Introduction to Informatics

Lecture 10: Encoding Numbers (Part II)
Readings until now

- **Lecture notes**
  - Posted online @ [http://informatics.indiana.edu/rocha/i101](http://informatics.indiana.edu/rocha/i101)
  - *The Nature of Information*
  - *Technology*
  - *Modeling the World*
  - @ *infoport*
  - From course package
      - Chapters 1, 4 (pages 1-12)
    - From Andy Clark’s book “*Natural-Born Cyborgs*”
      - Chapters 2 and 6 (pages 19 - 67)
    - From Irv Englander’s book “*The Architecture of Computer Hardware and Systems Software*”
      - Chapter 3: Data Formats (pp. 70-86)
Assignment Situation

- **Labs**
  - **Past**
    - Lab 1: Blogs
      - Closed (Friday, January 19): Grades Posted
    - Lab 2: Basic HTML
      - Closed (Wednesday, January 31): Grades Posted
    - Lab 3: Advanced HTML: Cascading Style Sheets
      - Closed (Friday, February 2): Grades Posted
    - Lab 4: More HTML and CSS
      - Closed (Friday, February 9): being graded
    - Lab 5: Introduction to Operating Systems: Unix
      - Due Friday, February 16
  - **Next: Lab 6**
    - More Unix and FTP
      - Due Friday, February 23

- **Assignments**
  - **Individual**
    - First installment
      - Closed: February 9: Being Graded
  - **Group Project**
    - First installment
      - Presented: February 20, Due: March 9th

- **Midterm Exam**
  - March 1st (Thursday)
Individual assignment

- Individual Project
  - 1st installment
    - Presented: February 1st
    - Due: February 9th
  - 2nd Installment
    - Presented: February 15th
    - Due: March 2nd
  - 3rd Installment
    - Presented: March 8th
    - Due: March 30th
  - 4th Installment
    - Presented: April 5th
    - Due: April 20th

The Black Box

What is it!!??
The Modeling Relation

Symbols (Images) \rightarrow Initial Conditions \rightarrow Model (syntax) \rightarrow Logical Consequence of Model \rightarrow Predicted Result & Observed Result

World$_1$ \rightarrow Physical Laws \rightarrow World$_2$

Hertz’ Modeling Paradigm

Luis M. Rocha and Santiago Schnell
Encoding in the Modeling Relation

- How to encode data?
  - What is data?
    - Information without context and knowledge
    - Part of Syntax
  - Keeping Numbers
    - The most fundamental need for modeling and information

World

Measure

Data (Symbols)

Encoding (Semantics)
Counting with the Binary System

- **Positional number system**
  - the value of each digit is determined by its position
    - 101 is different from 110
    - The lowest place value is the rightmost position, and each successive position to the left has a higher place value

- **Base 2**
  - The value of each position corresponds to powers of 2
    - \( \ldots d_4d_3d_2d_1d_0 = \ldots + d_4 \times 2^4 + d_3 \times 2^3 + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0 \)
    - Each digit to the left is 2 times the previous digit.
      - \( 111100011 \) (483) = \( 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \)

  - To multiply a number by 2 you can simply shift it to the left by one digit, and fill in the rightmost digit with a 0
    - \( 101 \times 2 = 1010 \) (5*2 = 10)

  - To divide a number by 2, simply shift the number to the right by one digit (moving the decimal place one to the left).
    - \( 101 \div 2 = 10.1 \) (5÷2=2.5)

  - With \( n \) digits, \( 2^n \) unique numbers can be represented
    - If \( n=8 \), 256 (=\(2^8\)) numbers can be represented \(0-11111111\).

\[ \ldots \times n \times n \times n \times \]
\[ [0, (n-1)] \quad [0, (n-1)] \quad [0, (n-1)] \quad [0, (n-1)] \]
Comparing Binary with Decimal

- Binary: Decimal
  - 0000: 00
  - 0001: 01
  - 0010: 02
  - 0011: 03
  - 0100: 04
  - 0101: 05
  - 0110: 06
  - 0111: 07

- Binary: Decimal
  - 1000: 08
  - 1001: 09
  - 1010: 10
  - 1011: 11
  - 1100: 12
  - 1101: 13
  - 1110: 14
  - 1111: 15
Binary Code

- Language with an alphabet of two symbols
  - “0” and “1”, “FALSE” or “TRUE”, etc.
- Most economical way of encoding information
  - Consider a sailor who wants to signal a number between 0 and 127 by means of flags.

Message from Ship

System 1: one flag per number

System 2: decimal system

System 2: binary system

Required Flags

0: 0  1: 1  2: 10  3: 11  4: 100  5: 101  6: 110  7: 111

128 Flags

21 Flags

14 Flags

Luis M. Rocha and Santiago Schnell
Memory: Punch Card

- Binary Representation
  - Holes denote 1’s
  - With 8 holes permissible $2^8 = 256$ numbers possible per column
Converting Binary to Decimal

- $2^8 = 256$
- $2^7 = 128$
- $2^6 = 64$
- $2^5 = 32$
- $2^4 = 16$
- $2^3 = 8$
- $2^2 = 4$
- $2^1 = 2$
- $2^0 = 1$

<table>
<thead>
<tr>
<th>$2^8$</th>
<th>$2^7$</th>
<th>$2^6$</th>
<th>$2^5$</th>
<th>$2^4$</th>
<th>$2^3$</th>
<th>$2^2$</th>
<th>$2^1$</th>
<th>$2^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

$128 + 64 + 8 + 1 = 201$

$...d_4d_3d_2d_1d_0 =$

$... + d_4 \times 2^4 + d_3 \times 2^3 + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0$
Base Conversion

- **Decimal to Binary**
  - **Repeated Division by 2**
    - Divide the decimal number by 2
    - If the remainder is 0, on the side write down a 0
    - If the remainder is 1, write down a 1
    - Continue until the quotient is 0
    - Remainders are written beginning at the least significant digit (right) and each new digit is written to left (the most significant digit) of the previous digit.

<table>
<thead>
<tr>
<th>decimal</th>
<th>quotient</th>
<th>Remain.</th>
<th>binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>14</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>0</td>
<td>010</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>1010</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>11010</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>111010</td>
</tr>
</tbody>
</table>
Dealing with rational numbers

- $2^4 = 16$
- $2^3 = 8$
- $2^2 = 4$
- $2^1 = 2$
- $2^0 = 1$
- $2^{-1} = 0.5$
- $2^{-2} = 0.25$
- $2^{-3} = 0.125$

...$d_2d_1d_0.d_{-1}d_{-2}...=$

$... + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0 + d_{-1} \times 2^{-1} + d_{-2} \times 2^{-2} + ...$
Binary groupings

- **Bit**
  - Size: 1
    - 0-1 (2 Values)

- **Nibble**
  - Size: 4
    - 0-15 (16 Values)
    - 1100

- **Byte**
  - Size: 8
    - 0-255 (256 Values)
    - 10110101

- **Word**
  - Size: 16
    - 0-65535 (65536)
    - 1100000010100101
Binary Arithmetic

- **Addition Rules**
  - $0+0 = 0$, with no carry,
  - $1+0 = 1$, with no carry,
  - $0+1 = 1$, with no carry,
  - $1+1 = 0$, and you carry a 1

\[
\begin{array}{c}
1111 \\
1010 \\
1110 \\
\hline
11000
\end{array} \\
\text{(10+14=24)}
\]

\[
\begin{array}{c}
1 \\
1010 \\
1100 \\
\hline
10110
\end{array} \\
\text{(10+12=22)}
\]
Binary Multiplication

<table>
<thead>
<tr>
<th>1010</th>
<th>1100</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
</tr>
</tbody>
</table>

(10 × 12 = 120)

| 10001100 |

(10 + 14 = 140)
Signed Integers

\[
x = (\pm) \left( b_6 \cdot 2^6 + b_5 \cdot 2^5 + \ldots + b_0 \cdot 2^0 \right)
\]

\[
= (-1) \left( 1 \cdot 2^6 + 0 \cdot 2^5 + \ldots + 0 \cdot 2^0 \right)
\]

\[
= (-1)(64 + 16 + 4)
\]

\[
= -84
\]
Fixed Point Reals

\[
x = b_9 2^5 + b_8 2^4 + \ldots + b_0 2^{-4}
\]
\[
= 1 \cdot 2^5 + 1 \cdot 2^4 + \ldots + 1 \cdot 2^{-4}
\]
\[
= 32 + 16 + 2 + 0.25 + 0.0625
\]
\[
= 50.3125
\]
Floating Point

\[ x = (+/-)(1+F) \cdot 2^{E-B} \]

- **Sign (+/-)**
  - 0 denotes a positive number
  - 1 denotes a negative number

- **Exponent (E-B)**
  - The exponent base (2) is implicit and need not be stored.
  - A *bias* (B) is added to represent both positive and negative exponents.
    - IEEE single-precision floats B=127.
    - If \( E = 127 \), exponent is zero
    - If \( E = 200 \), exponent is \((200-127) = 73\).
    - IEEE double precision, exponent field is 11 bits, and bias is 1023.

- **Mantissa (1 + F)**
  - Fraction (F) plus an implicit leading digit.
Floating Point Reals

\[ x = (+/-)(1+F) \cdot 2^{E-B} \]

\[ = (-)(1+(0.125 + 0.0625)) \cdot 2^{(2+16)-15} \]

\[ = (-)(1.1875) \cdot 2^3 \]

\[ = -8.1875 \]

**Given:**
- length of exponent
- bias (here: 15)

From Cathy Wyss (1308)
Hexadecimal

- **Base 16**
  - 16 symbols: 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F
  - Easy to convert to and from Binary
    - 16 is a power of 2: $16 = 2^4$
    - It takes 4 binary digits for every hexadecimal one
    - Good to represent binary in compressed form!

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bin</th>
<th>Hex</th>
<th>Bin</th>
<th>Hex</th>
<th>Bin</th>
<th>Hex</th>
<th>Bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>4</td>
<td>0100</td>
<td>8</td>
<td>1000</td>
<td>C</td>
<td>1100</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>5</td>
<td>0101</td>
<td>9</td>
<td>1001</td>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>6</td>
<td>0110</td>
<td>A</td>
<td>1010</td>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>7</td>
<td>0111</td>
<td>B</td>
<td>1011</td>
<td>F</td>
<td>1111</td>
</tr>
</tbody>
</table>
Encoding Text

- **ASCII**
  - American Standard Code for Information Interchange
    - between binary numbers and computer and roman symbols
    - Standard to allow computers to communicate textual data
  - Uses 7 bits to encode 128 symbols or characters
    - $2^7 = 128$. It fills a byte, but the 8th bit is used to encode additional symbols for other languages and graphics
  - Usually described in hexadecimal
  - 4 groups of 32 characters
    - 00 to 1F: **control characters**
      - Mostly printer/display operations: *carriage return* (0Dh), *line feed* (0Ah), *back space* (08h), etc.
    - 20 to 3F: punctuation, numeric, and special characters
      - Space (20h), digits 0-9 (30h-39h)
        - Arranged so that by subtracting 30h from the ASCII code for any digit, we obtain the numeric equivalent of the digit
    - 40 to 5F: uppercase letters, plus some special characters
    - 60 to 7F: lowercase letters, plus some special characters and a control character (DEL)
### ASCII Table

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hx</th>
<th>Oct</th>
<th>Char</th>
<th>Dec</th>
<th>Hx</th>
<th>Oct</th>
<th>Char</th>
<th>Dec</th>
<th>Hx</th>
<th>Oct</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>000</td>
<td>BEL (bell)</td>
<td>32</td>
<td>20</td>
<td>040</td>
<td>Space</td>
<td>64</td>
<td>40</td>
<td>100</td>
<td>E0 (ESC)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>001</td>
<td>BEL (start of heading)</td>
<td>33</td>
<td>21</td>
<td>041</td>
<td>Digit 1</td>
<td>65</td>
<td>41</td>
<td>101</td>
<td>E1 (ESI)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>002</td>
<td>BEL (start of text)</td>
<td>34</td>
<td>22</td>
<td>042</td>
<td>Digit 2</td>
<td>66</td>
<td>42</td>
<td>102</td>
<td>E2 (ES2)</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>003</td>
<td>BEL (end of text)</td>
<td>35</td>
<td>23</td>
<td>043</td>
<td>Digit 3</td>
<td>67</td>
<td>43</td>
<td>103</td>
<td>E3 (ES3)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>004</td>
<td>BEL (end of transmission)</td>
<td>36</td>
<td>24</td>
<td>044</td>
<td>Digit 4</td>
<td>68</td>
<td>44</td>
<td>104</td>
<td>E4 (ES4)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>005</td>
<td>BEL (enquiry)</td>
<td>37</td>
<td>25</td>
<td>045</td>
<td>Digit 5</td>
<td>69</td>
<td>45</td>
<td>105</td>
<td>E5 (ES5)</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>006</td>
<td>BEL (acknowledge)</td>
<td>38</td>
<td>26</td>
<td>046</td>
<td>Digit 6</td>
<td>70</td>
<td>46</td>
<td>106</td>
<td>E6 (ES6)</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>007</td>
<td>BEL (acknowledgment)</td>
<td>39</td>
<td>27</td>
<td>047</td>
<td>Digit 7</td>
<td>71</td>
<td>47</td>
<td>107</td>
<td>E7 (ES7)</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>010</td>
<td>BEL (backspace)</td>
<td>40</td>
<td>28</td>
<td>050</td>
<td>Digit 8</td>
<td>72</td>
<td>48</td>
<td>110</td>
<td>E8 (ES8)</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>011</td>
<td>BEL (horizontal tab)</td>
<td>41</td>
<td>29</td>
<td>051</td>
<td>Digit 9</td>
<td>73</td>
<td>49</td>
<td>111</td>
<td>EX (ESC)</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>012</td>
<td>BEL (vertical tab)</td>
<td>42</td>
<td>30</td>
<td>052</td>
<td>Digit A</td>
<td>74</td>
<td>50</td>
<td>112</td>
<td>EY (ESy)</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>013</td>
<td>BEL (form feed, new page)</td>
<td>43</td>
<td>31</td>
<td>053</td>
<td>Digit B</td>
<td>75</td>
<td>51</td>
<td>113</td>
<td>EZ (ESz)</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>014</td>
<td>BEL (carriage return)</td>
<td>44</td>
<td>32</td>
<td>054</td>
<td>Digit C</td>
<td>76</td>
<td>52</td>
<td>114</td>
<td>ESC (ESC)</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>015</td>
<td>BEL (shift out)</td>
<td>45</td>
<td>33</td>
<td>055</td>
<td>Digit D</td>
<td>77</td>
<td>53</td>
<td>115</td>
<td>E (ESC)</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>016</td>
<td>BEL (shift in)</td>
<td>46</td>
<td>34</td>
<td>056</td>
<td>Digit E</td>
<td>78</td>
<td>54</td>
<td>116</td>
<td>\ (ESC)</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>017</td>
<td>BEL (shift lock)</td>
<td>47</td>
<td>35</td>
<td>057</td>
<td>Digit F</td>
<td>79</td>
<td>55</td>
<td>117</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>030</td>
<td>BEL (data link escape)</td>
<td>48</td>
<td>36</td>
<td>060</td>
<td>Digit 10</td>
<td>80</td>
<td>56</td>
<td>120</td>
<td>` (ESC)</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>031</td>
<td>BEL (device control 1)</td>
<td>49</td>
<td>37</td>
<td>061</td>
<td>Digit 11</td>
<td>81</td>
<td>57</td>
<td>121</td>
<td>' (ESC)</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>032</td>
<td>BEL (device control 2)</td>
<td>50</td>
<td>38</td>
<td>062</td>
<td>Digit 12</td>
<td>82</td>
<td>58</td>
<td>122</td>
<td>( (ESC)</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>033</td>
<td>BEL (device control 3)</td>
<td>51</td>
<td>39</td>
<td>063</td>
<td>Digit 13</td>
<td>83</td>
<td>59</td>
<td>123</td>
<td>) (ESC)</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>034</td>
<td>BEL (device control 4)</td>
<td>52</td>
<td>40</td>
<td>064</td>
<td>Digit 14</td>
<td>84</td>
<td>60</td>
<td>124</td>
<td>¬ (ESC)</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>035</td>
<td>BEL (negative acknowledge)</td>
<td>53</td>
<td>41</td>
<td>065</td>
<td>Digit 15</td>
<td>85</td>
<td>61</td>
<td>125</td>
<td>_ (ESC)</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>036</td>
<td>BEL (synchronous idle)</td>
<td>54</td>
<td>42</td>
<td>066</td>
<td>Digit 16</td>
<td>86</td>
<td>62</td>
<td>126</td>
<td>- (ESC)</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>037</td>
<td>BEL (end of trans. block)</td>
<td>55</td>
<td>43</td>
<td>067</td>
<td>Digit 17</td>
<td>87</td>
<td>63</td>
<td>127</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>038</td>
<td>BEL (cancel)</td>
<td>56</td>
<td>44</td>
<td>068</td>
<td>Digit 18</td>
<td>88</td>
<td>64</td>
<td>130</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>039</td>
<td>BEL (end of medium)</td>
<td>57</td>
<td>45</td>
<td>069</td>
<td>Digit 19</td>
<td>89</td>
<td>65</td>
<td>131</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>032</td>
<td>BEL (substitute)</td>
<td>58</td>
<td>46</td>
<td>070</td>
<td>Digit 20</td>
<td>90</td>
<td>66</td>
<td>132</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>033</td>
<td>BEL (escape)</td>
<td>59</td>
<td>47</td>
<td>071</td>
<td>Digit 21</td>
<td>91</td>
<td>67</td>
<td>133</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>034</td>
<td>BEL (file separator)</td>
<td>60</td>
<td>48</td>
<td>072</td>
<td>Digit 22</td>
<td>92</td>
<td>6C</td>
<td>134</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>035</td>
<td>BEL (group separator)</td>
<td>61</td>
<td>49</td>
<td>073</td>
<td>Digit 23</td>
<td>93</td>
<td>6D</td>
<td>135</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>036</td>
<td>BEL (record separator)</td>
<td>62</td>
<td>4A</td>
<td>074</td>
<td>Digit 24</td>
<td>94</td>
<td>6E</td>
<td>136</td>
<td>^ (ESC)</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>037</td>
<td>BEL (unit separator)</td>
<td>63</td>
<td>4F</td>
<td>077</td>
<td>Digit 25</td>
<td>95</td>
<td>6F</td>
<td>137</td>
<td>^ (ESC)</td>
</tr>
</tbody>
</table>

Source: www.asciiTable.com

Luis M. Rocha and Santiago Schnell
# Extended ASCII Table

<table>
<thead>
<tr>
<th>Code</th>
<th>Character</th>
<th>Code</th>
<th>Character</th>
<th>Code</th>
<th>Character</th>
<th>Code</th>
<th>Character</th>
<th>Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>Ç</td>
<td>144</td>
<td>é</td>
<td>160</td>
<td>€</td>
<td>176</td>
<td>✏</td>
<td>193</td>
<td>✈</td>
</tr>
<tr>
<td>129</td>
<td>ü</td>
<td>145</td>
<td>ö</td>
<td>161</td>
<td>í</td>
<td>177</td>
<td>✖</td>
<td>194</td>
<td>✧</td>
</tr>
<tr>
<td>130</td>
<td>á</td>
<td>146</td>
<td>â</td>
<td>162</td>
<td>ò</td>
<td>178</td>
<td>✱</td>
<td>195</td>
<td>☀</td>
</tr>
<tr>
<td>131</td>
<td>à</td>
<td>147</td>
<td>ø</td>
<td>163</td>
<td>ā</td>
<td>179</td>
<td>✲</td>
<td>196</td>
<td>✢</td>
</tr>
<tr>
<td>132</td>
<td>ë</td>
<td>148</td>
<td>ŏ</td>
<td>164</td>
<td>ë</td>
<td>180</td>
<td>✳</td>
<td>197</td>
<td>✣</td>
</tr>
<tr>
<td>133</td>
<td>è</td>
<td>149</td>
<td>ò</td>
<td>165</td>
<td>ñ</td>
<td>181</td>
<td>✷</td>
<td>198</td>
<td>✦</td>
</tr>
<tr>
<td>134</td>
<td>å</td>
<td>150</td>
<td>ä</td>
<td>166</td>
<td>ā</td>
<td>182</td>
<td>✹</td>
<td>199</td>
<td>✩</td>
</tr>
<tr>
<td>135</td>
<td>ç</td>
<td>151</td>
<td>ň</td>
<td>167</td>
<td>ë</td>
<td>183</td>
<td>✵</td>
<td>200</td>
<td>✪</td>
</tr>
<tr>
<td>136</td>
<td>ñ</td>
<td>152</td>
<td>ñ</td>
<td>168</td>
<td>ñ</td>
<td>184</td>
<td>✭</td>
<td>201</td>
<td>✫</td>
</tr>
<tr>
<td>137</td>
<td>ï</td>
<td>153</td>
<td>ï</td>
<td>169</td>
<td>ï</td>
<td>185</td>
<td>✮</td>
<td>202</td>
<td>✬</td>
</tr>
<tr>
<td>138</td>
<td>â</td>
<td>154</td>
<td>ã</td>
<td>170</td>
<td>ã</td>
<td>186</td>
<td>✯</td>
<td>203</td>
<td>✯</td>
</tr>
<tr>
<td>139</td>
<td>ë</td>
<td>155</td>
<td>ë</td>
<td>171</td>
<td>ë</td>
<td>187</td>
<td>✰</td>
<td>204</td>
<td>✰</td>
</tr>
<tr>
<td>140</td>
<td>í</td>
<td>156</td>
<td>è</td>
<td>172</td>
<td>è</td>
<td>188</td>
<td>✱</td>
<td>205</td>
<td>✱</td>
</tr>
<tr>
<td>141</td>
<td>ı</td>
<td>157</td>
<td>ñ</td>
<td>173</td>
<td>ñ</td>
<td>189</td>
<td>✲</td>
<td>206</td>
<td>✲</td>
</tr>
<tr>
<td>142</td>
<td>å</td>
<td>158</td>
<td>å</td>
<td>174</td>
<td>å</td>
<td>190</td>
<td>✳</td>
<td>207</td>
<td>✳</td>
</tr>
<tr>
<td>143</td>
<td>ö</td>
<td>160</td>
<td>ö</td>
<td>175</td>
<td>ö</td>
<td>191</td>
<td>✴</td>
<td>208</td>
<td>✴</td>
</tr>
</tbody>
</table>

Sources: www.pubblinet.com

Luis M.Rocha and Santiago Schnell
Alternative Extended ASCII
Hi, my honey. I'm really happy that my mailbox is full of those pretty hearts every day. So, I just thought I would return the favor, just in case you'd not yet realized just how I love you. You are just so very, very, very extraordinarily special and I adore you!
Unicode

- Extends ASCII
  - Much greater support for international characters, glyphs, math symbols, etc.

- Universal Code for Text
  - Each character has a single and unique code in every computer everywhere
    - code point
  - Initially using 16-bits
    - 65536 possible code points
    - Sufficient space to include all the characters for every language on the planet
  - Characters organized into different ranges
    - Greek stored between 880 and 1023 (0x370 and 0x3FF)
  - Accepted by the International Standards Organization (ISO)
  - Version 3.1 in 2001 was expanded to 21-bits
    - over 1 million different code points
    - Logical "planes" contain broad classes of characters
<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>0A</th>
<th>0B</th>
<th>0C</th>
<th>0D</th>
<th>0E</th>
<th>0F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>NUL</td>
<td>STX</td>
<td>SOT</td>
<td>ETX</td>
<td>EOT</td>
<td>ENQ</td>
<td>ACK</td>
<td>BEL</td>
<td>BS</td>
<td>HT</td>
<td>LF</td>
<td>VT</td>
<td>FF</td>
<td>CR</td>
<td>SO</td>
<td>SI</td>
</tr>
<tr>
<td>10</td>
<td>DLE</td>
<td>DC1</td>
<td>DC2</td>
<td>DC3</td>
<td>DC4</td>
<td>NAK</td>
<td>SYN</td>
<td>ETB</td>
<td>CAN</td>
<td>EM</td>
<td>SUB</td>
<td>ESC</td>
<td>FS</td>
<td>GS</td>
<td>RS</td>
<td>US</td>
</tr>
<tr>
<td>20</td>
<td>SP</td>
<td>!</td>
<td>&quot;</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>'</td>
<td>(</td>
<td>)</td>
<td>*</td>
<td>+</td>
<td>,</td>
<td>-</td>
<td>.</td>
<td>/</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>;</td>
<td>&lt;</td>
<td>=</td>
<td>&gt;</td>
<td>?</td>
</tr>
<tr>
<td>40</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
</tr>
<tr>
<td>60</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>{</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
</tr>
<tr>
<td>90</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>{</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://www.unicode.org/charts/
Markup Languages

- In addition to the symbols, specify formatting, hyperlinks, images, media, etc.
- represents text as well as details about the structure and appearance of the text
- SGML: Standard Generalized Markup Language
  - Specifies a syntax for including the markup in documents, as well as a description of what the markup meant
- HTML: HyperText Markup Language
  - Does not require a definition of what the markup means
- XML: Extensible Markup Language
  - Allows the creation of special-purpose markup languages
  - Simplified subset of SGML, also requiring a definition of what the markup means
  - Can describing many different kinds of data
- LaTeX:
  - best way to typeset complex mathematical formulas
More about data representation

- **I308: Information Representation**
- **C.M. Wyss**
- The basic structure of information representation in social and scientific applications.
- Information access and representation on the World Wide Web; object-oriented design and relational databases; AI knowledge representation and discovery.
Next Class!

Topics
- Encoding Multimedia

Readings for Next week
- Lecture notes Posted online @ http://informatics.indiana.edu/rocha/i101
  - Modeling the World
- @ infoport
  - Read Binary encoding resources at Infoport!!
- From course package
  - From Irv Englander’s book “The Architecture of Computer Hardware and Systems Software”
  - Chapter 3: Data Formats (pp. 70-86)

Lab 6
- More Unix and FTP