Week 1 – Bits & Bytes

Student number: 588406

Assignment 1.1: Bits & Bytes intro

What are Bits & Bytes?

A bit is the smallest form of data in a computer, which has only two values - 0 or 1. A byte always has 8 bits and is normally used to store letters, numbers, and other symbols. Everything that is digital in nature in a computer is made up of bits and bytes. File sizes in units of KB, MB, and GB show storage in bytes.

What is a nibble?

A nibble is regarded as half of a byte and consists of 4 bits, thus because a nibble is comprised of 4 bits, there are 16 possible values for the nibble (2^4 = 16), and this is just what is required to describe one hexadecimal character, the digits 0 through 9 and A through F. The application of nibbles in the encoding of hexadecimal characters and programming is demonstrated below.

What relationship does a nibble have with a hexadecimal value?

A nibble is made up of 4 bits. One hexadecimal digit also stands for a total of 4 bits. It is for this reason that a nibble directly translates to a hex digit. This aspect of the hex system is highly useful for the representation of binary.

Why is it wise to display binary data as hexadecimal values?

Binary numbers are very lengthy, making it difficult for a human to read. Hexadecimal numbers are more readable while still carrying a close relationship with binary numbers. In hex numbers, 4 bits are represented in one digit, which makes it easy for binary data to be converted into hex without losing data.

What kind of relationship does a byte have with a hexadecimal value?

A byte consists of 8 bits, and each hexadecimal digit represents 4 bits. This means that one byte is represented by two hexadecimal digits. For example, the byte 11010110 in binary is written as D6 in hexadecimal.

An IPv4 subnet is 32-bit, show with a calculation why this is the case.

An IPv4 address consists of 4 bytes, with each bytes containing 8 bits.
Calculation: 4 octets × 8 bits per octet = 32 bits
Example: A typical IPv4 address like 192.168.1.1 breaks down as:

- 192 = 8 bits (first octet)
- 168 = 8 bits (second octet)
- 1 = 8 bits (third octet)
- 1 = 8 bits (fourth octet)

Total: 8 + 8 + 8 + 8 = 32 bits
This 32-bit structure allows for 2^{32} = 4,294,967,296 possible unique IP addresses in the IPv4 system.

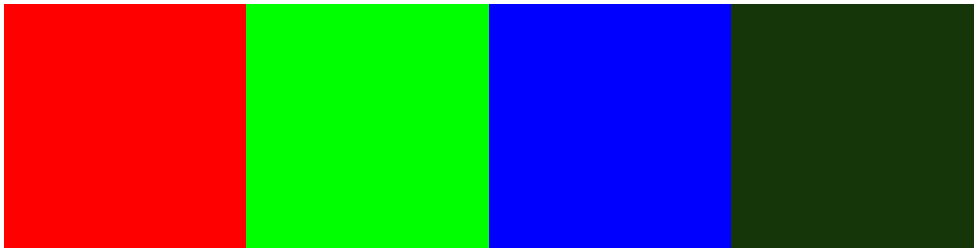
Assignment 1.2: Your favourite color

Hexadecimal color code: #093615

Assignment 1.3: Manipulating binary data

Color	Color code hexadecimaal (RGB)	Big Endian	Little Endian
RED	#FF0000	FF 00 00	00 00 FF
GREEN	#00FF00	00 FF 00	00 FF 00
BLUE	#0000FF	00 00 FF	FF 00 00
WHITE	#FFFFFF	FF FF FF	FF FF FF
Favourite (previous assignment)	#093615	09 36 15	15 36 09

Screenshot modified BMP file in hex editor:



Assignment 1.4: Student number to HEX and Binary

Convert your student number to a hexadecimal number and a binary number.

Explain in detail that the calculation is correct. Use the PowerPoint slides of week 1.

Hexadecimal of my student number 588406:
588406 ÷ 16 = 36775 remainder 6
36775 ÷ 16 = 2298 remainder 7
2298 ÷ 16 = 143 remainder 10 (A)
143 ÷ 16 = 8 remainder 15
(F) 8 ÷ 16 = 0 remainder 8
Result: 8FA76

Binary of my student number 588406:
588406 ÷ 2 = 294203 remainder 0
294203 ÷ 2 = 147101 remainder 1
147101 ÷ 2 = 73550 remainder 1
73550 ÷ 2 = 36775 remainder 0
36775 ÷ 2 = 18387 remainder 1
18387 ÷ 2 = 9193 remainder 1

9193 ÷ 2 = 4596 remainder 1
4596 ÷ 2 = 2298 remainder 0
2298 ÷ 2 = 1149 remainder 0
1149 ÷ 2 = 574 remainder 1
574 ÷ 2 = 287 remainder 0
287 ÷ 2 = 143 remainder 1
143 ÷ 2 = 71 remainder 1
71 ÷ 2 = 35 remainder 1
35 ÷ 2 = 17 remainder 1
17 ÷ 2 = 8 remainder 1
8 ÷ 2 = 4 remainder 0
4 ÷ 2 = 2 remainder 0
2 ÷ 2 = 1 remainder 0
1 ÷ 2 = 0 remainder 1
Result: 10001111101001110110

Ready? Save this file and export it as a pdf file with the name: **week1.pdf**