# Little and Big Endian Mystery

What are these?
Little and big endian are two ways of storing multibyte data-types (int, float, etc). In little endian machines, last byte of binary representation of the multibyte data-type is stored first. On the other hand, in big endian machines, first byte of binary representation of the multibyte data-type is stored first.

Suppose integer is stored as 4 bytes (For those who are using DOS based compilers such as C++ 3.0 , integer is 2 bytes) then a variable x with value 0x01234567 will be stored as following.
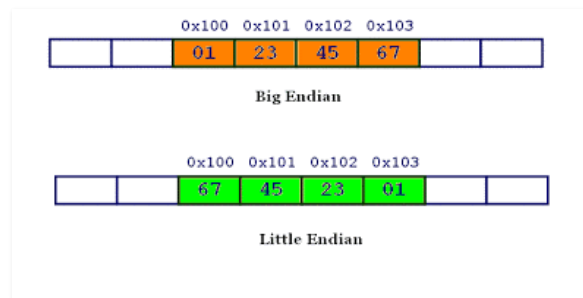
Memory representation of integer ox01234567 inside Big and little endian machines
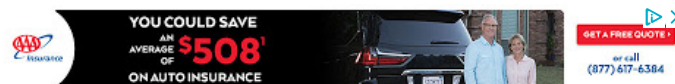
How to see memory representation of multibyte data types on your machine?
Here is a sample C code that shows the byte representation of int, float and pointer.

```
#include <stdio.h>

/* function to show bytes in memory, from location start to start+n*/
```

```
            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;
    }
```

When above program is run on little endian machine, gives "67 45 23 01" as output , while if it is run on big endian machine, gives "01 23 45 67" as output.

Is there a quick way to determine endianness of your machine?
There are n no. of ways for determining endianness of your machine. Here is one quick way of doing the same.

C++   **C**

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    unsigned int i = 1;
    char *c = (char*)&i;
    if (*c)
        cout<<"Little endian";
    else
        cout<<"Big endian";
    return 0;
}

// This code is contributed by rathbhupendra
```

Output:

```
Little endian
```

Most popular in Articles

PDFmyURL easily turns web pages and even entire websites into PDF!

PDFmyURL

In the above program, a character pointer c is pointing to an integer i. Since size of character is 1 byte when the character pointer is de-referenced it will contain only first byte of integer. If machine is little endian then *c will be 1 (because last byte is stored first) and if machine is big endian then *c will be 0.

Does endianness matter for programmers?
Most of the times compiler takes care of endianness, however, endianness becomes an issue in following cases.

It matters in network programming: Suppose you write integers to file on a little endian machine and you transfer this file to a big endian machine. Unless there is little endian to big endian transformation, big endian machine will read the file in reverse order. You can find such a practical example here.

Standard byte order for networks is big endian, also known as network byte order. Before transferring data on network, data is first converted to network byte order (big endian).

Sometimes it matters when you are using type casting, below program is an example.

```c
#include <stdio.h>
int main()
{
    unsigned char arr[2] = {0x01, 0x00};
    unsigned short int x = *(unsigned short int *) arr;
    printf("%d", x);
    getchar();
    return 0;
}
```

In the above program, a char array is typecasted to an unsigned short integer type. When I run above program on little endian machine, I get 1 as output, while if I run it on a big endian machine I get 256. To make programs endianness independent, above programming style should be avoided.

What are bi-endians?
Bi-endian processors can run in both modes little and big endian.

What are the examples of little, big endian and bi-endian machines ?
Intel based processors are little endians. ARM processors were little endians. Current generation ARM processors are bi-endian.

Motorola 68K processors are big endians. PowerPC (by Motorola) and SPARK (by Sun) processors were big endian. Current version of these processors are bi-endians.
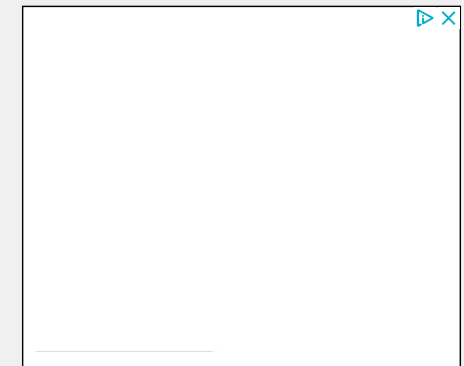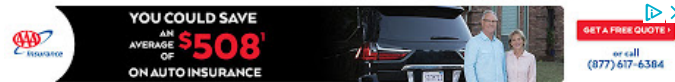
Does endianness affects file formats?

File formats which have 1 byte as a basic unit are independent of endianness e.g., ASCII files . Other file formats use some fixed endianness forrmat e.g, JPEG files are stored in big endian format.

Which one is better — little endian or big endian?

The term little and big endian came from Gulliver's Travels by Jonathan Swift. Two groups could not agree by which end an egg should be opened -a- the little or the big. Just like the egg issue, there is no technological reason to choose one byte ordering convention over the other, hence the arguments degenerate into bickering about sociopolitical issues. As long as one of the conventions is selected and adhered to consistently, the choice is arbitrary.

Recommended Posts:

Sum of Bitwise OR of all pairs in a given array

Maximum subset sum such that no two elements in set have same digit in them

Count pairs of elements such that number of set bits in their AND is B[i]

Number of 0s and 1s at prime positions in the given array

Data Structures and Algorithms Online Courses : Free and Paid

Find uncommon characters of the two strings | Set 2

Introduction to Model View View Model (MVVM)

Check if the two given stacks are same

Fibonacci Power

Find N distinct numbers whose bitwise Or is equal to K

Find a sub matrix with maximum XOR

Multiply a number by 15 without using * and / operators

Queries for bitwise OR in the given matrix

Find average of two numbers using bit operation

Improved By : **skbarnwal, louisportay**

PDFmyURL easily turns web pages and even entire websites into PDF!

PDFmyURL

Article Tags : Articles Bit Magic Big Endian Endianness Little Endian

Practice Tags : Bit Magic

👍

18

☐ To-do ☐ Done

2.4

Based on 57 vote(s)

Feedback/ Suggest Improvement   Add Notes   Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use **ide.geeksforgeeks.org**, generate link and share the link here.

Load Comments

GeeksforGeeks
A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY
About Us
Careers
Privacy Policy
Contact Us

LEARN
Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE
Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE
Write an Article
Write Interview Experience
Internships
Videos

f in ▶ 🐦 ▶

@geeksforgeeks, Some rights reserved