Sibel GENÇTÜRK

190201106

[Sblgnctrk19@gmail.com](mailto:Sblgnctrk19@gmail.com)

GİRİŞ

Bu projede karakter tabanlı LZ77, DEFLATE sıkıştırma algoritmaları kullanılarak dosya sıkıştırma uygulaması yapmanız istenmektedir. Proje için C / C++ programlama dili kullanılacaktır.

Projede istenenler sırasıyla aşağıda verilmektedir.

* Girdi olarak kullanılacak sıkıştırılmamış dosya, metin dosyası (metin.txt) şeklinde olacaktır.
* Sıkıştırılacak dosyadaki her bir karakter okunacaktır.
* En son her bir sıkıştırma algoritması için çıktı dosyası(metin dosyasının sıkıştırılmış hali) oluşturulacaktır.
* Kullandığınız sıkıştırma algoritmalarının performansları birbiri ile karşılaştırılacaktır.

LZ77 Sıkıştırma Algoritması

Genellikle hazırlanan metin belgelerinde, kendini tekrar eden kısımlar bulunmaktadır. Özellikle;

bağlaç ve zamirler,

diğer kelimelerin başına ve sonuna gelen ekler,

metinlerin içinde sık sık tekrar edilmektedirler.

LZ77 algoritmasının amacı, metin belgelerindeki tekrar eden bölümleri ortadan kaldırmak ve buna yönelik metin belgesini sıkıştırmaktır.

LZ77 yaklaşımında sözlük, daha önce kodlanmış serinin bir parçasıdır. Algoritmadaki arama tamponunun büyüklüğü, daha önce kodlanmış serinin ne büyüklükte bir parçasında arama yapılacağını belirler. Arama tamponu büyütüldükçe, sıkıştırma oranı artar, fakat aynı zamanda sıkıştırma zamanı da artar.

Örnek: “abracadabra” kelimesini LZ77 algoritması ile sıkıştırma.

4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | r | a  7 | c | a | d | a | b | r | a | a | a | a |

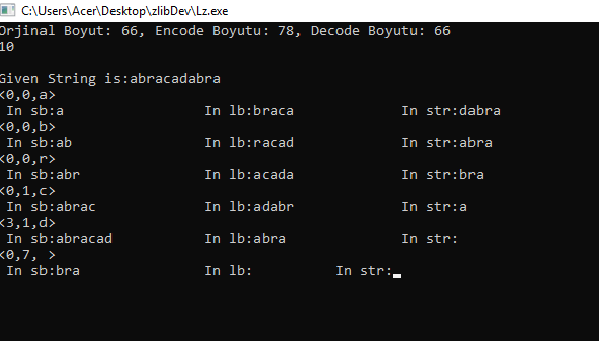
İleri tampon

Arama tamponu

İleri tamponun ilk karakteri olan a, arama tamponunda sondan başa doğru aranır.İkinci karşılaştırmada benzerlik bulunur, fakat bu karakterden sonra b karakteri değil ded karakteri yer aldığı için benzerlik uzunluğu sadece 1’dir. Arama devam ettirilir. İkikarakter sonra bir a daha bulunur, sonrasında c yer aldığı için bunun da benzerlikuzunluğu 1’dir. Aramaya devam edilir. Arama tamponunun başında, yani ileri tampondaaranan karakterden 7 uzaklıkta (offset=7) bir a daha bulunur. Bu defa benzerlikuzunluğu 4’tür (abra).

İleri tamponda“abra” serisinden sonra yer alan a karakteri ilebirlikte [7,4,C(a)] şeklinde üçlü olarak kodlanır. İleri tamponun en sonundaki akarakteri ise [0,0,C(a)] şeklinde kodlanır.

LZ77 ALGORİTMASI



**DEFLATE Sıkıştırma Algoritması**

Deflate Sıkıştırma Algoritması kayıpsız veri sıkıştırma formatıdır. Huffman Kodlaması ve LZSS algoritmasının bir bileşiminden oluşmaktadır.

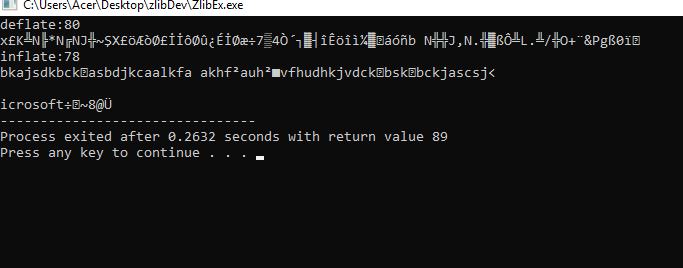
Sıkıştırılacak olan veri birbirini takip eden bloklar kümesi olarak düşünülmektedir ve her blok LZSS algoritması ve Huffman kodlamasının birlikte kullanılması ile sıkıştırılmaktadır. Her blok için oluşturulan Huffman ağacı bir önceki ve bir sonraki

bloktan bağımsızdır. Sıkıştırılabilen blokların büyüklüğü değişken olabilmektedir.

Deflate algoritmasında eğer Huffman ağacının kodlama yapamayacak kadar büyümesi durumu oluşursa, yeni bir Huffman ağacı oluşturmak için o bloğu sonlandırarak yeni bir blok başlatmaktadır.

Her blok, sıkıştırılmış verilerin bulunduğu kısım ve bu verileri temsil eden Huffman kod ağaçlarının saklandığı kısım olmak üzere iki kısımdan oluşur. KullanılanHuffman ağaçları her bloğun başında sıkıştırılmış veriden önce yer alır, ve bu ağaçlar da huffman algoritması kullanılarak sıkıştırılır.

DEFLATE ALGORİTMASI



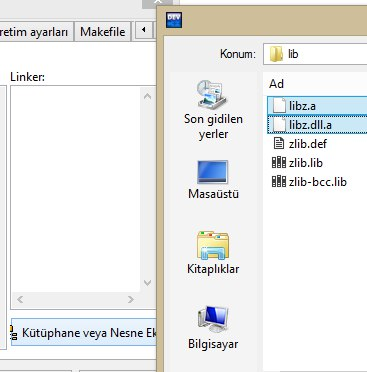
AMAÇ

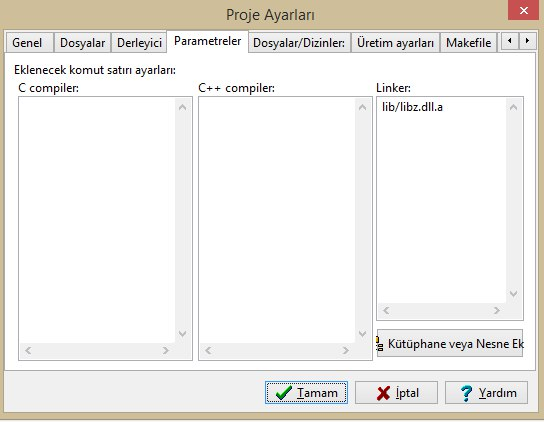
Bizden bu projede bu iki algoritmayı kullanarak algoritmaların sıkıştırdıkları boyutu süre açısını değerlendirip bu iki kayıpsız sıkıştırma algoritmalarılarını karşılaştırma yapmamız isteniyor.

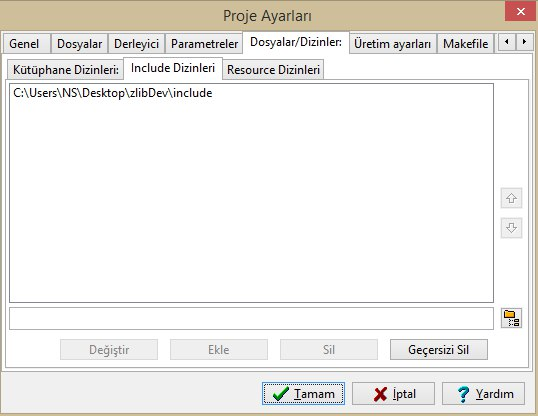
KULLANILAN DOSYALAR

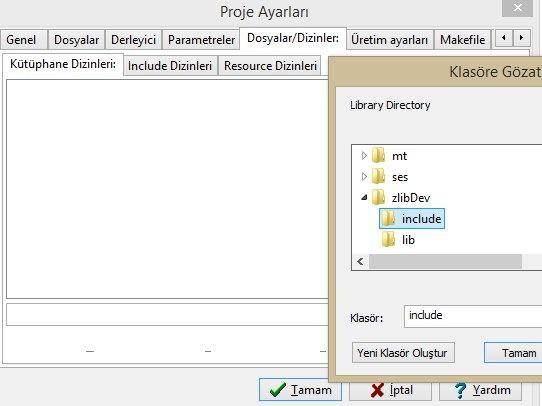
* Zlip kütüphanesi(bizim için gerekli olan zlip.h deflate.h dosyaları ana klasördeki include klasörüne eklenilerek kullanılmıştır.)
* Libz.dll dosyası(Ana klasördeki lib klasöründe bulunmaktadır.)
* Zlip kütüphanesi win32 içinde Makefile.gcc var.
* Makefile diye değiştirdim.
* win32 klasöründeki tüm dosyaları kopyaladım bir geri gelip yapıştırdım cmd ile girdim
* mingw32-make yazıp derledim
* zlip dll.a ve zlip1.dll dosyaları oluştu.
* Daha sonra bu dosyaları ana klasördeki lib klasörüne attım.

KAYNAK EKLEME AŞAMALARI









PROJEDE KARŞILAŞILAN SORUNLAR

* Projede deflate sıkıştırma algoritması kaynak bulma sorunu ve referans hatsı sorunu yaşanmış olup daha sonra gerekli kütüphanelerin proje dosyasının parametreler kısmına eklenmesiyle referans problemi çözülmüştür.

ÇALİSMA TAKVİMİ

* Kaynakların toplanması ve listelenmesi :4 gün
* Projenin teorik kısmının yazılması :2 hafta
* Projenin nihai olarak yazılması :4 gün

KULLANILAN KÜTÜPHANELER

* #include <stdlib.h>
* #include <stdio.h>
* #include <string.h>
* #include "zlib.h"
* #include <stdlib.h>
* #include <conio.h>

KULLANILAN YÖNTEMLER

* Proje c proglama dili ile yazılmıştır.
* Dll dosyaları kullanılmıştır.

KAYNAKÇA

* <http://www.emo.org.tr/ekler/dcd552693b5bec0_ek.pdf>
* <https://alwaysemmyhope.com/tr/algorithm/5468-identifying-deflate-algorithm-variant-being-used-in-proprietary-file-format-algorithm-compression-reverse-engineering-huffman-code-deflate.html>
* <http://www.durgut.com/gzip-ve-deflate/>
* <https://github.com/topics/deflate-algorithm>
* <https://github.com/nyucel/algoritma>
* <https://ysar.net/algoritma/lz77.html>
* <https://www.slideshare.net/veysiertekin/lz77-lempelziv-algorithm>
* <https://github.com/topics/lz77>
* <https://github.hillwoodhome.net/topics/lz77-compress>
* <https://www.tr.freelancer.com/projects/c-programming-algorithm/compression-algorithm-implementation/>
* <https://www.wikihow.com.tr/DLL-Dosyalar%C4%B1-Nas%C4%B1l-A%C3%A7%C4%B1l%C4%B1r>
* <https://sanalkurs.net/dll-dosyasi-olusturma-ve-kullanma-5780.html>