Little and Big Endian

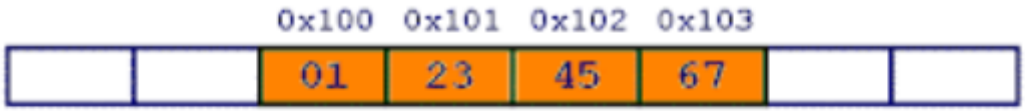
Endianness

Little and big endian are two ways of storing multibyte data-types into memory. For single byte data-types like a char, it doesn't matter what the endian because it is only one byte. Duh.

Let's assume that an integer is 4 bytes in the 32 bit system that we are working with in C.

Let's have the hexadecimal value **0x01234567** this is 32 bit remember every 2 hexadecimal digit is one byte, there is 4 pair of the hexadecimal digits hence 32 bits.

Now we will be storing it into memory and these are the two ways.



Big Endian

Little Endian

Big Endian

So we have our hexadecimal **0x01234567** when we are just writing the numbers the most significant bit is always on the left hand side, i.e. 01 in hex or 0000 0001 is the most significant byte.

If the machine is big endian then it will store each byte into memory as the way it is written:

- 01 into memory address 0x100
- 23 into memory address 0x101
- 45 into memory address 0x102
- 67 into memory address 0x103

In this example, the memory address increases.

Little Endian

With little endian we will be storing the least significant byte into the address and going backward:

- 67 into memory address 0x100
- 45 into memory address 0x101
- 23 into memory address 0x102
- 01 into memory address 0x103

Example program to show endian

```
void show_mem_rep(char *start, int n)
{
    int i;
    for (i = 0; i < n; i++)
         printf(" %.2x", start[i]);
    printf("\n");
}

/*Main function to call above function for 0x01234567*/
int main()
{
   int i = 0x01234567;
   show_mem_rep((char *)&i, sizeof(i));
   getchar();
   return 0;
}
```

It uses a char pointer to go through the integer and show the endian of the machine.

If it is big endian it will print out 01 23 45 67

If it is little endian it will print out 67 45 23 01

Do I need to care about endian?

Most of the time no, the compiler or the interpreter will usually take care of the endianness.

But when you work with network programming, i.e. sending data over a network socket, you will have to be cognisance of the endian by say converting the integer that you are sending over the wire into network bytes order (big endian), then converted back to host byte order (whether it is big endian or little endian).

Revision #2
Created 2023-05-05 17:16:21 UTC by Tamarine

Updated 2023-05-06 01:33:43 UTC by Tamarine