk etapta eklenecek düğüm için yer açılır. Düğümdeki kelime değişkenine 100 karakterlik alan verilir. Düğümdeki değişkene strcpy fonksiyonu ile kelime değişkeni aktarılır. Düğümdeki adet değişkenine ise öncesinde hesaplanmış adet bilgisi aktarılır. İterator ilk elemana eşitlenir ve iteratorın bir sonraki değerinin adet bilgisi elimizde bulunan kelimenin adet bilgisinden büyük olduğu sürece döndürülür ve bu süreçte iterator bir ileri noktaya taşınır. İteratorın sonraki değeri node tipinde bir değişken ile tutulur, iteratorın sonraki düğümüne eklenecek eleman koyulur ve bu eklenen elemanda en başta sakladığımız düğümü gösterir.

yazdir fonksiyonu ile düğümde bulunan tüm elemanlar iterator yardımıyla yazdırılır.

Yazılım Mimarisi

Tasarlanan sistem 2 çeşit düğüm (**node**) yapısından oluşmaktadır. Bu düğüm yapısında her bir kelime için kelime adı (**string**) ve adet miktarı (**int**) bilgileri bulunmaktadır. Tasarlanan sistemde tüm düğümler kelimelerin adet miktarına göre eklenmiştir.





Kaynakça

https://www.tutorialspoint.com/c_standard_library/c_function_fseek.htm

https://www.tutorialspoint.com/c_standard_library/c_function_ftell.htm

https://www.tutorialspoint.com/c_standard_library/string_h.htm

<http://bilgisayarkavramlari.com/2007/05/03/linked-list-linkli-liste-veya-bagli-liste/?highlight=ba%C4%9Fl%C4%B1>