Chapter 2: Metamorphosis of Information

How does the computer store information?

Metamorphosis of Information

- In this lecture:
  - What are the common types of information that can be manipulated by the computer?
  - Why does the computer use binary numbers?
  - How does the computer deal with numbers, text, pictures, sound, and programs?
  - What type of program manipulates text?

What is Information?

- A chess board diagram.
- Satellite photos of the surface of Mars.
- The fuel capacity of a Boeing 747.
- A Tarzan yell.
- A computer program.
- The fingerprint files of a police department.
- Your name and address.
- The script of Gone with the Wind.
- A Bach fugue in 4 parts.
- The value of \( \pi \) to 100,000 decimal places.
- A recipe for Quiche Lorraine.
- An automotive service handbook.
- Your favorite song.
- A recording of bird-calls.

What is Information?

- The five types of information the computer commonly manipulates:
  - Numeric
  - Character
  - Visual
  - Audio
  - Instructional

- First, the information must be transformed (converted) into an acceptable representation that the computer will accept.
- That format: NUMBERS
What is Information?

- All modern computers work with a system of numbers called binary numbers.
  - Use only two symbols: 0 and 1.

- Binary circuits: Electronic devices are cheapest and function most reliably if they assume only two states.

Representation of Numbers

- The three-light system:
  - Has eight possible combinations of on and off.
  - Could be used to indicate the numbers 0, 1, 2, 3, 4, 5, 6, 7.

<table>
<thead>
<tr>
<th>Number</th>
<th>Three-Light Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
</tr>
</tbody>
</table>

- Decimal numeration system:
  - Uses 10 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.
  - The place values of each position are powers of ten.
  - A number such as 1357 will be expanded as:
    \[
    1357 = (1 \times 1000) + (3 \times 100) + (5 \times 10) + (7 \times 1)
    \]

- Binary numeration system:
  - Uses 2 symbols: 0, and 1. Each is called a bit for binary digit.
  - The place values of each position are powers of two.
  - A binary number such as 10110 will be expanded as:
    \[
    10110 = (1 \times 16) + (0 \times 8) + (1 \times 4) + (1 \times 2) + (0 \times 1)
    \]

<table>
<thead>
<tr>
<th>Number</th>
<th>Binary Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
</tr>
</tbody>
</table>

- Only 22 in the decimal system!
Representation of Numbers

- **Decimal**
  - Each number has a unique representation.
  - Counting:
    - When you run out of digits, make it a zero and increment the next place value to the left.
    - 99 becomes 100

- **Binary**
  - Each number has a unique representation.
  - Counting:
    - When you run out of digits, make it a zero and increment the next place value to the left.
    - 111 becomes 1000

Representation of Symbols and Text

- To store any kind of information in the computer’s memory, it must first be transformed into a binary numeric form.
- **Symbols and Text**
  - Includes characters, punctuation, symbols representing numbers.
  - Each symbol can be assigned a numeric value.
  - Two standardized sets of codes for symbols:
    - **ASCII**: American Standard Code for Information Interchange.
    - **EBCDIC**: Extended Binary Coded Decimal Interchange Code.
      - Not really used anymore!
    - **UNICODE**: Will discuss later!

Representation of Symbols and Text

A partial listing of the ASCII character set

- Ctrl+@ - 0000000
- Ctrl+A - 0000001
- Ctrl+B - 0000010
- Ctrl+C - 0000011
- Ctrl+D - 0000100
- Ctrl+E - 0000101
- Ctrl+F (Bell) - 0000110
- Space - 0100000
- Delete - 1111111
- 0 - 0110000
- 1 - 0110001
- 2 - 0110010
- 3 - 0110011
- 4 - 0110100
- 5 - 0110101
- 6 - 0110110
- 7 - 0110111
- 8 - 0111000
- 9 - 0111001
- A - 1000001
- B - 1000010
- C - 1000011
- D - 1000100
- E - 1000101
- F - 1000110
- G - 1000111
- H - 1001000
- I - 1001001
- J - 1001010
- K - 1001011
- a - 1000001
- b - 1000010
- c - 1000011
- d - 1000100
- e - 1000101
- f - 1000110
- g - 1000111
- h - 1001000
- i - 1001001
- j - 1001010
- k - 1001011

Representation of Images

- **Pictures**
  - A picture must be transformed into numeric form before it can be stored or manipulated by the computer.
  - Each picture is subdivided into a grid of squares called **pixels** (picture elements).
  - If the squares are small enough, we will see a reasonably good image.
Chapter 2

Representation of Images

- In a picture with only black and white pixels:
  - 1 represents black.
  - 0 represents white.

Photographic quality images have a **gray-scale**.
- Several shades between black and white are used.
- 4 level gray-scale means 4 shades are used.
  - Each pixel needs 2 bits.
  - 00 - represents white
  - 01 - represents light gray
  - 10 - represents dark gray
  - 11 - represents black
- 256 level gray scale means 8 bits per pixel are needed for 256 shades of gray

Three approaches to display color:

- **CMYK**:
  - Use of four standard colors: cyan, magenta, yellow, and black, in the printing industry.
- **RGB**:
  - Uses three values per pixel
  - One number is used for each of the amounts of Red, Green and Blue on the computer screen.
- A "Modified" **RGB**:
  - One number given per pixel to represent all three colors.

The baby's picture with smaller pixels - more detail.

The baby's picture with 4 levels of gray.

256 levels of gray

Full color image
Chapter 2

Representation of Images

- Digitizer or Scanner
  - A device that is used to convert an image to numbers representing a pixel form of the image.

Puzzler – Arecibo Message

- Designed by Frank Drake, sent into space 1974
- 1679 bits, divisible only by 23 x 73
- Counting in binary from 10 to 1 below, figure out scheme?

Representation of Sounds, Music and Speech

- Sounds, Music and Speech:
  - Each sound must be transformed into numeric form before it can be stored or manipulated by the computer.

What can be given numerical values in a piece of music?
- What notes are being played?
  - What is the frequency of each note?
    - Hertz is a unit of measurement that indicates the number of cycles per second of a particular sound’s vibration.
    - As an example, the sound of middle C is 256 Hertz.
- The tempo of the music (beats per minute).
- Lengths of the notes (half note, whole note, quarter note...).
- Could store the actual notes, instrument profile, all converted to binary (used in MIDI).
Representation of Sounds, Music and Speech

- **Representation of any Sound by Digital Recording:**
  - The sounds were divided into tiny segments and stored as binary numbers.
  - The computer transforms these binary numbers and reproduces the voltages.
  - These voltages are sent down the speaker wires to produce sound.

- **Constructing natural sounding words and phrases:**
  - Phonemes of a particular language are chosen.
  - Binary numbers are assigned to each phoneme.
  - Three additional factors have an affect on how a word or phrase sounds:
    - **Inflection:** Involves the rising or falling pattern of pitch on an individual phoneme.
    - **Duration:** Sound factor affecting the way a particular word sounds.
    - **Elision:** The connection of two or more phonemes sliced together so that when one ends, the next begins.

Representing the Instructions of Programs

- Instructions are imperative: they command action.
  - Each instruction must be clearly understood by its intended receiver.
  - The information needed to process the instruction must be readily available.
    - Automobile’s fasten-seat-belt command.
    - Highway patrol officer’s pull-over command.
    - Cooking recipe’s mix-ingredients-thoroughly instruction.

- A computer’s instructions must be stored in binary form within the computer before they can be used.
  - **Program:** A collection or list of commands designed for a computer to follow, which gives some desired result.
Example: Word Hunt

**Purpose:** To understand how a series of instructions can be stored in the computer as a group of binary numbers.

**Instruction set:** The pre-determined list of commands that comprise all of the possible instructions needed to perform a particular task.

**Syntax** (format) of all Word Hunt instructions:

<table>
<thead>
<tr>
<th>ACTION</th>
<th>OBJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION:</td>
<td>OBJECT:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTION:</td>
<td>OBJECT:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTION:</td>
<td>OBJECT:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### The Word Hunt Instruction Set:

1. **GOTO**  # Turn to page number designated.
2. **SELECT**  # Counting down from line 1, count down to the designated line number (line 1 is 1, blank lines don’t count).
3. **FORWARD**  # Count the number of words designated to the right (the first word is word 1).
4. **BACKUP**  # Beginning with the word immediately to the left of your current position, count backwards the number designated.
5. **WRITE**  word  Write a copy of the word on a piece of paper.
6. **STOP**  The message is complete.

### A Word Hunt program puzzle:

```
GOTO 7
SELECT 3
FORWARD 15
WRITE word
GOTO 5
FORWARD 12
WRITE word
GOTO 14
SELECT 30
FORWARD 4
WRITE word
BACKUP 3
WRITE word
STOP
```

### The Word Hunt Program in decimal

<table>
<thead>
<tr>
<th>Word Hunt Instruction Set:</th>
<th>Word Hunt Program</th>
<th>Program in decimal</th>
<th>Program in binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOTO 7</td>
<td>1</td>
<td>001</td>
<td>000011</td>
</tr>
<tr>
<td>SELECT 3</td>
<td>2</td>
<td>010</td>
<td>000011</td>
</tr>
<tr>
<td>FORWARD 15</td>
<td>3</td>
<td>011</td>
<td>001111</td>
</tr>
<tr>
<td>WRITE word</td>
<td>4</td>
<td>101</td>
<td>000101</td>
</tr>
<tr>
<td>GOTO 5</td>
<td>5</td>
<td>001</td>
<td>001111</td>
</tr>
<tr>
<td>FORWARD 12</td>
<td>6</td>
<td>101</td>
<td>0010101</td>
</tr>
<tr>
<td>WRITE word</td>
<td>7</td>
<td>011</td>
<td>0010010</td>
</tr>
<tr>
<td>GOTO 14</td>
<td>8</td>
<td>101</td>
<td>00000110</td>
</tr>
<tr>
<td>SELECT 30</td>
<td>9</td>
<td>011</td>
<td>011110</td>
</tr>
<tr>
<td>FORWARD 4</td>
<td>10</td>
<td>101</td>
<td>01101110</td>
</tr>
<tr>
<td>WRITE word</td>
<td>11</td>
<td>011</td>
<td>000100100</td>
</tr>
<tr>
<td>STOP</td>
<td>12</td>
<td>011</td>
<td>000000011</td>
</tr>
</tbody>
</table>

### The Word Hunt Program in binary

<table>
<thead>
<tr>
<th>Word Hunt Instruction Set:</th>
<th>Word Hunt Program</th>
<th>Program in decimal</th>
<th>Program in binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOTO 7</td>
<td>1</td>
<td>001</td>
<td>000011</td>
</tr>
<tr>
<td>SELECT 3</td>
<td>2</td>
<td>010</td>
<td>000011</td>
</tr>
<tr>
<td>FORWARD 15</td>
<td>3</td>
<td>011</td>
<td>001111</td>
</tr>
<tr>
<td>WRITE word</td>
<td>4</td>
<td>101</td>
<td>000101</td>
</tr>
<tr>
<td>GOTO 5</td>
<td>5</td>
<td>001</td>
<td>001111</td>
</tr>
<tr>
<td>FORWARD 12</td>
<td>6</td>
<td>101</td>
<td>0010101</td>
</tr>
<tr>
<td>WRITE word</td>
<td>7</td>
<td>011</td>
<td>0010010</td>
</tr>
<tr>
<td>GOTO 14</td>
<td>8</td>
<td>101</td>
<td>00000110</td>
</tr>
<tr>
<td>SELECT 30</td>
<td>9</td>
<td>011</td>
<td>011110</td>
</tr>
<tr>
<td>FORWARD 4</td>
<td>10</td>
<td>101</td>
<td>01101110</td>
</tr>
<tr>
<td>WRITE word</td>
<td>11</td>
<td>011</td>
<td>000100100</td>
</tr>
<tr>
<td>STOP</td>
<td>12</td>
<td>011</td>
<td>000000011</td>
</tr>
</tbody>
</table>
Two main differences between our Word Hunt program and a computer program:

- The computer’s program would have originally been written in a programming language, then be translated into binary code for the computer.
- Each instruction in the instruction set would have to be something that the computer was capable of doing:
  - ADD numbers
  - MOVE data from one place to another
  - Change the bits of a location in memory
  - ...

Even though instruction sets differ, they all contain these classes of instructions:

- Arithmetic Instructions
- Data Movement Instructions
- Logical or Comparison Instructions
- Control Instructions
- Input/Output Instructions

All instructions must have:

- **Opcode** (Operation Code): The part of the instructions that tells the computer what to do.
- **Operand**: The "object" of the operation to be performed.
  - Example: If the operation is to add a number, then the operand will tell where to find the number that is to be added.

How can the computer tell what this string of binary numbers is used for? 01011010

- An instruction?
- A number?
- A sound’s frequency?
- The value of a pixel in a gray-scale image?
- An ASCII character?

It is the program that is active that determines the interpretation of the string of binary numbers!
Software Application: Word Processing

- Word Processing Programs
  - We’ll skip this part, but please skim the section and read it more carefully if you are not familiar with word processing.