

Bağlı Liste ile Kelime Sayma

Kaan Kalaycı, Alperen Ünlü

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

ahmet.alper96@gmail.com.tr, kaan.klyc1@hotmail.com

Problem Tanımı:

Bu projede bizden text içerisindeki kelimeleri sayarak bağlı listeye adet olarak büyükten küçüğe doğru eleman ekleyen bir program istenmiştir. Bu programı oluştururken sadece bağlı liste kullanmamız istenmiştir. Programın çalışırken istenen işlemlerin sırası; metin içerisinde kelimeyi bulup, adet sayısını hesaplayıp, adet sayısına göre kelimesiyle birlikte büyükten küçüğe doğru bağlı listeye eklenmesidir. Bağlı listeye eklemek için başa, sona ve araya ekleme metotlarının tümü kullanılmak zorundadır.

Yapılan Araştırmalar:

Text dosyasındaki metni kodumuzda kullanılabilir hale getirmekte sıkıntı yaşamıştık. Buna çözüm olarak fseek ve ftell fonksiyonlarını kullanarak bir char tipi pointera metni attık. Bazen farklı metinleri test ederken programın metni almadığını fark ettik. Bunun sebebi metinde bir boşluğun eksik olup strtok fonksiyonunun yanlış çalışmasıydı. Programı çalıştırdığımızda kelime saydığımız döngüde kelimenin sayısını belirten sayacın artmadığını fark ettik. Sorun, kelimeleri karşılaştırırken if koşulunda direkt olarak "==" yapısını kullanmamızdı. Bunu çözmek için kelimeleri karşılaştıran koşulumuzda strcmp fonksiyonunu kullandık. Son olarak, "Araya Ekle" metodunu uygun bir if koşulu bulamadığımız için kod boş çıktı veriyordu. Bunu çözmek için metoda koşul eklemek yerine else kullanarak diğer sağlanan koşullar dışındaki her durumda kelimenin ve adetin listenin arasına eklenmesini sağladık.

Genel Yapı:

Kodumuzda ilk olarak FILE türünde "text" adlı pointerı tanımladık ve fopen fonksiyonuyla kullanacağımız .txt dosyanın dosya yolunu aldık. Dosya adresinin boş dönmesi durumunda kullanıcıyı uyararak bir koşul yazdık. Sonra fseek ve ftell fonksiyonlarını kullanarak .txt dosyasındaki metnin boyutunu aldık ve bu boyutu kullanarak metni işleme sokabileceğimiz bir char türünde pointer tanımladık. fgets fonksiyonuyla pointerın içine metni attık ve dosyayı kapattık. Metindeki kelimeleri kullanabilmemiz için kelime adında char türü pointer tanımladık. Bağlı listede kullanacağımız düğüm olan "struct Node" yapısını kurduk. Bu yapı, char tipinde kelimeyi tutacak pointerı, sonraki düğümü tutacak pointerı ve kelimenin sayısını tutan int tipinde adet'e sahiptir. Öncelikle metindeki kelimeleri işleme sokmak için metnin tamamını kelime kelime strtok fonksiyonu ile ayırarak root1 isimli bağlı listeye attık. Bunun için void tipinde "ekle" fonksiyonu oluşturduk. Daha sonra root1'de gezinme işlemi yapabilmemiz için struct Node tipinde iter1 tanımladık ve bunu root1'e eşitledik. Daha sonra projede kelime ve adet bilgisini tutan asıl bağlı liste olan root2'yi tanımladık. Bir while döngüsü oluşturduk. Döngü, iter1 NULL olana kadar devam edecek. Döngü içinde kelimenin adetini saymak için bir sayaç tanımladık ve root1 listesinde tekrar dolaşabilmek için iter2 tanımladık ve bunu da root1'e eşitledik. Sonra root2'nin içinde iter1'de bulunan kelimenin olup olmadığını kontrol eden int

tipinde "is_in_list" fonksiyonunu içeren bir koşul tanımladık. Bu fonksiyon, iter1'deki kelime, root2'de bulunuyorsa koşulu sağlar ve iter1 kendini sonraki düğümüne eşitler ve döngü başa sarar. Sağlamıyorsa döngü devam eder. Bu sefer iter2 NULL olana kadar devam eden bir döngü açtık. Bu döngü, iter1de bulunan kelimenin metindeki sayısına bakarak sayacı arttırdı. Eğer root2 boşsa iter1deki kelimeyi ekleyen bir koşul oluşturduk. Ayrıca root2'nin içinde dolaşabilmemizi sağlayan iter3'ü tanımladık ve root2'ye eşitledik. root2 boş değilse iter3 NULL olana kadar devam eden bir döngü oluşturduk. Bu döngüde sayacın ve iter3'ün tuttuğu kelimenin adeti işleme sokularak başa ekle, araya ekle ve sona ekle metotlarından hangisini kullanacağını belirlemek için koşullar oluşturduk. Bu koşullardan herhangi biri sağlanırsa iter3'ü tekrar root2'ye, iter1'i sonraki düğümüne eşitleyip goto ile ana while döngüsüne dönülür. Sağlanmazsa iter3'ün sonraki düğümüne eşitlenir. Son olarak oluşturulan root2'nin kelimeler ve adetleri printList fonksiyonu ile ekrana yazdırılır.

Yazılım Mimarisi:

Bu programı C programlama dilini kullanarak Codeblocks geliştirme ortamında yazdık. Standart kütüphaneler dışında "string.h" kütüphanesini kullandık. Programımızı yazarken aşağıdaki fonksiyonları tasarlayıp kullandık.

-void printList(struct Node *node): Bu fonksiyon ile gönderilen bağlı listenin kelimelerini ve kelimelerin adetlerini ekrana yazdırdık.

- int is_in_list(struct Node **basla, char *kelime): Bu fonksiyon ile gönderilen bağlı listede gönderilen kelimenin olup olmadığını bakıp duruma göre bir değer döndürdük.

-void swap(struct Node *a, struct Node *b): Bu fonksiyon ile gönderilen iki düğümün kelime ve adetlerinin yerini değiştirdik. Bu fonksiyonu "bubbleSort" fonksiyonunda kullandık.

-void bubbleSort(struct Node *basla): Bu fonksiyon ile gönderilen düğümün içeriğine göre büyükten küçüğe sıralama yaptık. Bu fonksiyonu "araya_ekle" fonksiyonunda kullandık.

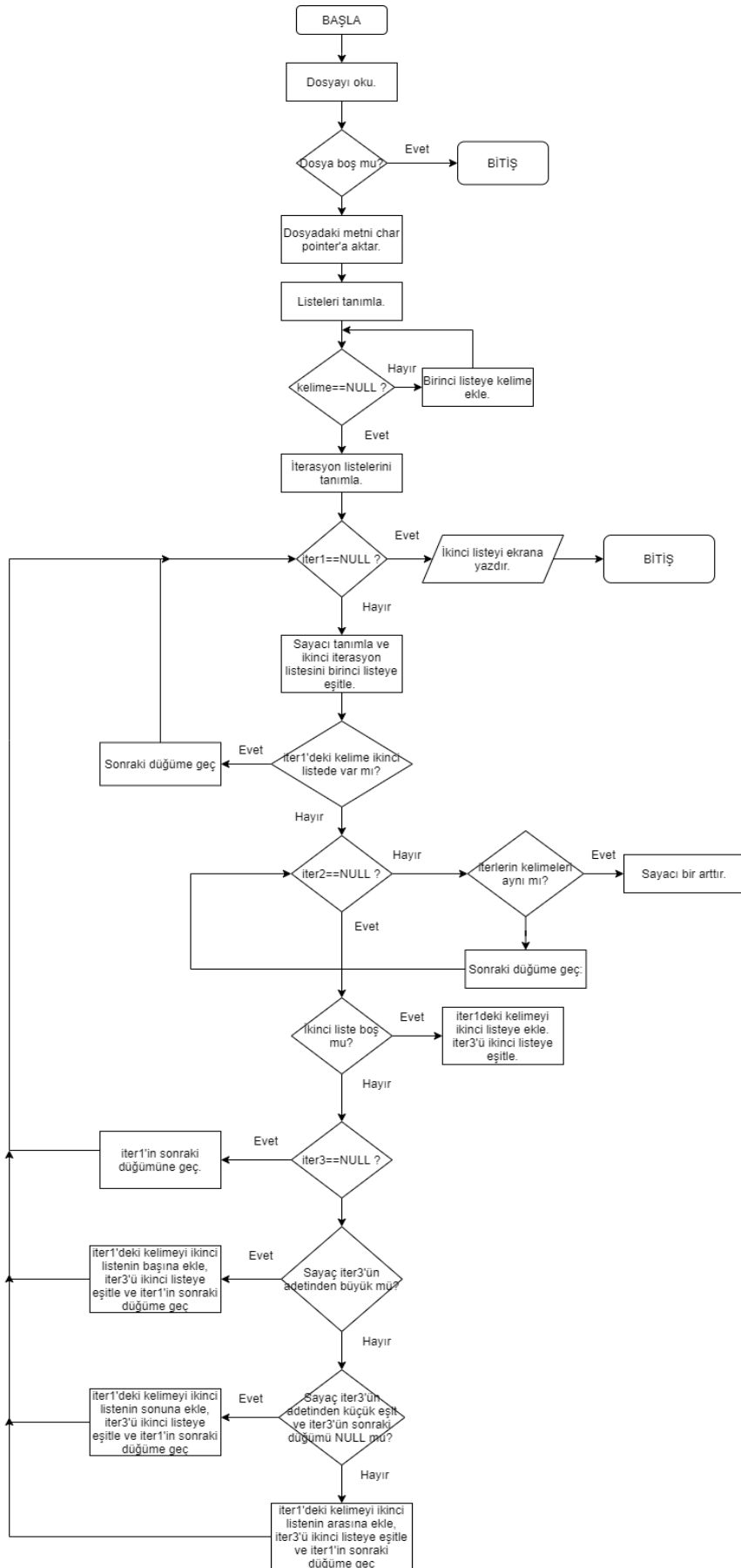
-void ekle(struct Node** basla, char *kelime): Bu fonksiyon ile gönderilen düğümüne gönderilen kelimeyi ekledik.

-void sona_ekle(struct Node** basla, char *kelime, int adet): Bu fonksiyon ile gönderilen düğümüne gönderilen kelimeyi ve kelimenin adetini düğümün sonuna ekledik.

-void basa_ekle(struct Node** basla, char *kelime, int adet): Bu fonksiyon ile gönderilen düğümüne gönderilen kelimeyi ve kelimenin adetini düğümün başına ekledik.

-void araya_ekle(struct Node** basla, char *kelime, int adet): Bu fonksiyon ile gönderilen düğümüne gönderilen kelimeyi ve kelimenin adetini düğümün arasına ekledik.

Akış Şeması:



Referanslar:

<http://bilgisayarkavramlari.com/2010/12/06/dosya-ve-bagli-liste-uygulamasi/>

<https://stackoverflow.com/questions/40592846/writing-data-from-a-linked-list-to-a-txt-file-in-c>

<https://stackoverflow.com/questions/41227292/c-linked-list-read-from-txt-file/41227579>

<https://www.geeksforgeeks.org/linked-list-set-1-introduction/>

<https://www.geeksforgeeks.org/c-program-bubble-sort-linked-list/>

http://embedded.kocaeli.edu.tr/?page_id=91

<https://www.geeksforgeeks.org/linked-list-set-2-inserting-a-node/?ref=lbp>