

# SİSTEM PROGRAMLAMA BAHAR 2021 PROJE ÖDEVİ

## Huffman Kodlama ve Çözümleme

**Ders yürütücüleri:** Prof.Dr. Ahmet ÖZMEN ve Doç.Dr. Ünal ÇAVUŞOĞLU

**Ders Asistanı:** Arş.Gör. Abdullah S. Bayraktar

**Veriliş tarihi:** 15. 4. 2021

**Grup bilgileri için Form Son veri giriş tarihi:** 21.04.2021, 23:59

**İlerleme raporu teslim tarihi:** 4.05.2021

**Teslim tarihi:** 16. 5. 2021 Pazar 23:59 Sabis üzerinden.

**İletişim:** Proje sürecinde sormak istediğiniz soruları ders asistanına mail ile iletebilirsiniz (sadece proje yöneticileri soru sorabilirler). Sualler ilgili duyuruda soru ve cevap olarak ve tüm öğrencilerin erişebileceği şekilde cevaplanacaktır ( direkt kod soruları veya bireysel yardım ve kontrol talepleri cevaplanmayacaktır). Duyuru linki: <https://asbayraktar.sakarya.edu.tr/tr/duyuru/goster/100900/sisprog-proje-soru-cevap>

### 1. GİRİŞ

Proje çalışmasında kriptoloji ile ilgili bir program geliştirilecektir. Program verilen anahtara göre giriş metnini kriptolayabilecek veya kriptolanmış bir metni eski haline döndürebilecektir. Program, kriptolama ve çözümleme için proje ile beraber verilen kriptolama kilitini kullanacaktır. Proje çalışması ders kapsamında öğretilen **libfdr** kütüphanesi kullanılarak C dilinde gerçekleştirilecektir, başka kütüphaneler kabul edilmeyecektir. Program Linux'te derlenip çalışacak şekilde gerçekleştirilecek, testler sadece Linux ortamında yapılacaktır.

Kriptolama için verilen json formatındaki kilit Huffman Code algoritması ile üretilmiştir.

**Huffman kodu** ([https://tr.wikipedia.org/wiki/Huffman\\_kodu](https://tr.wikipedia.org/wiki/Huffman_kodu)): Bilgisayar biliminde, veri sıkıştırması için kullanılan, bir entropi kodlama algoritmasıdır. David A. Huffman tarafından 1952 yılında geliştirilmiştir. Konu hakkında detaylı bilgi verilen web sayfası linkinden ve diğer kaynaklardan elde edilebilir.

### 2. GÖREV ve GİRİŞ ÇIKIŞ DOSYALARI

1) Kilit dosyasında json formatında bulunan her bir **kelime** - **Huffman kodu** ikilisi Red-Black Tree (**jrb**) üzerine yerleştirilecektir.

2) Girişe uygulanan dosyaların kodlanması veya çözümlenmesi için oluşturulan ağaç kullanılacaktır.

**Kullanım veya komut yapısı:** Olası işlemler için komut satırında kullanılacak anahtarlar ve açıklamaları aşağıdaki tabloda verilmiştir.

Komut	Açıklama
<code>\$&gt; krypto -d giriş_metin cikis_metin</code>	Kriptolanmış giriş metnini, aynı dizinde bulunan ".kilit" adlı dosyada bulunan kriptolama kilitine göre çözümler ve okunabilir çıkış metni üretir. Dizinde ".kilit" adlı dosya yoksa veya bozulmuşsa uygun şekilde hata mesajı verip çıkar.
<code>\$&gt; krypto -e giriş_metin cikis_metin</code>	Normal giriş metnini, aynı dizindeki ".kilit" adlı dosyada bulunan kriptolama kilitine göre kriptolar ve okunabilir kriptolama metni üretir. ".kilit" dosyası yoksa veya bozulmuşsa uygun şekilde hata mesajı verip çıkar.

**Not:** Adları "." ile başlayan dosyalar Linux'te (ve Unix'te) "\$> ls" komutu ile görülmezler. Görebilmek için "\$> ls -a" kullanmalısınız. Bu özellik nedeniyle Linux/Unix ortamında konfigürasyon dosyaları bu şekilde adlandırılır.

## GİRİŞ DOSYALARI:

**1) Normal giriş dosyası:** ASCII tabanlı metin dosyalarıdır (Türkçe karakter yok). İçinde rakamlar veya boşluk/tab ile ayrılmış kelimeler olabilir. Giriş dosyasının okunması ve çözülmesi işlemlerinde **fields** kullanılmalıdır. Giriş dosyası en fazla 500 kelime içerebilir.

**2) Kriptolu giriş dosyası:** ASCII tabanlı metin dosyalarıdır, içinde sadece birbirinden boşluk/tab ile ayrılmış {0, 1}'den oluşan diziler bulunabilir. Kriptolu giriş dosyasında ASCII kelimeler de olabilir. Bu kelimeler direk çıkışa aktarılacaktır. Kriptolu giriş dosyasının okunması ve çözülmesi işlemlerinde **fields** kullanılmalıdır, en fazla 500 kelime içerebilir.

**2) Kripto kilidi:** Kilit dosyası örnek bir metinden **json** biçiminde üretilecek ve sizlere verilecektir, içinde Huffman algoritmasına göre üretilmiş **{kelime, kod}** ikililerini barındıran ve kelime veya kodun en fazla 100 karakterden oluştuğu okunabilir bir dosyadır. Kilit dosyasının adı daima ".kilit" olacaktır ve proje dizininde bulunacaktır. Farklı kilitler için, sadece dosyanın içeriği değiştirilecektir. İsteyen örnek huffman algoritmasına uygun farklı kilitler oluşturabilir. Program çalıştığında -e veya -d opsiyonlarına göre **{kelime, kod}** ikilileri ağaca aşağıda verilen şekilde konacaktır:

-e opsiyonu (encode): Kodlama yapılacağı zaman kilit dosyasından okunan **{kelime, kod}** ikilisi JRB ağaçta → **{key, val}** şeklinde konulacaktır.

-d opsiyonu (decode): Çözümleme yapılacağı zaman kilit dosyasından okunan **{kelime, kod}** ikilisi JRB ağaçta → **{val, key}** şeklinde konulacaktır.

Her bir encode veya decode öncesi mutlaka ilk iş kilit dosyası okunacak ve ağaç uygun şekilde doldurulacak, daha sonra istenilen işlem gerçekleştirilecektir.

### Kilit dosyası örneği:

```
{
  "hi": "000",
  "hallo": "001",
  "merhaba": "01",
  "selamunaleykum": "1"
}
```

Kilit Dosyası örneklerine aşağıdaki linkten erişilebilir :

[https://asbayraktar.sakarya.edu.tr/sites/asbayraktar.sakarya.edu.tr/file/ornek\\_kilit\\_v1.0\\_.zip](https://asbayraktar.sakarya.edu.tr/sites/asbayraktar.sakarya.edu.tr/file/ornek_kilit_v1.0_.zip)

## ÇIKIŞ DOSYALARI:

**1) Kriptolanmış dosya:** Orijinal metindeki kelimeler yerine boşlukla ayrılmış Huffman kodları yazılmış metin tabanlı (ASCII) dosyadır. Kilit dosyasında olmayan bir kelime karşılaşırsa kelimenin orijinali çıkış dosyasına yazılacaktır. Bu bakımdan, eğer kurallara uygun boş bir kilit dosyası ile girişe bir dosya okutulursa (ağaç boş iken), giriş dosyası aynen çıkışa aktarılacaktır.

**2) Çözümlemiş dosya:** Bir kilit kullanılarak çözümlemiş okunabilir metin (ASCII) dosyadır. Kriptolanmış dosya içinde ağaçta olmayan bir kelimeye rastlanırsa, kriptolanmış kelime çıkışa yazılacaktır. Eğer kilit dosyası kurallara uygun ve boşsa, girişe uygulanan dosya aynen çıkışa aktarılacaktır.

## 3. ÇALIŞMA ŞEKLİ

**Grup çalışması:** Bu proje 4 kişiden oluşan grup çalışması ile yapılacaktır. Gruplarda 4'ten az öğrenci olabilir ancak tercih edilmez. Grup içi iletişim çok önemlidir, bu bakımdan iyi bir iş bölümü yaparak sonuca girmelisiniz.

**GitHub kullanımı:** Proje çalışması GitHub üzerinde yürütülecektir, bunun için her grup üyesi kendine ait bir [Github](#) hesabı açmalıdır (veya var olan hesabını kullanabilir). Grubun proje yöneticisi kendi hesabı ile bir **(kişisel-private)** proje oluşturup bu projeye sadece grup üyeleri ve ders yardımcısı Arş.Gör. Abdullah Salih Bayraktar'ı ( <https://github.com/sauseng> ) dahil edilecektir. Proje duyurulduktan sonra en geç 1 hafta içinde Git hesapları açılmış ve grup elemanları aşağıda linki verilen forma işlenmiş olmalıdır.

Form linki: <https://forms.gle/fvBYn5UugUTgcuEt6>

İş bölümü için aşağıda bir paylaşım önerilmiştir. Forma, grup elemanlarının hangi bölümde görev aldığı yazılmalıdır. Proje teslimi Git üzerinden olacak ve yapılan çalışma içindeki katkılar Git üzerinden "Commit" ler, yorumlar ve kodlara bakılarak yapılacaktır.

Örnek proje iş bölümü:

- 1) Proje sorumlusu (Github'da projenin oluşturulması, grup proje toplantıları ve bunların raporlanması, proje raporlarının takip edilip iletilmesi )
- 2) Deployment sorumlusu (Makefile ve argümanlar, programlama standartları), dokümantasyon ve düzen
- 3) Json formatında kilit dosyasının okunması
- 4) Encode ve decode işlemlerinin yapılması

Gruplar arası tartışma yapılabilir, ancak kod paylaşımı ASLA yapılmamalıdır. Tüm dosyalar benzerlik testine tabi tutulacaktır, kopya olduğu tespit edilen ödevler 0 (sıfır) ile notlandırılacaktır.

**Proje alanında bulunması istenen dosyalar:** Nihai proje: (Git bulunacağı gibi Sabis'e yönetici haricindeki diğer gupr üyelerinden herhangi biri tarafından yüklenecektir)

- 1) Readme: Grup elemanlarının adları ve kısa bir açıklama ile derleme sürecinin anlatıldığı metin tabanlı (ASCII) dosya. Varsa programın çalışmayan kısımları burada rapor edilmelidir.
- 2) Makefile: Otomatik derleme işlemleri için gereklidir. (make, clean komutları çalışmalıdır)
- 3) \*.c ve \*.h dosyalar: main.c ve projeniz için oluşturduğunuz diğer kaynak dosyalar.
- 4) Kilit dosyası.
- 5) Örnek giriş dosyası.

**İlerleme Raporu içeriği:** (Sabiste ödev kısmına yönetici tarafından yüklenecektir.)

- İlerleme raporu tek bir pdf dosyasından oluşmalıdır.
- Grubun üyelerinin isimlerinin altında projede sorumlu oldukları görevleri açıklanmalıdır.
- Projede gelinmiş olunan seviye ve karşılaşılan problemler aktarılmalıdır. (Ekran görüntüsü ile desteklenebilir.)
- Grubun yapmış olduğu en az bir çevrimiçi (Google meet vb,) toplantıya ait (ekran görüntüsü) iletilmelidir.
- Proje yöneticisinin proje süreciyle alakalı ilerleme özeti en alt kısımda "Yönetici ilerleme özeti" başlığı altında sunulmalıdır.

Git hesabınıza ASLA binary dosya yüklemeyiniz. Test sırasında kodlarınız indirilecek ve güncel Linux ortamında test edilecektir. Ayrıca, sadece kaynak kodların olduğu projenizin son hali zip ile sıkıştırılarak **MUTLAKA** SABİS'teki proje sayfasına (sadece proje yöneticisi tarafından ve ilerleme raporunun üstüne) yüklenecektir.

Her türlü yardım için ders asistanına ve yürütücülerine email atabilirsiniz.

**BAŞARILAR!**