The Hardware/Software Interface
CSE 351 Autumn 2022

Instructor:
Justin Hsia

Teaching Assistants:
Angela Xu
Arjun Narendra
Armin Magness
Assaf Vayner
Carrie Hu
Clare Edmonds
David Dai
Dominick Ta
Effie Zheng
James Froelich
Jenny Peng
Kristina Lansang
Paul Stevans
Renee Ruan
Vincent Xiao

http://xkcd.com/676/
Lecture Outline

- **Course Introduction**
- **Course Policies**
- **Binary and Numerical Representation**
Introductions: Course Staff

❖ Instructor: just call me Justin
  ▪ CSE Associate Teaching Professor
  ▪ Raising a toddler this quarter, will be tired

❖ TAs:
  ▪ Available in section, office hours, and on Ed Discussion

❖ More than anything, we want you to feel...
  ✓ Comfortable and welcome in this space
  ✓ Able to learn and succeed in this course
  ✓ Comfortable reaching out if you need help or want change
Introductions: You!

- ~320 students registered, split across two lectures
- CSE majors, ECE majors, and more
  - Most of you will find almost everything in the course new
  - Many of you are new to CSE and/or UW (and campus)!

Get to know each other! Help each other out!

- Science says that learning happens best in groups
- Working well with others is a valuable life skill
- Diversity of perspectives expands your horizons
- Take advantage of group work, where permissible, to learn, not just get a grade
Welcome to CSE351!

❖ Our goal is to teach you the key abstractions “under the hood”

- How does your source code become something that your computer understands?
- What happens as your computer is executing one or more processes?
Layers of Computing Below Programming

Software Applications
(written in Java, Python, C, etc.)

Programming Languages & Libraries
\(\text{\textit{e.g.}},\) Java Runtime Env, C Standard Lib

Operating System
\(\text{\textit{e.g.}},\) Linux, MacOS, Windows

Hardware
\(\text{\textit{e.g.}},\) CPU, memory, disk, network, peripherals
“House” of Computing Metaphor

- We continue to build upward but everything relies on the base & foundation
  - We’ll explore parts of Hardware, OS, and PL

- Built a long time ago
  - Some parts have been updated over the years, some have not
  - More remodeling necessary, but should understand how and why things are this way before demolishing anything
The Hardware/Software Interface

- Topic Group 1: Data
  - Memory, Data, Integers, Floating Point, Arrays, Structs

- Topic Group 2: Programs
  - x86-64 Assembly, Procedures, Stacks, Executables

- Topic Group 3: Scale & Coherence
  - Caches, Processes, Virtual Memory, Memory Allocation

- Learning in this class
  - You might miss Java, but we just ask you to keep your heart open; something unexpected might pique your interest!
  - Notice and nurture any wants to linger in some space
    - Many future classes to explore this space more
Some fun topics that we will touch on

- Which of the following seems the most interesting to you? (vote in Ed Lessons)
  
a) What is a GFLOP and why is it used in computer benchmarks?
b) How and why does running many programs for a long time eat into your memory (RAM)?
c) What is stack overflow and how does it happen?
d) Why does your computer slow down when you run out of disk space?
e) What was the flaw behind the original Internet worm, the Heartbleed bug, and the Cloudbleed bug?
f) What is the meaning behind the different CPU specifications? (e.g., # of cores, size of cache)
Lecture Outline

❖ Course Introduction

❖ **Course Policies**

❖ Binary and Numerical Representation
Bookmarks

❖ Website:  https://courses.cs.washington.edu/courses/cse351/22au/
  ▪ Schedule, policies, materials, videos, assignment specs, etc.

❖ Ed Course:  https://edstem.org/us/courses/23927
  ▪ Discussion:  announcements, ask and answer questions
  ▪ Lessons:  readings, lecture questions, homework
  ▪ Resources:  links to other tools and information

❖ Linked from website and Ed
  ▪ Canvas:  surveys, grade book, Zoom links
  ▪ Gradescope:  lab submissions, take-home exams
  ▪ Panopto:  lecture recordings
Grading

❖ **Pre-lecture Readings:** 5%  
  - Can reveal solution after one attempt (completion)

❖ **Homework:** 20% total  
  - Unlimited submission attempts (autograded correctness)

❖ **Labs:** 40% total  
  - Last submission graded (correctness)

❖ **Exams:** Midterm (16%) and Final (16%)  
  - Take-home; individual, but some discussion permitted

❖ **EPA:** Effort, Participation, and Altruism (3%)
Group Work in 351

❖ Group work will be *emphasized* in this class
  ▪ Lecture and section will have built-in group work time – you will get the most out of it if you actively participate!
    • TAs will circle around the room and interact with groups
    • Raise your hand to get the attention of a staff member
  ▪ Most assignments allow collaboration – talking to classmates will help you synthesize concepts and terminology
    • *The major takeaways for this course will be the ability to explain the major concepts verbally and/or in writing to others*
  ▪ However, the responsibility for learning falls on *you*
Lab Collaboration and Academic Integrity

- All submissions are expected to be yours and yours alone
- You are encouraged to discuss your assignments with other students (*ideas*), but we expect that what you turn in is yours
- It is NOT acceptable to copy solutions from other students or to copy (or start your) solutions from the Web (including Github, Chegg, and similar sites)
- Our goal is that *YOU* learn the material so you will be prepared for exams, interviews, and the future
Office Hours

- Check Weekly Calendar on website for scheduled office hours:
  - In-person or virtual, but NOT hybrid
  - Zoom meeting links found in Zoom tab within Canvas
- All office hours will use a Google Sheets queue:
  - Fill out first 3 columns to enter queue:
- We encourage you to chat with other students if the TAs are busy!
In-Person Office Hours

- Allen 3rd floor breakout
  - Up the stairs in the CSE Atrium (Allen Center, not Gates)
  - At the top of two flights, the open area with the whiteboard wall is the 3rd floor breakout!
Extensions, Accommodations, Help

❖ Extenuating circumstances
  ▪ Students (and staff) face an extremely varied set of environments and circumstances
  ▪ For formal accommodations, go through Disability Resources for Students (DRS)
  ▪ We will try to be accommodating otherwise, but the earlier you reach out, the better

❖ Don’t suffer in silence – talk to a staff member!
  ▪ We have a 1-on-1 meeting request form
To-Do List

❖ Admin
  ▪ Explore/read the course website *thoroughly*
  ▪ Check that you can access Ed Discussion & Lessons
  ▪ Get your machine set up to access the CSE Linux environment (CSE VM or attu) *as soon as possible*
  ▪ Optionally, sign up for CSE 391: System and Software Tools

❖ Assignments
  ▪ Pre-Course Survey and hw0 due Friday (9/30)
  ▪ hw1 and Lab 0 due Monday (10/3)
  ▪ Pre-lecture readings due before each lecture – 11 am
Lecture Outline

❖ Course Introduction

❖ Course Policies
  ▪ Return to in-person instruction

❖ Binary and Numerical Representation
Reading Review

❖ Terminology:
  ▪ numeral, digit, base, symbol, digit position, leading zeros
  ▪ binary, bit, nibble, byte, hexadecimal
  ▪ numerical representation, encoding scheme

❖ Questions from the Reading?
Review Questions

❖ What is the decimal value of the numeral 107₈?
   A. 71
   B. 87
   C. 107
   D. 568

❖ What is the decimal number 108 in hex?
   A. 0x6C
   B. 0xA8
   C. 0x108
   D. 0x612

❖ Represent 0b100110110101101 in hex.

❖ Represent 0x3C9 in binary.
Base Comparison

❖ Why does all of this matter?
  ▪ *Humans* think about numbers in **base 10**, but *computers* “think” about numbers in **base 2**
  ▪ **Binary encoding** is what allows computers to do all of the amazing things that they do!

❖ You should have this table memorized by the end of the class
  ▪ Might as well start now!

<table>
<thead>
<tr>
<th>Base 10</th>
<th>Base 2</th>
<th>Base 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>E</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>F</td>
</tr>
</tbody>
</table>
Numerical Encoding

❖ AMAZING FACT: You can represent anything countable using numbers!
  - Need to agree on an encoding
  - Kind of like learning a new language

❖ Examples:
  - Decimal Integers: 0→0b0, 1→0b1, 2→0b10, etc.
  - English Letters: CSE→0x435345, yay→0x796179
  - Emoticons: 😃 0x0, 😞 0x1, 😊 0x2, 😎 0x3, 😈 0x4, 🙋 0x5
Binary Encoding

❖ With \( n \) binary digits, how many “things” can you represent?

- Need \( n \) binary digits to represent \( N \) things, where \( 2^n \geq N \)
- **Example**: 5 binary digits for alphabet because \( 2^5 = 32 > 26 \)

❖ A binary digit is known as a **bit**

❖ A group of 4 bits (1 hex digit) is called a **nibble**

❖ A group of 8 bits (2 hex digits) is called a **byte**

- 1 bit → 2 things, 1 nibble → 16 things, 1 byte → 256 things
So What’s It Mean?

❖ A sequence of bits can have many meanings!

❖ Consider the hex sequence 0x4E6F21
  ▪ Common interpretations include:
    • The decimal number 5140257
    • The real number $7.203034 \times 10^{-39}$
    • The characters “No!”
    • The background color of this slide

❖ It is up to the program/programmer to decide how to interpret the sequence of bits
## Binary Encoding – Characters/Text

- **ASCII Encoding** ([www.asciitable.com](http://www.asciitable.com))
  - **American Standard Code for Information Interchange**

<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>Html</th>
<th>Char</th>
<th>Dec</th>
<th>Hx</th>
<th>Oct</th>
<th>Html</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>NUL (null)</td>
<td>32</td>
<td>20</td>
<td>040</td>
<td>&lt;#32;</td>
<td>Space</td>
<td>64</td>
<td>40</td>
<td>100</td>
<td>&lt;#64;</td>
<td>@</td>
<td>96</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>SOH (start of heading)</td>
<td>33</td>
<td>21</td>
<td>041</td>
<td>&lt;#33;</td>
<td>!</td>
<td>65</td>
<td>41</td>
<td>101</td>
<td>&lt;#65;</td>
<td>A</td>
<td>97</td>
</tr>
<tr>
<td>2</td>
<td>002</td>
<td>STX (start of text)</td>
<td>34</td>
<td>22</td>
<td>042</td>
<td>&lt;#34;</td>
<td>&quot;</td>
<td>66</td>
<td>42</td>
<td>102</td>
<td>&lt;#66;</td>
<td>B</td>
<td>98</td>
</tr>
<tr>
<td>3</td>
<td>003</td>
<td>ETX (end of text)</td>
<td>35</td>
<td>23</td>
<td>043</td>
<td>&lt;#35;</td>
<td>#</td>
<td>67</td>
<td>43</td>
<td>103</td>
<td>&lt;#67;</td>
<td>C</td>
<td>99</td>
</tr>
<tr>
<td>4</td>
<td>004</td>
<td>EOT (end of transmission)</td>
<td>36</td>
<td>24</td>
<td>044</td>
<td>&lt;#36;</td>
<td></td>
<td>68</td>
<td>44</td>
<td>104</td>
<td>&lt;#68;</td>
<td>D</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>005</td>
<td>ENQ (enquiry)</td>
<td>37</td>
<td>25</td>
<td>045</td>
<td>&lt;#37;</td>
<td>%</td>
<td>69</td>
<td>45</td>
<td>105</td>
<td>&lt;#69;</td>
<td>E</td>
<td>101</td>
</tr>
<tr>
<td>6</td>
<td>006</td>
<td>ACK (acknowledge)</td>
<td>38</td>
<td>26</td>
<td>046</td>
<td>&lt;#38;</td>
<td></td>
<td>70</td>
<td>46</td>
<td>106</td>
<td>&lt;#70;</td>
<td>F</td>
<td>102</td>
</tr>
<tr>
<td>7</td>
<td>007</td>
<td>BEL (bell)</td>
<td>39</td>
<td>27</td>
<td>047</td>
<td>&lt;#39;</td>
<td></td>
<td>71</td>
<td>47</td>
<td>107</td>
<td>&lt;#71;</td>
<td>G</td>
<td>103</td>
</tr>
<tr>
<td>8</td>
<td>010</td>
<td>BS (backspace)</td>
<td>40</td>
<td>28</td>
<td>050</td>
<td>&lt;#40;</td>
<td></td>
<td>72</td>
<td>48</td>
<td>110</td>
<td>&lt;#72;</td>
<td>H</td>
<td>104</td>
</tr>
<tr>
<td>9</td>
<td>011</td>
<td>TAB (horizontal tab)</td>
<td>41</td>
<td>29</td>
<td>051</td>
<td>&lt;#41;</td>
<td></td>
<td>73</td>
<td>49</td>
<td>111</td>
<td>&lt;#73;</td>
<td>I</td>
<td>105</td>
</tr>
<tr>
<td>10</td>
<td>A12</td>
<td>LF (NL line feed, new line)</td>
<td>42</td>
<td>2A</td>
<td>052</td>
<td>&lt;#42;</td>
<td>*</td>
<td>74</td>
<td>4A</td>
<td>112</td>
<td>&lt;#74;</td>
<td>J</td>
<td>106</td>
</tr>
<tr>
<td>11</td>
<td>B13</td>
<td>VT (vertical tab)</td>
<td>43</td>
<td>2B</td>
<td>053</td>
<td>&lt;#43;</td>
<td></td>
<td>75</td>
<td>4B</td>
<td>113</td>
<td>&lt;#75;</td>
<td>K</td>
<td>107</td>
</tr>
<tr>
<td>12</td>
<td>C14</td>
<td>FF (NP form feed, new page)</td>
<td>44</td>
<td>2C</td>
<td>054</td>
<td>&lt;#44;</td>
<td></td>
<td>76</td>
<td>4C</td>
<td>114</td>
<td>&lt;#76;</td>
<td>L</td>
<td>108</td>
</tr>
<tr>
<td>13</td>
<td>D15</td>
<td>CR (carriage return)</td>
<td>45</td>
<td>2D</td>
<td>055</td>
<td>&lt;#45;</td>
<td></td>
<td>77</td>
<td>4D</td>
<td>115</td>
<td>&lt;#77;</td>
<td>M</td>
<td>109</td>
</tr>
<tr>
<td>14</td>
<td>E16</td>
<td>SO (shift out)</td>
<td>46</td>
<td>2E</td>
<td>056</td>
<td>&lt;#46;</td>
<td></td>
<td>78</td>
<td>4E</td>
<td>116</td>
<td>&lt;#78;</td>
<td>N</td>
<td>110</td>
</tr>
<tr>
<td>15</td>
<td>F17</td>
<td>SI (shift in)</td>
<td>47</td>
<td>2F</td>
<td>057</td>
<td>&lt;#47;</td>
<td></td>
<td>79</td>
<td>4F</td>
<td>117</td>
<td>&lt;#79;</td>
<td>O</td>
<td>111</td>
</tr>
<tr>
<td>16</td>
<td>102</td>
<td>DLE (data link escape)</td>
<td>060</td>
<td>40</td>
<td>060</td>
<td>&lt;#60;</td>
<td>0</td>
<td>80</td>
<td>50</td>
<td>120</td>
<td>&lt;#80;</td>
<td>P</td>
<td>112</td>
</tr>
<tr>
<td>17</td>
<td>101</td>
<td>DC1 (device control 1)</td>
<td>061</td>
<td>41</td>
<td>061</td>
<td>&lt;#61;</td>
<td>1</td>
<td>81</td>
<td>51</td>
<td>121</td>
<td>&lt;#81;</td>
<td>Q</td>
<td>113</td>
</tr>
<tr>
<td>18</td>
<td>102</td>
<td>DC2 (device control 2)</td>
<td>062</td>
<td>42</td>
<td>062</td>
<td>&lt;#62;</td>
<td>2</td>
<td>82</td>
<td>52</td>
<td>122</td>
<td>&lt;#82;</td>
<td>R</td>
<td>114</td>
</tr>
<tr>
<td>19</td>
<td>103</td>
<td>DC3 (device control 3)</td>
<td>063</td>
<td>43</td>
<td>063</td>
<td>&lt;#63;</td>
<td>3</td>
<td>83</td>
<td>53</td>
<td>123</td>
<td>&lt;#83;</td>
<td>S</td>
<td>115</td>
</tr>
<tr>
<td>20</td>
<td>104</td>
<td>DC4 (device control 4)</td>
<td>064</td>
<td>44</td>
<td>064</td>
<td>&lt;#64;</td>
<td>4</td>
<td>84</td>
<td>54</td>
<td>124</td>
<td>&lt;#84;</td>
<td>T</td>
<td>116</td>
</tr>
<tr>
<td>21</td>
<td>105</td>
<td>NAK (negative acknowledge)</td>
<td>065</td>
<td>45</td>
<td>065</td>
<td>&lt;#65;</td>
<td>5</td>
<td>85</td>
<td>55</td>
<td>125</td>
<td>&lt;#85;</td>
<td>U</td>
<td>117</td>
</tr>
<tr>
<td>22</td>
<td>106</td>
<td>SYN (synchronous idle)</td>
<td>066</td>
<td>46</td>
<td>066</td>
<td>&lt;#66;</td>
<td>6</td>
<td>86</td>
<td>56</td>
<td>126</td>
<td>&lt;#86;</td>
<td>V</td>
<td>118</td>
</tr>
<tr>
<td>23</td>
<td>107</td>
<td>ETB (end of trans. block)</td>
<td>067</td>
<td>47</td>
<td>067</td>
<td>&lt;#67;</td>
<td>7</td>
<td>87</td>
<td>57</td>
<td>127</td>
<td>&lt;#87;</td>
<td>W</td>
<td>119</td>
</tr>
<tr>
<td>24</td>
<td>103</td>
<td>CAN (cancel)</td>
<td>070</td>
<td>50</td>
<td>070</td>
<td>&lt;#70;</td>
<td>8</td>
<td>88</td>
<td>58</td>
<td>130</td>
<td>&lt;#88;</td>
<td>X</td>
<td>120</td>
</tr>
<tr>
<td>25</td>
<td>101</td>
<td>EM (end of medium)</td>
<td>071</td>
<td>51</td>
<td>071</td>
<td>&lt;#71;</td>
<td>9</td>
<td>89</td>
<td>59</td>
<td>131</td>
<td>&lt;#89;</td>
<td>Y</td>
<td>121</td>
</tr>
<tr>
<td>26</td>
<td>A32</td>
<td>SUB (substitute)</td>
<td>072</td>
<td>52</td>
<td>072</td>
<td>&lt;#72;</td>
<td></td>
<td>90</td>
<td>5A</td>
<td>132</td>
<td>&lt;#90;</td>
<td>Z</td>
<td>122</td>
</tr>
<tr>
<td>27</td>
<td>B33</td>
<td>ESC (escape)</td>
<td>073</td>
<td>53</td>
<td>073</td>
<td>&lt;#73;</td>
<td></td>
<td>91</td>
<td>5B</td>
<td>133</td>
<td>&lt;#91;</td>
<td></td>
<td>123</td>
</tr>
<tr>
<td>28</td>
<td>C34</td>
<td>FS (file separator)</td>
<td>074</td>
<td>54</td>
<td>074</td>
<td>&lt;#74;</td>
<td></td>
<td>92</td>
<td>5C</td>
<td>134</td>
<td>&lt;#92;</td>
<td></td>
<td>124</td>
</tr>
<tr>
<td>29</td>
<td>D35</td>
<td>GS (group separator)</td>
<td>075</td>
<td>55</td>
<td>075</td>
<td>&lt;#75;</td>
<td></td>
<td>93</td>
<td>5D</td>
<td>135</td>
<td>&lt;#93;</td>
<td></td>
<td>125</td>
</tr>
<tr>
<td>30</td>
<td>E36</td>
<td>RS (record separator)</td>
<td>076</td>
<td>56</td>
<td>076</td>
<td>&lt;#76;</td>
<td></td>
<td>94</td>
<td>5E</td>
<td>136</td>
<td>&lt;#94;</td>
<td></td>
<td>126</td>
</tr>
<tr>
<td>31</td>
<td>F37</td>
<td>US (unit separator)</td>
<td>077</td>
<td>57</td>
<td>077</td>
<td>&lt;#77;</td>
<td></td>
<td>95</td>
<td>5F</td>
<td>137</td>
<td>&lt;#95;</td>
<td></td>
<td>127</td>
</tr>
</tbody>
</table>

What’s Missing? [DEL]
Binary Encoding – Characters/Text

❖ ASCII Encoding ([www.asciitable.com](http://www.asciitable.com))
  ▪ *American* Standard Code for Information Interchange

❖ Created in 1963
  ▪ Memory was expensive, 32KB in brand new machines
  ▪ *Economic incentive* to use fewer bits for encoding

❖ **Design Goals:**
  ▪ Represent everything on an *American* typewriter as *efficiently* as possible
  ▪ Organize similar characters together
    • Numbers, uppercase, lowercase, then other stuff
Binary Encoding – Unicode & Emoji

- Unicode Standard is managed by the Unicode Consortium
  - “Universal language” that uses 1-4 bytes to represent a much larger range of characters/languages, including emoji
  - Adds new emojis every year, though adoption often lags: 🎒
    - [https://emojipedia.org/new/](https://emojipedia.org/new/)

- Emojipedia demo: [http://www.emojipedia.org](http://www.emojipedia.org)
  - Desktop Computer: 🖥️
  - Code points: U+1F5A5, U+FE0F
  - Display:
Binary Encoding – Colors

❖ RGB – Red, Green, Blue
  ▪ Additive color model (light): byte (8 bits) for each color
  ▪ Commonly seen in hex (in HTML, photo editing, etc.)
  ▪ Examples: Blue → 0x0000FF, Gold → 0xFFD700,
    White → 0xFFFFFFF, Deep Pink → 0xFFF1493
Binary Encoding – Files and Programs

❖ At the lowest level, all digital data is stored as bits!

❖ Layers of abstraction keep everything comprehensible
  ▪ Data/files are groups of bits interpreted by program
  ▪ Program is actually groups of bits being interpreted by your CPU

❖ Computer Memory Demo (if time)
  ▪ Linux tool: `xxd`
Summary

❖ Humans think about numbers in decimal; computers think about numbers in binary
  ▪ Base conversion to go between them
  ▪ Hexadecimal is more human-readable than binary

❖ All information on a computer is binary

❖ Binary encoding can represent anything!
  ▪ Computer/program needs to know how to interpret the bits
  ▪ Encodings aren’t “neutral”; priorities are baked in