Representing text in binary

How is a text file represented?

Representing plain text in binary format

- Computers deal with numbers (represented in binary form)
- How can we then store characters?
- There’s a table for that!
- http://wiki.juneday.se/mediawiki/index.php/Term:ascii_table
- In the ASCII table, a set of characters are assigned numbers
- Note, the ASCII table is a standard and not a piece of software!
- It’s just a mapping between a selected set of characters and numbers

Example from the ASCII table

Let's look at an example of storing ABC using the ASCII table. Here's the relevant part of the ASCII table for the letters A, B and C:

- A 65
- B 66
- C 67
- In binary, A is coded as 65 in binary, which is 100 0001, B is 66 which is 100 0010, C is 67 which is 100 0011

So when the computer stores ABC in e.g. a file, using the ASCII table, the computer will store the following bits using 8 bit bytes:
0100 0001 0100 0010 0100 0011.

Storing ABC

So when the computer stores ABC in e.g. a file, using the ASCII table, the computer will store the following bits using 8 bit bytes:
0100 0001 0100 0010 0100 0011.

Note that we group the bits in chunks of four for increased readability here (the computer stores them consecutively).

This is so common, that one actually uses special notation for a chunk of four bits, using base 16 (a.k.a. hexadecimal).

Hexadecimal

Since in base 16, unsigned numbers go from 0 to 15, we need notation to express those values. In hexadecimal, values go from 0 to F:

0 1 2 3 4 5 6 7 8 9 A (10_{10}) B (11_{10}) C (12_{10}) D (13_{10}) E (14_{10}) F (15_{10})

Storing XYZ

Since we need four bits to represent decimal values 0 - 15, grouping bits in fours and translating to hexadecimal is very convenient (and common).

- X binary: 0101 1000 decimal: 88 hex: 58
- Y binary: 0101 1001 decimal: 89 hex: 59
- Z binary: 0101 1010 decimal: 90 hex: 5A

So the data stored will be: 0101 1000 0101 1001 0101 1010 which can be written for us humans as hex: (58 59 5a)$_{16}$

```
0101 1000 0101 1001 0101 1010
   5    8    5    9    5    A
```

Whitespace

- What about text with lines, tabs, spaces and sections?
- Those characters are also present in the ASCII table
- A line of text is simply text ending with a newline character
- The newline character has decimal value 10 in the ASCII table
- Two newline characters in a row will make the text look like it is divided in sections
- A tab is simply a number in the ASCII table, 9
- Space has number 32 (decimal) in the ASCII table.

Meaning of plain text

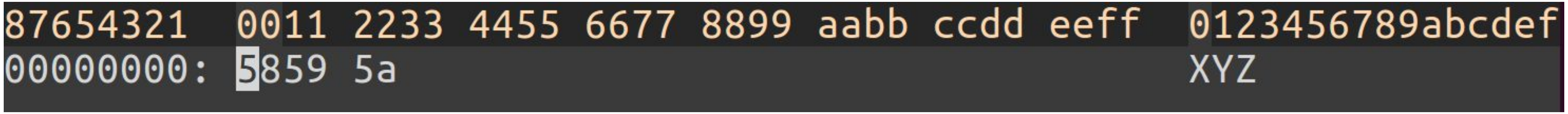
- Note that we can only represent structured text using the ASCII table
- There is no formatting such as fonts, font faces (bold, italics, underline etc)
- Data which contains only structured text from the ASCII table is called "plain text"
- In order to represent style and fonts, we need a different encoding (like a file format for Libre Office Write, or Microsoft Word)
- Styled documents are NOT called text documents or plain text
    - They are called something else, like word processor documents etc.

So, how is plain text stored, again?

- The ascii table is often used
- All text, including white space, is encoded as the binary representation of the corresponding ASCII numbers as a stream of e.g. bytes
- A text editor reads these bytes and shows them as text (and whitespace)

Hex view of a file with only XYZ

I saved the text XYZ in a file and opened it in emacs in hexl-mode:



```
87654321  0011 2233 4455 6677 8899 aabb ccdd eeff  0123456789abcdef
00000000: 5859 5a                                  XYZ
```

0x58 0x59 0x5A!

Looking at the file using the unix xxd command:

```
$ xxd xyz.txt
00000000: 5859 5a                                  XYZ
```

Plain text including whitespace

Text in file:

```
I am a plain text file.

Bye!
```

Hex codes:

```
$ xxd message.txt
00000000: 4920 616d 2061 2070 6c61 696e 2074 6578  I am a plain tex
00000010: 7420 6669 6c65 2e0a 0a42 7965 21         t file...Bye!
```

Can you spot the two linefeeds (which make it look like two sections)?

Plain text including whitespace

Can you spot the two linefeeds (which make it look like two sections)?

```
I am a plain text file.

Bye!
```

Hex codes:

```
$ xxd message.txt
00000000: 4920 616d 2061 2070 6c61 696e 2074 6578  I am a plain tex
00000010: 7420 6669 6c65 2e0a 0a42 7965 21         t file...Bye!
```

0A is the ASCII code for linefeed (in unix only LF is used for newline).

Plain text including whitespace

All hex codes on one line:

```
I am a plain text file.

Bye!
```

Hex codes:

```
$ xxd -p message.txt
4920616d206120706c61696e20746578742066696c652e0a0a42796521
```

Further reading

- https://en.wikipedia.org/wiki/ASCII