Jun 1, 2014

C Programming #01: Bit, Byte etc

Welcome to series of articles on C Programming Language. I hope to cover the entire C Programming from ground up. I don't want to just limit it to simple topics but also cover advanced topics. You can find the entire list of articles here.

C is a programming language. C is used for low level programming like writing device driver, operating system core component. Before we go into details of language constructs and its details, lets cover briefly terms which would be used often like Bit, Byte, Word and ASCII.

Bit

Bit is short for **BI**nary digi**T**. Computer understand only in 0 and 1 hence binary. Bit can have value either 0 or 1. Inside a computer, they are represented by range of voltage levels.

- 0 will be in range 0 – 0.9 V
- 1 will be in range 1.7 – 3.3 V

Above numbers are just examples and can vary of course.

Byte

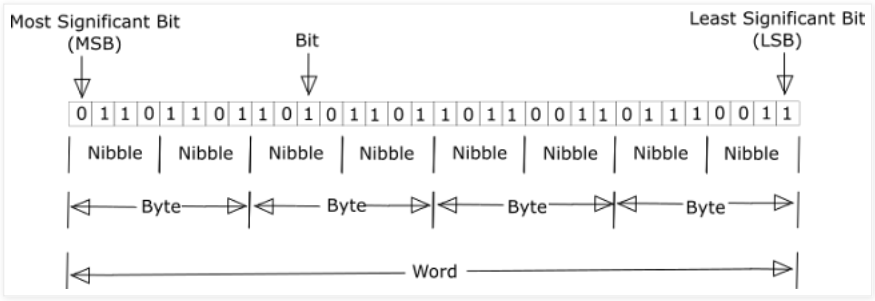
8 bit is collectively called as byte. 01011101 is byte, which has 8 bits in them. first bit (0 in above case) is called as *MSB – Most Significant bit*. last bit (1 in above case) is called as *LSB – Least Significant bit*. Byte has its origin from English word "*bite*", it is purposefully misspelled to give new meaning which is *bite of memory*.

Nibble

4 bit is collectively called as nibble. Byte consists of 2 nibble, which are referred to as upper and lower nibble. Say for example 01011101 is the byte then 0101 is upper nibble, 1101 is lower nibble. Actual meaning of Nibble is *small bite*.

Word

Word is the minimum number of bits that is processed by the Central Processing Unit (CPU) at a time. It is generally size of the accumulator register in CPU. Unlike bit, byte and nibble; word has no fixed length of bits. If you are using a 32 bit computer then word length is 32 bit (4 byte), if you are using 64 bit computer then word length is 64 bit (8 byte).



Hex

Hex is short for Hexa decimal number. Binary is base 2 (0,1 symbols), decimal is base 10 (0,1,2,3,4,5,6,7,8,9 symbols) and Hex is base 16 (0,1,2,3,4,5,6,7,8,9,A,B,C,E,F symbols). Hex helps in shorter representation of the binary numbers. Let me explain it by taking a below example.

Lets take all combination of nibble and their corresponding Hex and Decimal representation.

Binary	Hex	Decimal
0000	0	0
0001	1	1

0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Hence binary number 10100101 is represented in Hex as A5H. Hex is alternate representation/short hand of binary. Most of binary numbers are represented in Hex always. Good practice will allow in mind conversion between these two representation.

ASCII

ASCII is abbreviation of **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. It is all fine with binary, hex and so... but we need to store alphabets, symbols and numbers in computer. ASCII is standard mapping that is provided between binary and symbols that needs representation. ASCII consists both printable and non-printable characters in them. http://www.asciitable.com/consists of complete list.
Some interesting ASCII values are

Dec	Hex	Char	Comments
0	0	NULL	NULL means it has value 0
7	7	BELL	it causes bell to be rung
48	30	0	Number 0
49	31	1	Number 1
57	39	9	Number 9
65	41	A	Capital A
66	42	B	Capital B
90	5A	Z	Capital Z
97	61	a	Small a
122	7A	z	Small z
32	20		Space

Generally when char 'A' is stored in memory of computer, computer will store 0100 0001 in its memory. Hence 0100 0001 binary, 41 hex, 65 decimal is ASCII value of 'A'.

Links

Quiz - C Quiz #01: **Bit, Byte etc**
Next Article - C Programming #02: **A Brief history of C**
Previous Article - None
All Article - **C Programming**

Posted by Harish

No comments :

Post a Comment

Even though increment and decrement operator is type of Arithmetic operator. Bu think it needs special attention hence this exclusive art...

Vedic Maths #01: Complement method - Subtraction from 100/1000/10000
This is the easiest of all the methods. Subtracting a number from 100 or 1000 or 10000 etc.. Regular method involves taking lot of borrow....

SICP# E1.1: Evaluate scheme expression
SICP# E1.1: Evalute scheme expression Below is a sequence of expressions. What the result printed by the interpretter in ...

To leave a comment, click the button below to sign in with Google.

SIGN IN WITH GOOGLE

Newer Post Home

Subscribe to: Post Comments (Atom)