SESSION 6 – CHARACTER DATA REPRESENTATION

Reading: Section 2.6

Reading

• Wikipedia – Unicode
  en.wikipedia.org/wiki/Unicode
Objectives

• Understand how text (sequence of characters) is represented in a computer
• Understand the difference between character code points expressed in Unicode and in character encodings
• Gain familiarity with the most popular character codes

You will frequently need to use hex encodings of characters (e.g., HTML special characters)

Characters

• Languages consist of a set of characters, usually defined as the smallest unit of information in the written form of a natural language
• Examples
  • English includes 26 letters (a-z), along with their capital equivalents, digits (0-9), and special symbols (e.g., “,”)
  • Chinese has 4,000 characters for general language coverage and 40,000 characters for more complete coverage
  • Japanese has 2,000 characters for general language coverage
  • There are approximately 6,800 living languages in the world today
Character Code Issues

- Character codes
  - Mapping of characters to strings of binary digits
  - E.g., “S” usually is usually mapped to $01000011_2$
  - Mapping to an 8-bit code restricts the language to 256 characters
  - Mapping to longer character codes can result in longer strings
    - Length of text strings sometimes a concern, (much less with inexpensive memory and disk)
    - Text is occasionally transmitted over low bandwidth communications links

Character Codes

- Character codes have evolved as trade becomes more global
- There is no universally agreed upon character set
Early Character Codes

• The earliest computer coding systems used six bits (BCD), allowing 64 characters
• In 1963
  • 8-bit EBCDIC was introduced by IBM
  • The 7-bit ASCII code was introduced and used by other computer HW manufacturers
• The codes are
  • Clearly inadequate for global commerce
  • Important to understand implementation of current codes (backwards compatibility)

ASCII Reference Table

<table>
<thead>
<tr>
<th>MSD</th>
<th>LSD</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>NUL</td>
<td>DLE</td>
<td>SP</td>
<td>0</td>
<td>@</td>
<td>P</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>SOH</td>
<td>DC1</td>
<td>!</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>STX</td>
<td>DC2</td>
<td>&quot;</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>ETX</td>
<td>DC3</td>
<td>#</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>s</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>EOT</td>
<td>DC4</td>
<td>$</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>ENQ</td>
<td>NAK</td>
<td>%</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>AC1</td>
<td>SYM</td>
<td>&amp;</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>BEL</td>
<td>ETB</td>
<td>'</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>BS</td>
<td>CAN</td>
<td>(</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>HT</td>
<td>EM</td>
<td>)</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>LF</td>
<td>SUB</td>
<td>*</td>
<td>:</td>
<td>J</td>
<td>Z</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>VT</td>
<td>ESC</td>
<td>+</td>
<td>:</td>
<td>K</td>
<td>[</td>
<td>k</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>FF</td>
<td>FS</td>
<td>.</td>
<td>&lt;</td>
<td>L</td>
<td>\</td>
<td>l</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>CR</td>
<td>GS</td>
<td>=</td>
<td>=</td>
<td>M</td>
<td></td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>SO</td>
<td>RS</td>
<td>.</td>
<td>&gt;</td>
<td>N</td>
<td>^</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>SI</td>
<td>US</td>
<td>/</td>
<td>?</td>
<td>O</td>
<td>_</td>
<td>o</td>
<td></td>
</tr>
</tbody>
</table>

Note the ordering of characters

© Robert F. Kelly, 2014-2016
Modern Approach to Encoding

- Establish
  - Universal set of characters that can be encoded in a variety of ways
  - Ordering of the characters
- Character repertoire - the full set of abstract characters that a system supports, and might allow
  - No additions – e.g., ASCII
  - Additions
- Examples
  - Unicode
  - ISO/IEC10646

Unicode

- Can encode the characters of every language in the world
- Contains
  - more than 120,000 characters (Universal Character Set)
  - 129 scripts (e.g., Latin, Arabic)
  - Codepoint for every character
  - A 6-part codespace (e.g., Western alphabet codes)
- Equivalent (almost) to ISO 10646
- Implemented by various encodings
  - UTF-8 – one byte for ASCII characters and up to 4 bytes for other characters
  - UTF-16 – 2-4 bytes for each character

Java uses Unicode as its default character set
Unicode Codespace Allocation

- The lowest-numbered Unicode characters comprise the ASCII code – preserves backwards compatibility

<table>
<thead>
<tr>
<th>Character Types</th>
<th>Language</th>
<th>Number of Characters</th>
<th>Hexadecimal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabets</td>
<td>Latin, Greek, Cyrillic, etc.</td>
<td>8192</td>
<td>0000 to 1FFF</td>
</tr>
<tr>
<td>Symbols</td>
<td>Dingbats, Mathematical, etc.</td>
<td>4096</td>
<td>2000 to 2FFF</td>
</tr>
<tr>
<td>CJK</td>
<td>Chinese, Japanese, and Korean, phonetic symbols and punctuation</td>
<td>4096</td>
<td>3000 to 3FFF</td>
</tr>
<tr>
<td>Han</td>
<td>Unified Chinese, Japanese, and Korean</td>
<td>40,960</td>
<td>4000 to DFFF</td>
</tr>
<tr>
<td>Han Expansion</td>
<td>Han Expansion</td>
<td>4096</td>
<td>E000 to FFFF</td>
</tr>
<tr>
<td>User Defined</td>
<td>User Defined</td>
<td>4095</td>
<td>F000 to FFFE</td>
</tr>
</tbody>
</table>

Example - HTML

- An HTML document consists of Unicode characters \&#38;.
- When transmitted, the document is encoded according to document / server instructions, as in
  `<meta http-equiv="content-type" content="text/html; charset=UTF-8" />`
  or in HTML5  `<meta charset="UTF-8">`

- Examples

<table>
<thead>
<tr>
<th>Entity Reference</th>
<th>Category</th>
<th>Displays As</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;#x5E7;</td>
<td>Hebrew</td>
<td>ג</td>
</tr>
<tr>
<td>&amp;#x645;</td>
<td>Arabic</td>
<td>م</td>
</tr>
<tr>
<td>&amp;#x8449;</td>
<td>Chinese</td>
<td>葉</td>
</tr>
<tr>
<td>&amp;#xB5AB;</td>
<td>Korean</td>
<td>뷁</td>
</tr>
</tbody>
</table>
**HxD**

- Freeware text editor that
  - Displays the hex representation of text in a file
  - Allows you to manipulate the binary data in a file

---

**How is “ISE218” Stored?**

- ise218.txt contains “ISE218”

---

Character representation (1 character per byte)
How is “ISE218” Stored in UTF-8?

- ISE218-UTF.txt contains “ISE218” stored in UTF-8

Note that the UTF-8 encoding corresponds to ASCII encoding for standard English characters.

What are the first 3 bytes?

Editing and Viewing Hex Codes

- HxD – Editing hex codes of a txt file
  - You can download from http://download.cnet.com/HxD-Hex-Editor/3000-2352_4-10891068.html
  - Use the Convert menu to display in a given character set or convert to a given character set
- EditPad Lite – viewing characters in various code representations
  http://www.editpadlite.com/
UTF-8 Example

- Let's look at the hex codes used to represent some special characters found in the UTF-8 Wiki page

Codes:
24 C2 A2 E2 82 AC

Did You Satisfy the Objectives?

- Understand how text (sequence of characters) is represented in a computer
- Understand the difference between character codepoints (e.g., Unicode) and character encodings
- Gain familiarity with the most popular character codes