

Veri Yapıları

2.Ödev Raporu

Bu ödevde bizden, verilen icerik.txt dosyasının içeriğini sırayla okuyup bunları iki yönlü bağlı listede tutmamız istendi. Listeye eklenen kelimeler daha önceden listeye eklenmişse o aynı kelimenin konumunu bulup dugume bulduğumuz konumu yazarak veri tekrarından kurtulmamız bu şekilde hafızada kazanç elde etmemiz istendi.

Ödevde çift yönlü bağlı listede yapmamız istendiği için düğüm sınıfı hem ileri hem geri olmak üzere 2 tane işaretçi(pointer), girilen kelimeyi tutmak için 1 tane string alan ve konumu tutmak için 1 tane integer alana ve kurucu fonksiyona sahiptir. Konum ve kelime alanları dışarıya açık değildir. Sadece fonksiyonlar ile ulaşılabilir. Çift yönlü bağlı liste sınıfımız tasarlanırken genel liste yapısındaki gereksiz fonksiyonlardan arındırıldı. Liste sınıfımız baş düğümü tutmak için 1 tane düğüm işaretçi alana, hafızadaki kazancımızı tutmak için 1 tane integer alana, kurucu fonksiyonlara, konum farkını hesaplayabilmemiz için konumdanbul fonksiyonuna, listeye yeni kelime ekleyebilmemiz için sona ekle fonksiyonuna, liste silinirken çöp veri bırakmamak için yıkıcı fonksiyona ve ayrıca en son düğümleri ve kazancı ekrana yazdırabilmemiz için yazdır fonksiyonuna sahiptir. Bunlardan baş düğümü tutan alan, konumdanbul fonksiyonu ve toplam kazancın tutulduğu alan kullanıcıya kapalı sadece sınıfa açıktır. Diğer fonksiyonlara dışarıdan ulaşılmaktadır.

Yazdığımız bu programın genel çalışma yapısını özetleyelim. Text dosyamızdaki kelimeler okunur. Bu okunan kelimeler sırayla listeye eklemek için sonaEkle fonksiyonuna parametre olarak verilir. Bu fonksiyon baş düğüm boş ise aldığı kelimeyi direk yeni bir düğüme ekler ve o düğüm baş düğüm olur. Eğer baş düğüm varsa kelimenin daha önce listede eklenip eklenmediğini kontrol etmek için konumdanBul fonksiyonu çağrılır. KonumdanBul fonksiyonu tüm listeyi baştan aşağı gezerek listede kelimeyi arar ve kelimeyi bulabilirse konumu bulamazsa -1 döndürür. Eğer dönen sonuç -1 ise bu kelimenin daha önce listeye eklenmediği anlamına gelir ve ekleme fonksiyonu içerisinde heap bellek bölgesinde yeni bir düğüm yaratılır içine kelime yerleştirilir. Eğer dönen sonuç -1 den farklı ise bu konum demektir. Yeni düğüm yaratılır ve konum farkı hesaplanır, hesaplanan konum farkı düğüme yazılır kelime yazılmaz. Eğer sonuç -1 değilse hafıza da alan kazandık demektir bunun için kazanç hesaplanır. Bu kazançta toplam kazanca eklenir ve ekleme fonksiyonu burada biter. Tüm kelimeler bu şekilde listemize eklendikten sonra listeyi ve kazancı yazdır fonksiyonunu kullanarak ekrana yazdırabiliriz. Makefile dosyasını hazırlarken eski makefile örneklerinden yararlanıldı. Tüm sınıflarımızın object dosyalarını çıkartıp en son test dosyamızla birleştirip derleme aşamasını bitirdik. Derleme sırasında string alanıyla ilgili hatalar alsamda bu problemin isim uzayıyla alakalı olduğunu anladıktan sonra çözüme kavuştum. Ödevde beni zorlayan diğer bir kısım icerik.txt dosyasının konumundan kaynaklı derleme ve çalışma anında farklı sonuçlar almamdı. İlk olarak icerik.txt

dosyasını doc klasorunun içine koydum ve yolunu ona göre ayarladım. Derleme anında çalışsa da daha sonra exe üzerinden çalıştırmak istediğimizde içerik.txt dosyasını görmüyordu. Bu yüzden program için içerik yolunu içerik.txt olarak verip hem derleme hem çalışma anında doğru çalışması için hem genel dosyaya hemde bin klasorunun içine içerik.txt dosyasını yerleştirdim. Ödevde istenilen her şeyi yaptım ve eksik kısım bıraktığımı düşünmüyorum. Ödev farklı ortamlarda test edilmiştir ve sisteme yüklenmiştir.

Hazırlayan : Burak YAZAN