

How Numbers are Expressed

Numbers, in the software of the computer at machine level, are expressed on hexadecimal format. Despite the chips being made of transistors and only being able to understand binary numbers, computers work in hexadecimal most of the time.

They are more human readable numbers that can be easily interpreted in bytes, the way in which bits in memory are grouped. For example, looking at the a CPU architecture, most of the buses, networks, registers and caches; are expressed in terms of bytes.

When using popular reverse engineering tools or visualizing files in C, most usually a hexadecimal number is displayed starting with `0x` and a binary number with `0b`. A couple of examples:

```
0b11111111 = 0xff
0b00001111 = 0x0f
0b11110010 = 0xf2
0b00000010 = 0x02
0b01010101 = 0x55
```

The 1 byte numbers that in binary are 8 characters long, expressed in hexadecimal are just 2. Each string of 4 binary numbers are just 1 hexadecimal characters.

This approach could not be taken using decimal (base 10) numbers, since the base is not a power unit of 2. With powers of 2, it is very easy to translate into binary numbers and vise-versa.

[Video Reference](#)