Unicode

In this coding exercise, you will write a general-purpose Unicode encoding C program that will accept as input the names of an old file and its encoding, as well as the names of the new file and the encoding it should have. This might sound daunting, but you will be calling the powerful standard Linux program `iconv` as a basis for your program. (A version of the `iconv` program is also available for MS-Windows.)

First run of the encoder

```
$ unicode1
Enter name of file 1: sample-japanese-shiftjis.txt
Enter name of encoding 1: shift-jis
Enter name of file 2: j2.txt
Enter name of encoding 2: utf-16
iconv -c -f shift-jis -t utf-16 sample-japanese-shiftjis.txt > j2.txt
```

Notes

Please match the output of the re-encoding program you write as closely as possible to the excerpts given here, within reason. Of course, you should also check the re-encoded files themselves for errors.

References

- Unicode - Wikipedia
- The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!) - Joel on Software
- Ubuntu Manpage: iconv - Convert encoding of given files from one encoding to another
- ICONV for Windows (slightly outdated)

Invoking the encoder

```
$ unicode1
```

Notes

This line shows the user calling the program `unicode1` at the Linux command line prompt. This is the name for the program that the user has written. You can call your program almost anything.

Input file and encoding

```
Enter name of file 1: sample-japanese-shiftjis.txt
Enter name of encoding 1: shift-jis
```

Notes

Here the `unicode1` program is prompting the user for the name of the file to be re-encoded (`file 1`), as well as its current encoding (`encoding 1`). The user responds with `sample-japanese-shiftjis.txt` for the filename, and `shift-jis` for its current encoding. (Shift JIS is a common eight-bit Japanese language encoding.)

You should be able to obtain samples of Japanese that use Shift JIS encoding, Russian in KOI8-R encoding, and Hindi in UTF-8 encoding from the course materials. If they are not available, you can obtain UTF-8 samples of all three languages from Wikipedia and convert them as necessary.

References

- Shift JIS - Wikipedia
- KOI8-R - Wikipedia
- Hindi Wikipedia article on the Hindi language
- Japanese Wikipedia article on the Japanese language
- Russian Wikipedia article on the Russian language

Output file and encoding

```
Enter name of file 2: j2.txt
Enter name of encoding 2: utf-16
```

Notes

The user enters `j2.txt` for the name of the new file, and `utf-16` for the name of the new encoding, in this case, UTF-16, the default Microsoft Windows wide

character Unicode encoding. After you take this step, please also convert the Russian and Hindi samples to UTF-16.

References

- UTF-16 - Wikipedia
- Wide character - Wikipedia
- Please understand that Han unification is *the* problem

Assembling the command line

```
iconv -c -f shift-jis -t utf-16 sample-japanese-shiftjis.txt > j2.txt
```

Notes

Here, the `unicode1` program assembles and displays a Linux command line that calls the Linux encoding program `iconv`, and then `unicode1` sends the string to the Linux shell, where it runs.

The `-c` parameter configures `iconv` to silently discard characters it can't convert, rather than just stop.

The `-f` parameter specifies the "from" or input encoding, that is, the encoding of the old file. In this case, that's Shift JIS.

The `-t` parameter specifies the "to" or output encoding, that is, the encoding of the new file.

The file `sample-japanese-shiftjis.txt` is the old file, and the text `> j2.txt` specifies that the output should be redirected to a new file called `j2.txt`, rather than just displayed.

You should now be able to open the file `j2.txt` in a Windows text editor and view its contents.

References

- C library function - sprintf()
- How do I execute a Shell built-in command with a C function? - Stack Overflow
- EditPad Lite: Compact Unicode Text Editor—Edit Text Files in Any Language, Script or Code Page
- Notepad++

Second run of the encoder

```
$ unicode1
Enter name of file 1: j2.txt
Enter name of encoding 1: utf-16
Enter name of file 2: j3.txt
Enter name of encoding 2: utf-8
iconv -c -f utf-16 -t utf-8 j2.txt > j3.txt
```

Notes

This run of `unicode1` is similar to the first one, except that it takes the Japanese-language UTF-16 Windows file that was the result last time and re-encodes it as a UTF-8 file called `j3.txt` that's more suitable for viewing and checking on Linux.

You might want to try the same thing with the Russian and Hindi samples. Converting the Hindi sample from UTF-8 to UTF-16 and then back to UTF-8 is an example of a useful trick called "round-tripping". Round-tripping a file to an encoding you can read can help you check whether you have encoded it correctly.

References

- UTF-8 - Wikipedia
- UTF-8 and Unicode FAQ for Unix/Linux
- Emacs - Wikipedia
- Working with Coding Systems and Unicode in Emacs - Mastering Emacs