

EECS 498 W26 System Design of a Search Engine

03 Utf8

Due Sunday, January 25, 2026, 11:59 pm

The purpose of this assignment is to learn how all the characters in all the languages in the entire world and all those countless emojis used on the web are encoded as Utf8 or Utf16. Later in the project, you'll use the Utf8 variable-length encoding scheme to encode deltas in your index.

Starter files

You are given these starter materials in `Utf8Starter.zip`.

Utf8.h	Defines some basic utility functions (that you will write) for reading and writing Utf8 and Utf16 characters, translating to/from Unicode.
Testcases	A directory containing a number of testcase text files in various formats.
Makefile	A makefile to build a flip utility (that you also write) that uses your Utf8 functions to translate Utf8 files to Utf16 and Utf16 files to Utf8.

The assignment

1. You are to implement the various functions for reading and writing Utf8 and Utf16 characters described in the supplied `Utf8.h` header as `Utf8.cpp`.
2. Using these functions, write a simple Linux utility, `flip.cpp`, that can convert a Utf8 text file to Utf16 or a Utf16 file to Utf8, writing the result to stdout.
 - a. You may use the usual C++ `iostream`, `fstream`, and other libraries or Linux memory-mapping, if you already know how. To avoid having to deal with buffer boundaries, I recommend reading the whole file into memory as binary data to start.
 - b. Look for a byte order mark (BOM) at the beginning of the input to indicate Utf8 or Utf16. If there's no BOM, assume it's Utf8.
 - c. Your Utf16 read and write routines only need to support little-endian format, the native encoding used on all PCs. If the BOM indicates you're reading big-endian Utf16, your flip utility should flip the byte sex of the data before decoding.
 - d. When decoding Utf8, overlong characters, Utf16 surrogates, noncharacters, and other security risk characters should be replaced by the replacement character.

- e. When writing Utf16, use little-endian format. The first 16-bit character should be the BOM. Values > 0x10ffff (21 bits) should be written as the replacement character.
 - f. When writing Utf8, do NOT write a BOM.
3. You are given the following files which may help with debug and test, in order of difficulty. The .txt version is the original Utf8. The .Utf16.txt version is a correct decoding of the Utf8 file, followed by a correct reencoding as Utf16 little-endian data. A .Utf16BE.txt version is Utf16 big-endian.

```
GettysburgAddress.txt
GettysburgAddress.Utf16.txt
GettysburgAddress.Utf16BE.txt
```

```
VariousLanguages.txt
VariousLanguages.Utf16.txt
VariousLanguages.Utf16BE.txt
```

```
PopularEmojis.txt
PopularEmojis.Utf16.txt
PopularEmojis.Utf16BE.txt
```

```
KuhnStressTest.txt
KuhnStressTest.Utf16.txt
KuhnStressTest.CleanUtf8.txt
KuhnStressTest.FlipTwice.txt
```

With the exception of the Kuhn stress test, flipping to Utf16 and then flipping back should result in the original text.

The stress test file tests the behavior of a Utf8 decoder on edge cases and on various overlong, malformed and other illegal characters, including characters that are legal in Utf8 but too big to be represented in Utf16. The .FlipTwice.txt file contains the result of flipping the stress test, then flipping back; it differs from original because so many characters get replaced either in Utf8 decoding or Utf16 encoding.

The .CleanUtf8.txt file contains the result of correctly decoding the stress test and correctly re-encoding it as clean Utf8. (Consider how you might create a version of your flip utility that does that for testing.)

4. Behavior should be as follows. The usage and error messages should be written to cerr. The return code is 1 for a usage message, zero otherwise. `cmp` compares files and generates no output if the files match. `od` is the octal dump; the `-x` option asks for hex output.

```
% g++ --std=c++17 flip.cpp utf8.cpp -o flip
% ./flip
Usage: flip <filename>
Recognize the input file as either Utf8 or Utf16, convert
it to the opposite encoding, and write the result to stdout.
% ./flip non-existent-file.txt
```

```

non-existent-file.txt not found.
% ./flip PopularEmojis.txt > PopularEmojis.Utf16testout.txt
% cmp PopularEmojis.Utf16.txt PopularEmojis.Utf16testout.txt
% ./flip PopularEmojis.Utf16BE.txt > PopularEmojis.Ut8testout.txt
% cmp PopularEmojis.Ut8testout.txt PopularEmojis.txt
% touch empty.txt
% ls -l empty.txt
-rw-r--r-- 1 nicole nicole 0 Dec 21 09:59 empty.txt
% ./flip empty.txt > empty.Utf16.txt
% od -x empty.Utf16.txt
00000000 feff
00000002
% ./flip empty.Utf16.txt > empty.Utf8.txt
% cmp empty.txt empty.Utf8.txt
% ls -l empty.Utf8.txt
-rw-r--r-- 1 nicole nicole 0 Dec 21 10:01 empty.Utf8.txt
%

```

5. Submit your Utf8.cpp and flip.cpp files to the autograder at <https://autograder.io/web/project/3485>.