Character Encoding

Tom Ko

How to represent a character in computer?

- We need a constant way to represent each character.
- Consider the Chinese character 我, its code point is
  - \xA7DA in Big5
  - \xCED0 in GB18030
  - \x6211 in Unicode

Unicode

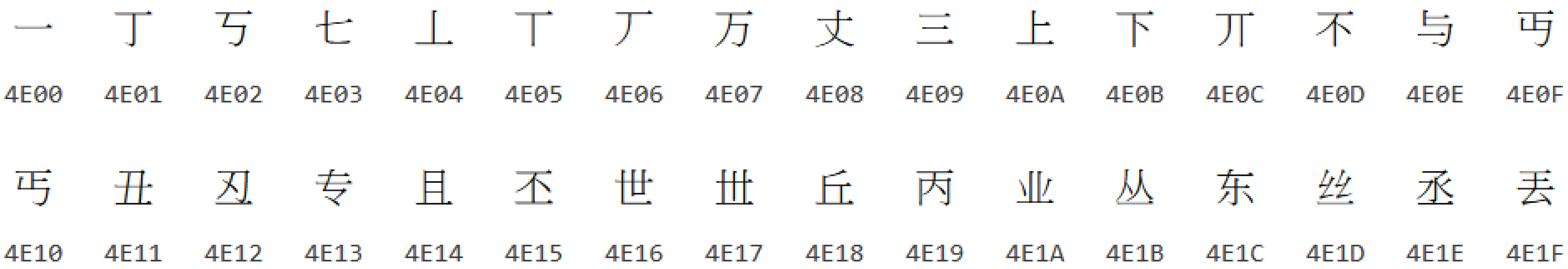
- **Unicode is a character set used to translate characters into numbers.**
- A character set is a list of characters with unique numbers (these numbers are sometimes referred to as "code points").
  - For example, in the Unicode character set, the number for A is 41.
- The first 128 Unicode code points represent the ASCII characters
- https://jrgraphix.net/r/Unicode/

Unicode Character Ranges

0020 – 007F	Basic Latin
00A0 – 00FF	Latin-1 Supplement
0100 – 017F	Latin Extended-A
0180 – 024F	Latin Extended-B
0250 – 02AF	IPA Extensions
02B0 – 02FF	Spacing Modifier Letters
0300 – 036F	Combining Diacritical Marks
0370 – 03FF	Greek and Coptic
0400 – 04FF	Cyrillic
0500 – 052F	Cyrillic Supplementary
0530 – 058F	Armenian
0590 – 05FF	Hebrew
0600 – 06FF	Arabic
0700 – 074F	Syriac
0780 – 07BF	Thaana
2580 – 259F	Block Elements
25A0 – 25FF	Geometric Shapes
2600 – 26FF	Miscellaneous Symbols
2700 – 27BF	Dingbats
27C0 – 27EF	Miscellaneous Mathematical Symbols-A
27F0 – 27FF	Supplemental Arrows-A
2800 – 28FF	Braille Patterns
2900 – 297F	Supplemental Arrows-B
2980 – 29FF	Miscellaneous Mathematical Symbols-B
2A00 – 2AFF	Supplemental Mathematical Operators
2B00 – 2BFF	Miscellaneous Symbols and Arrows
2E80 – 2EFF	CJK Radicals Supplement
2F00 – 2FDF	Kangxi Radicals
2FF0 – 2FFF	Ideographic Description Characters
3000 – 303F	CJK Symbols and Punctuation

CJK character range

- CJK Unified Ideographs (4E00–9FFF)



一	丁	丂	七	丄	丅	丆	万	丈	三	上	下	丌	不	与	丏
4E00	4E01	4E02	4E03	4E04	4E05	4E06	4E07	4E08	4E09	4E0A	4E0B	4E0C	4E0D	4E0E	4E0F
丐	丑	丒	专	且	丕	世	丗	丘	丙	业	丛	东	丝	丞	丟
4E10	4E11	4E12	4E13	4E14	4E15	4E16	4E17	4E18	4E19	4E1A	4E1B	4E1C	4E1D	4E1E	4E1F

Unicode Transformation Format (UTF)

- UTF is an encoding used to translate numbers into binary data.
- UTF-8, UTF-16, UTF-32
- An encoding is an algorithm that translates a list of numbers to binary so it can be stored on disk. For example UTF-8 would translate the number sequence 1, 2, 3, 4 like this:

- 00000001 00000010 00000011 00000100

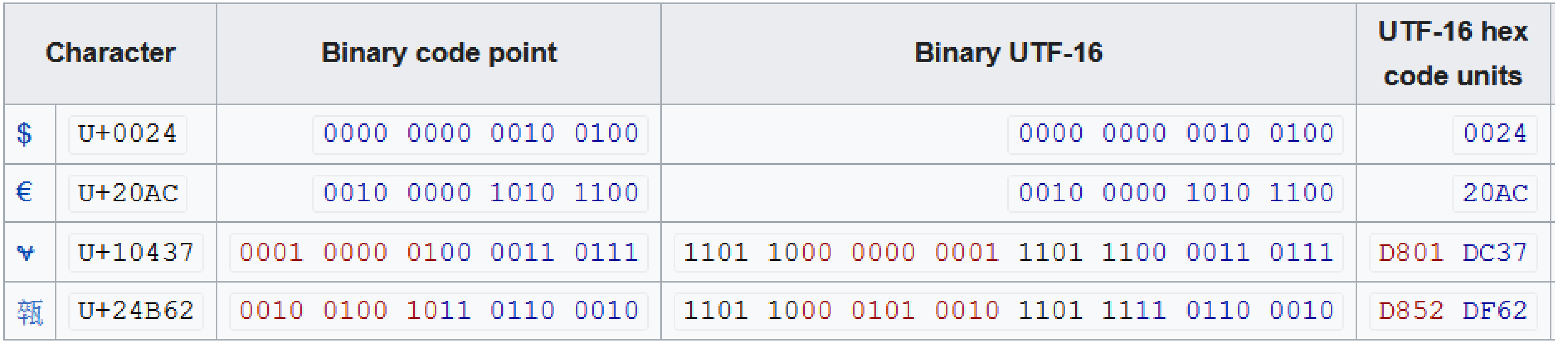
UTF-8

- It uses one to four one-byte (8-bit) code units (variable-length encoding).
- Code points with lower numerical values, which tend to occur more frequently, are encoded using fewer bytes.

Number of bytes	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	U+0000	U+007F	0xxxxxxx			
2	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-16

- It uses one or two 16-bit code units (variable-length encoding), depending on the character range.
  - U+0000 to U+D7FF and U+E000 to U+FFFF
  - U+D800 to U+DFFF
  - U+010000 to U+10FFFF



Character		Binary code point	Binary UTF-16	UTF-16 hex code units
$	U+0024	0000 0000 0010 0100	0000 0000 0010 0100	0024
€	U+20AC	0010 0000 1010 1100	0010 0000 1010 1100	20AC
𐐷	U+10437	0001 0000 0100 0011 0111	1101 1000 0000 0001 1101 1100 0011 0111	D801 DC37
𤭢	U+24B62	0010 0100 1011 0110 0010	1101 1000 0101 0010 1101 1111 0110 0010	D852 DF62

UTF-32

- It is a fixed-length encoding used to encode Unicode code points that uses exactly 32 bits (four bytes) per code point.
- A number of leading bits must be zero as there are far fewer than 2^{32} Unicode code points

Chinese character encoding

- Big5
  - Two-byte encoding
  - A440 一 乙 丁 七 乃 九 了 二 人 儿 入 八 几 刀 刁 力
  - A450 匕 十 卜 又 三 下 丈 上 丫 丸 凡 久 么 也 乞 于

- GB18030
  - variable-width encoding
  - CED0 涡 窝 我 斡 卧 握 沃 巫 呜 钨 乌 污 诬 屋 无 芜
  - CEE0 梧 吾 吴 毋 武 五 捂 午 舞 伍 侮 坞 戊 雾 晤 物

File viewing

- How does editors know the encoding of the file?
  - They don’t know, but they make a guess
- Files, e.g. plain text files, don't often explicitly state their encodings
- Different editors use different heuristics to detect the encodings
- It may fail to open files with a particular encoding
  - E.g. when it read a leading byte start with bits 10
- The LANG environment variable may affect the editor

Vim

- From :help fileencodings:
  - This is a list of character encodings considered when starting to edit an existing file. When a file is read, Vim tries to use the first mentioned character encoding. If an error is detected, the next one in the list is tried. When an encoding is found that works, fileencoding is set to it.
- You should put these settings in your .vimrc file so that you don't have to input them manually each time you start vim.

Helpful Linux commands

- file
  - For checking file encoding
- iconv
  - For converting the encoding of a file

Duplicate characters in Unicode

- A problem for text processing
- You can check the code points at
  - https://www.branah.com/unicode-converter
- 告 告
- 群 羣
- 券 券
- 奬 奬
- 麪 麪 麪 麵 麵

Duplicate characters in Unicode

- 㓟　㓟
- 冚　冚
- 餸　餸
- 瞓　瞓
- 軚　軚