Chapter 1

Introduction

Computers perform any job that their programs tell them to do.
- A program is a set of instructions that a computer follows to perform a task.
- Programs are commonly referred to as software.

Hardware

- The term *hardware* refers to any physical device that a computer is made of.
- Computers are, in fact, large systems that consist of different devices that work together.

Hardware Categories

- Input Devices
- Process Devices
- Output Devices
- Store Devices

Input Devices

- Convert analog to digital data
- *Input* - feed information into a computer
  - words and symbols
  - numbers
  - pictures
- Common forms of input
  - keyboard
  - mouse
Process Devices

- Computers can analyze and create data
- Basic Terms:
  - process - actions used to manipulate data
  - processor - device that processes data
- Common forms
  - Pentium Processor
  - various other chips and hardware

Output Devices

- Convert digital data to analog
- Output – send information out of a computer
- Common forms of output
  - Monitor
  - Printers
  - Music and Sound – using speakers

Storage Devices

- Copy data for later use
- The data is kept in digital form
- Common Forms:
  - Memory
  - CD ROM
  - Hard Disk
  - Solid State Drives (SSD) – i.e. your thumb drives

Storage Devices

- **Primary** Storage
  - helps run your computer
  - this includes motherboard memory and memory used to run programs
- **Secondary** Storage
  - much slower than primary storage
  - allows data to be stored permanently

Functions of a Computer

- Input Data -> Process Data -> Output Data
- Store Data

Functions of a Computer

- Analog <-> Digital
The Processor

Chapter 1.2

The Central Processing Unit (CPU) is the most complex part of a computer.

In fact, it is the computer.

It works far differently from high-level programming languages.

Over time, thousands of processors have been developed.

Examples:
- Intel x86
- IBM PowerPC
- MOS 6502
- ARM

Modern processors are small enough to fit in your hand.

And they contain millions of transistors.

The size of computers has changed drastically in the last 70 years.

The first computer was huge by today's standards.

Electronic Numerical Integrator And Computer
Operational February 1946

Features
- 5 KHz (5000 Hz)
- programmed by rewiring – pre 1948
- based on decimal – not binary
- weighed 30 tons, 18 feet high, 80 feet long

Designed to be Turing Complete

Development
- John Eckert and John W. Mauchly
- U.S. Ballistics Research Laboratory
- Needed to fight World War II – then Cold War
- Compute ballistic firing tables
ENIAC

- A tube burned out once every 2 days
- Retired in 1955
- operational for only 9 years
- But... in just 9 years, it is estimated to have done more calculations than all of humanity, combined, had ever done before

Primary Storage

Chapter 1.2

Random Access Memory

- Random Access Memory (RAM)
  - memory used to run data and programs
  - fast
  - temporary – it is gone after the computer is turned off (power is lost)
- The more memory you have...
  - the more you can open/run at one time
  - stored on DIMM cards that can be added to motherboards

Virtual Memory

- Used when the system runs out of memory
  - computers with limited RAM can run large programs
  - this is a type of "emergency" memory
- Uses hard disk space
  - slow... not as fast as RAM
  - "invisible" to application software
Secondary Storage

Chapter 1.2

Hard Disks

- Use magnets to store data
- Hard disk **platter**
  - flat, rigid, maintainable disk used to store bits
  - there are multiple platters in each hard drive
- Head crash
  - the read-write head **hits** into a dust particle or other contaminant on the disk
  - head crash damages some data on disk

Inside a Hard Disk

Solid State Storage

- Data is stored on a low-power chip
- Advantages
  - non-volatile (stored when power is lost)
  - portable
  - versatile – used from digital cameras to computers

Solid State Technology

- Cards
  - Compact Flash
  - MMC
  - Secure Digital
  - Smart Media
- USB Flash Drive
  - plugs into any USB port
  - acts like a hard drive

Optical Storage

- Data is read using lasers
  - light spots are called **lands**
  - dark spots are called **pits**
- Safer than magnetic media
  - data not lost over time
  - safe from magnets
  - resists the other elements
Current Mediums

- **CD**
  - Compact Disk
  - Holds 700 MB of data (80 min)

- **DVD**
  - Digital Versatile Disk
  - Holds about 4.7 GB of data
  - Double layer can store 8.5 GB

- **Blue-ray**
  - Named after blue laser used to read/write the data
  - Official acronym is BD
  - Holds about 25 GB of data
  - Double layer can store 50 GB

Binary Numbers

Chapter 1.3

- **Hindu-Arabic Number System**
  - Positional grouping system
  - Each position represents an increasing power of 10
  - Used throughout the World

- **Binary numbers**
  - Based on the same system
  - Use powers of 2 rather than 10

What is a Number?

- **Hindu-Arabic Number System**
  - Positional grouping system
  - Each position represents an increasing power of 10
  - Used throughout the World

- **Binary numbers**
  - Based on the same system
  - Use powers of 2 rather than 10

Evolution of a Genius System

Base 10 Number

The number 1783 is ...

<table>
<thead>
<tr>
<th>10⁴</th>
<th>10³</th>
<th>10²</th>
<th>10¹</th>
<th>10⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1000</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>

1000 + 700 + 80 + 3 = 1783
Binary Number Example

The number 0100 1010 is ...

<table>
<thead>
<tr>
<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>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

64 + 8 + 2 = 74

The number 1101 1011 is ...

<table>
<thead>
<tr>
<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>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

128 + 64 + 16 + 8 + 2 + 1 = 219

Bits and Bytes

- Everything in a modern computer is stored using a combination of ones and zeros
- *Bit* is one binary digit
  - either 1 or 0
  - shorthand for a bit is b
- *Byte* is a group of 8 bits
  - e.g. 0010 0100
  - shorthand for a byte is B

Adding Binary Integers

- Computer's add binary numbers the same way that we do with decimal
- Columns are aligned, added, and "1's" are carried to the next column
- On processors, this is called an *adder*
### Adding Base 10 Numbers

\[
\begin{array}{c}
1 & 1 \\
\hline
2 & 7 & 8 & 1 \\
3 & 7 & 2 & 1 \\
\hline
6 & 5 & 0 & 2
\end{array}
\]

### Adding Binary Example

\[
\begin{array}{c}
1 & 1 & 1 & 1 \\
\hline
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
& 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1
\end{array}
\]

### How Text Is Stored

Chapter 1.3

- Computer often store and transmit textual data
- Examples:
  - punctuation
  - numerals 0 – 9
  - letter
- Each of these symbols is called a **character** and are the basis for written communication

### Characters

- Processors rarely know what a "character" is, and instead store each as an integer
- In this case, each character is given a unique value
- For instance
  - "A" could have the value of 1
  - "B" is 2
  - "C" is 3, etc...

### Characters

- Characters and their matching values are a **character set**
- There have been many characters sets developed over time
Character Sets

- **ASCII**
  - 7 bits – 128 characters
  - uses a full byte, one bit is not used
  - created in the 1967
- **EBCDIC**
  - Alternative system used by old IBM systems
  - Not used much anymore

ASCII Chart

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 A B C D E F</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL SOH STX ETX EOT ENQ ACK BEL BS HT LF VT FF CR SO SI</td>
</tr>
<tr>
<td>DLE DC1 DC2 DC3 DC4 NAK SYN ETB CAN EM SUB ESC FS GS RS US</td>
</tr>
</tbody>
</table>

ASCII Codes

- Each character has a unique value
- The following is how "OMG" is stored in ASCII

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hex</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>0100 1111</td>
<td>79</td>
</tr>
<tr>
<td>M</td>
<td>0100 1101</td>
<td>77</td>
</tr>
<tr>
<td>G</td>
<td>0100 0111</td>
<td>71</td>
</tr>
</tbody>
</table>

Only Control Characters Still Used

Unicode Character Set

- ASCII is only good for the United States
  - Other languages need additional characters
  - Multiple competing character sets were created
- Unicode was created to support every spoken language
  - Developed in Mountain View, California

Unicode Character Set

- Originally used 16 bits
  - that's over 65,000 characters!
  - includes every character used in the World
- Expanded to 21 bits
  - 2 million characters!
  - now supports every character ever created
- Unicode can be stored in different formats
How a Program Works

Chapter 1.4

- The CPU does all the operations on the computer
- Each operation is called an instruction
- The collection of the instructions that can be performed on a computer is its instruction set

- Examples:
  - read a piece of data
  - add two numbers
  - multiply two numbers
  - moving data around
  - comparing two pieces of data
  - etc...

How a Program Works

- Computer programs, just like everything else, are just 1's and 0's
- So, the processor only reads, and "understands", binary
- These binary instructions are called machine language

High-Level Programming

- Most programs are written in languages such as C#, Java, Visual Basic, etc...
- These are high-level languages which are written in simple readable text

High-Level Programming

- Examples: Java, C++, Python, Lua, etc...
- Programs written in high-level languages are referred to as source code
Programming Languages

- **Language**
  - series of symbols & words that form a meaningful pattern
  - This is true of spoken languages such as English, Spanish, Hindi, Arabic, etc...

- **Programming**
  - language used to write programs
  - there are many different programming languages

Example Programming Languages

- Ada
- BASIC
- FORTRAN
- COBOL
- C
- C++
- C#
- Java
- JavaScript
- Lua
- Perl
- Python
- Ruby
- Smalltalk
- Swift
- Visual Basic .NET

IDEs

- Many high-level languages are written in an Integrated Development Environment (aka Integrated Design Environment)
- It is a program designed to help the programmer create the program
- Contains features for debugging, managing, compiling, etc...

From Abstract to Machine

- High-Level Language 3rd Generation
- Assembly 2nd Generation
- Machine Code 1st Generation

Compilers and Interpreters

- **Compilers**
  - convert a high-level language directly to assembly or machine code

- **Interpreters**
  - looks at a high-level language and executes it immediately – using its own code
  - similar to the concept of macros – which might have heard about

Compilers: 3rd → 2nd Generation

- Compiler
  - moves $r9$, 1846
  - add $r0$, 42
  - mov $r9$, 3
  - mov $[r9+r0]$, $r8$
Assembler

- Converts assembly into machine code
- Each computer instruction is written using a mnemonic – a short name for the instruction
- Programmers have the full power of the processor – but have to write programs carefully

Assembler: 2nd → 1st Generation

<table>
<thead>
<tr>
<th>Assembler</th>
<th>Machine Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>mov r8, 1846</td>
<td>01000100</td>
</tr>
<tr>
<td>add r8, 42</td>
<td>01100101</td>
</tr>
<tr>
<td>mov r9, 3</td>
<td>01110110</td>
</tr>
<tr>
<td>mov [a + r9*8], r8</td>
<td>01101001</td>
</tr>
<tr>
<td></td>
<td>01101110</td>
</tr>
</tbody>
</table>

Types of Software

Chapter 1.5

Software Major Categories

- **System Software**
  - runs programs & manages data
  - operating System – Windows, Mac-OS
  - includes utility programs
- **Application Software**
  - works with the user to perform a task
  - example: Microsoft Word, Solitaire

What an Operating System Does

- Master controller for all of the activities that take place within a computer
- **Basic Duties:**
  - memory management
  - track resources
  - communicate with devices
  - interact with application software
  - interact with the user

Microsoft Windows

- The most common operating system on Intel-PCs
- **Major Versions:**
  - Windows 95 – 1995
  - Windows XP – 2001
  - Windows 7 – 2010
  - Windows 8 – 2012
  - Windows 10 – 2015
Windows

Evolved from UNIX
Multiple competing versions
- Red Hat
- Ubuntu
- Android phones
- etc....
Popular for small servers &
computer science workstations

Linux

- Created by the Apple
  Corporation for the Macintosh
- Major Versions:
  - System 1 – 1984
  - System 6 – 1988
  - System 7 – 1991
  - Mac-OS X – 2001