

UTF 8 kodlama için C Dili String ve Dosya İşlemleri

Yrd.Doç.Dr. Fehim KÖYLÜ
Erciyes Üniversitesi Bilgisayar Mühendisliği

Evrensel Kodlama (Unicode)

- Dünyadaki tüm kodlama sistemleri ve karakterler için ortak bir standart temsil biçimidir.
- Unicode toplam 1.114.112 karakter kodlayabilir.
- Karakterler 0x000000 ile 0x10FFFF arasında bir kod alabilir.
- Unicode 9.0 (21 Haz 2016) toplam 128.172 karakteri kodlamıştır.
- UTF-8 8 bitlik bir evrensel kodlama (unicode) biçimidir.

UTF öncesi

- UTF-8, UTF-16 evrensel kodlamaya dayalı geliştirilmiş karakter kodlamalarıdır.
- UTF öncesi ASCII sistemi ile karakter kodlaması yapılmıştır.
- ASCII 7 bit kullanarak karakterleri kodlamıştır. 0 ile 127 arası kod ile toplam 128 karakter kodlanabilmiştir.
- ASCII tablosu İngilizcede yer almayan karakterleri kodlayabilmek için 8 bit uzunluğunda 1 Byte a çıkarılmıştır.
- İlk 8 bitlik ASCII tablosu ISO 8859-1 (Latin 1) latin alfabesi olmuştur.
- 256 karakter ile dile özgü karakterleri kodlamıştır. Ancak 256'dan fazla karakter olan diller için yetersiz kalmıştır.
- Bir başka dezavantajı farklı dillerin karakterleri aynı belgede yer alamamıştır.
- İnternet üzerinde Türkçe için geçerli Latin-5 kodlama belirtilmemişse/yanlışsa Türkçe karakterlerden ı, ş harfleri ý, þ olarak görünür. Varsayılan Latin-1 kodlamasında İzlanda alfabesindeki ý, þ karakterlerine karşılık geldiği içindir.

UTF 8

*Hangi şifreleme yöntemi ile şifrelendiğini bilmediğiniz bir string hiçbir anlam taşımaz. Daha ne kadar kafanızı kuma sokup ASCII plain-text'miş gibi davranacaksınız.

Karakterler

UN

Unicode

Code Charts

Home | Site Map | Search

Unicode 9.0 Character Code Charts

SCRIPTS | SYMBOLS | NOTES

Find chart by hex code: Related links: [Name index](#) [Help & links](#)

Scripts

European Scripts	African Scripts	South Asian Scripts	Indonesia & Oceania Scripts
Armenian	Adlam	Ahom	Ballinese
Armenian Ligatures	Bamum	Bengali and Assamese	Batak
Caucasian Albanian	Bamum Supplement	Bhaiksuki	Buginese
Cypriot Syllabary	Bassa Vah	Brahmi	Buhid
Cyrillic	Coptic	Chakma	Hanunoo
Cyrillic Supplement	Coptic in Greek block	Devanagari	Javanese
Cyrillic Extended-A	Coptic Epact Numbers	Devanagari Extended	Rejang
Cyrillic Extended-B	Egyptian Hieroglyphs (1MB)	Grantha	Sundanese
Cyrillic Extended-C	Ethiopic	Gujarati	Sundanese Supplement
Elbasan	Ethiopic Supplement	Gurmukhi	Tagalog
Georgian	Ethiopic Extended	Kaithi	Tagbanwa
Georgian Supplement	Ethiopic Extended-A	Kannada	East Asian Scripts
Glagolitic	Mende Kikakui	Kharoshthi	Bopomofo
Glagolitic Supplement	Meroitic	Khojki	Bopomofo Extended
Gothic	Meroitic Cursive	Khudawadi	CJK Unified Ideographs (Han) (35MB)
Greek	Meroitic Hieroglyphs	Lepcha	CJK Extension-A (6MB)
Greek Extended	N'Ko	Limbu	CJK Extension B (40MB)
Ancient Greek Numbers	Osmanya	Mahajani	CJK Extension C (3MB)
Latin	Tifinagh	Malayalam	CJK Extension D
Basic Latin (ASCII)	Vai	Meetei Mayek	CJK Extension E (3.5MB)
Latin-1 Supplement	Middle Eastern Scripts	Meetei Mayek Extensions	(see also UniHan Database)
Latin Extended-A	Anatolian Hieroglyphs	Modi	CJK Compatibility Ideographs
Latin Extended-B	Arabic	Mro	CJK Compatibility Ideographs Supplement
Latin Extended-C	Arabic Supplement	Multani	CJK Radicals / KangXi Radicals

UTF 8

- *”Unicode, karakterler için farklı bir düşünme tarzına sahip ve Unicode'un olayları nasıl düşündüğünü anlamak zorundasınız ve bunun dışındakiler mantıksız”.* <http://turkish.joelonsoftware.com/Articles/Unicode.html>

10C00

**<http://www.unicode.org/charts/PDF/U10C00.pdf>

Old Turkic

10C4F

	10C0	10C1	10C2	10C3	10C4
0	 10C00	 10C10	 10C20	 10C30	 10C40
1	 10C01	 10C11	 10C21	 10C31	 10C41
2	 10C02	 10C12	 10C22	 10C32	 10C42



UTF 8

*"Böylece UTF-8 adlı parlak fikir ortaya çıktı. UTF-8, hafızada 8bit kaplayan harika U+ numaraları olan Unicode kod noktalarınızı saklamak için kullanılan bir diğer sistemdir. UTF-8'de 0'dan 127'ye kadar olan kod noktaları tek bir byte'ta saklandı. Sadece 128 ve yukarısındaki kod noktaları 2,3 ve hatta 6 byte'ta saklandı."

Hex Min	Hex Max	Byte Sequence in Binary
00000000	0000007F	0vvvvvvv
00000080	000007FF	110vvvvv 10vvvvvv
00000800	0000FFFF	1110vvvv 10vvvvvv 10vvvvvv
00010000	001FFFFFFF	11110vvv 10vvvvvv 10vvvvvv 10vvvvvv
00200000	03FFFFFFF	111110vv 10vvvvvv 10vvvvvv 10vvvvvv 10vvvvvv
04000000	7FFFFFFF	1111110v 10vvvvvv 10vvvvvv 10vvvvvv 10vvvvvv 10vvvvvv

UTF 8

- Türkçe karakterler iki bayt yer kaplamaktadır.

```
000000: 22 65 76 72 61 6B 22 2C 22 4B 65 6C 69 6D 65 22 "evrak","Kelime"
000010: 2C 22 66 72 65 6B 61 6E 73 22 0A 31 2C 22 61 6C ,"frekans".1,"al
000020: 20 67 C3 BC 6C C3 BC 6D 20 76 65 72 20 67 C3 BC gÃ¼lÃ¼m ver gÃ¼
000030: 6C C3 BC 6D 22 2C 33 0A 31 2C 22 62 61 6C 67 C3 lÃ¼m",3.1,"balgÃ
000040: BC 6D 65 63 69 22 2C 35 0A 31 2C 22 62 65 73 6C ¼meci",5.1,"besl
```

Yerel kütüphaneler



Internationalization Libraries

- [Alis Batam](#) multilingual processing and display library.
- [AutoUniConv](#) (by Lingua-Systems)
A C/C++ library that automatically identifies an input's charset and converts it to Unicode. It supports a variety of input charsets and both detects and encodes UTF-8, UTF-16BE, UTF-16LE, UTF-32BE and UTF-32LE.

- <http://www.unicode.org/resources/libraries.html>

- [UTF-8 CPP, version 1.0](#)
A simple, portable and lightweight C++ library for handling UTF-8 encoded strings, with [documentation](#).
- [Unicode Controls & Classes for VB6](#) Version 2.6
A lightweight API driven ActiveX component that will turn your VB6 programs into fully Unicode (UTF16) applications.
- [xIUA](#) by X.Net, Inc. is an open source software package for Unicode enabling

UTF-8 cpp kütüphanesi

version 1.0

```
#ifndef UTF8_FOR_CPP_2675DCD0_9480_4c0c_B92A_CC14C027B731
#define UTF8_FOR_CPP_2675DCD0_9480_4c0c_B92A_CC14C027B731

#include "utf8/checked.h"
#include "utf8/unchecked.h"

#endif // header guard
```

utf-8 örnek kod (1/2)

```
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
#include "utf8.h"
using namespace std;
int main(int argc, char** argv)
{
    if (argc != 2) {
        cout << "\nUsage: docsample filename\n";
        return 0;
    }

    const char* test_file_path = argv[1];
    // Open the test file (contains UTF-8 encoded text)
    ifstream fs8(test_file_path);
    if (!fs8.is_open()) {
        cout << "Could not open " << test_file_path << endl;
        return 0;
    }

    unsigned line_count = 1;
    string line;
    // Play with all the lines in the file
    while (getline(fs8, line)) {
        // check for invalid utf-8 (for a simple yes/no check, there is also utf8::is_valid function)
        string::iterator end_it = utf8::find_invalid(line.begin(), line.end());
        if (end_it != line.end()) {
            cout << "Invalid UTF-8 encoding detected at line " << line_count << "\n";
            cout << "This part is fine: " << string(line.begin(), end_it) << "\n";
        }
    }
}
```

utf-8 örnek kod (2/2)

```
// Convert it to utf-16
vector<unsigned short> utf16line;
utf8::utf8to16(line.begin(), end_it, back_inserter(utf16line));

// And back to utf-8
string utf8line;
utf8::utf16to8(utf16line.begin(), utf16line.end(), back_inserter(utf8line));

// Confirm that the conversion went OK:
if (utf8line != string(line.begin(), end_it))
    cout << "Error in UTF-16 conversion at line: " << line_count << "\n";

    line_count++;
}
return 0;
}
```