Template Week 1 – Bits & Bytes

Student number:

Assignment 1.1: Bits & Bytes intro

What are Bits & Bytes?

Bit: Short for "binary digit," a bit is the smallest unit of data in a computer and can have a value of either 0 or 1. Think of it as a tiny switch that can be either off (0) or on (1).

Byte: A byte is a collection of 8 bits and can represent 256 different values (from 0 to 255). Bytes are used to encode a wide variety of data, such as letters, numbers, and other characters in a computer. For example, one byte can represent a single character in the ASCII encoding, which includes letters, numbers, and punctuation.

What is a nibble?

A nibble (or nybble) is a unit of digital information that consists of four bits. Since a byte is made up of eight bits, a nibble is exactly half of a byte. A nibble can represent 16 different values (from 0 to 15).

What relationship does a nibble have with a hexadecimal value?

- A hexadecimal (or hex) system is base-16, meaning it uses 16 different symbols: 0-9 and A-F (where A stands for 10, B for 11, and so on up to F, which stands for 15).
- Each of these hex symbols can be represented by a 4-bit binary number. For example, the binary number 1010 is A in hexadecimal, and 1111 is F.

So, a nibble (which is 4 bits) maps perfectly to a single hex digit. This makes the hex system a very convenient way to represent binary data, as each hex digit represents exactly half a byte.

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

Displaying binary data as hexadecimal values is smart because:

1. **Compact**: Hex is shorter and more manageable than binary.

2. **Easy Conversion**: Each hex digit represents 4 binary bits.

3. **Readable**: Hex is easier for humans to read than long binary strings.

4. **Standard**: Widely used in programming and computing.

In a nutshell, hex simplifies and streamlines the representation of binary data.

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

A byte, which is 8 bits, has a straightforward relationship with hexadecimal values because:

- **8 bits (1 byte)**: Can be represented by **2 hexadecimal digits**.

Each 4-bit nibble in the byte corresponds to a single hex digit. This means a byte can be split into two nibbles, and each nibble can be easily converted to its hex equivalent.

So, each byte maps directly to two hex digits, making it easy to represent and work with binary data.

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

1. **IPv4 Address Breakdown**:
    - An IPv4 address is written in dotted decimal notation, like 192.168.1.1.
    - Each segment of this address represents 8 bits (1 byte).
2. **Binary Representation**:
    - Let's take the example address 192.168.1.1 and convert each segment to its binary form:
        - 192 in binary: 11000000
        - 168 in binary: 10101000
        - 1 in binary: 00000001
        - 1 in binary: 00000001

So, 192.168.1.1 in binary is: 11000000.10101000.00000001.00000001

3. **Total Bits**:
    - Each of the four octets is 8 bits long.
    - Thus, the total number of bits is 8 bits×4 octets=32 bits8 \, \text{bits} \times 4 \, \text{octets} = 32 \, \text{bits}.
4. **Subnet Mask**:
    - An IPv4 subnet mask also uses 32 bits and is used to define which portion of the IP address refers to the network and which part refers to the host. For example, a subnet mask of 255.255.255.0 in binary is:
        - 255 in binary: 11111111
        - 255 in binary: 11111111
        - 255 in binary: 11111111
        - 0 in binary: 00000000

- Combined binary subnet mask: 11111111.11111111.11111111.00000000

So, the calculation here shows that both IPv4 addresses and their corresponding subnet masks are made up of 32 bits each. This 32-bit structure allows for a maximum of 2322^{32} unique IPv4 addresses, although in practice, not all of these addresses are usable due to network constraints.

Assignment 1.2: Your favourite colour

Hexadecimal colour code:

#66074e

Assignment 1.3: Manipulating binary data

Colour	Colour 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)	#66074e	66 07 4E	4E 07 66

Screenshot modified BMP file in hex editor:

Bonus point assignment – week 1

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.

566040

DEC to Binary

566040 divided by 2 = 283020 1

283020 / 2 = 141510 0

141510 / 2 = 70755 0

70755 / 2 = 35377 0

35377 / 2 = 17688 1

17688 / 2 = 8844 0

8844 / 2 = 4422 1

4422 / 2 = 2211 0

2211 / 2 = 1105 0

1105 / 2 = 552 0

552 / 2 = 276 1

276 / 2 = 138 1

138 / 2 = 69 0

69 / 2 = 34 0

34 / 2 = 17 0

17 / 2 = 8 1

8 / 2 = 4 0

4 / 2 = 2 0

2 / 2 = 1 0

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