

第一次作业—系统整数分析

康锦辉 3180105099 2020.02.29

1、运行以下C语言程序：

```
1 #include <iostream>
2 #include <stdio.h>
3
4 int main(int argc, char** argv){
5     int x=-2147483648;
6     printf("%d, %d", x, -x);
7 }
```

分析以上程序的运行结果，为什么。

solution:

输出结果：

-2147483648, -2147483648

原因：

计算机用补码的形式存放数据，int为32位整型，

-2147483648的二进制形式为：1000 0000 0000 0000 0000 0000 0000 0000，

运算符“-”的处理方式为取反加一，

将1000 0000 0000 0000 0000 0000 0000 0000取反，

得0111 1111 1111 1111 1111 1111 1111 1111，再加1，

得1000 0000 0000 0000 0000 0000 0000 0000 = -2147483648

2、系统里以字节(byte)为寻址单位，写个C语言程序，说明一个4字节的int: 0x12345678在机器里是怎么存放的？

solution:

C语言程序：

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main(){
5      int x=0x12345678;
6      int *p=&x;
7      char *pp=(char*)p;
8      printf("%p:%x\n", pp, *pp);
9      printf("%p:%x\n", pp+1, *(pp+1));
10     printf("%p:%x\n", pp+2, *(pp+2));
11     printf("%p:%x\n", pp+3, *(pp+3));
12
13     return 0;
14 }
```

输出结果：

```
1  000000000061FE0C: 78
2  000000000061FE0D: 56
3  000000000061FE0E: 34
4  000000000061FE0F: 12
```

可以看到，计算机采用小端规则来存放数据，即低位字节在前，高位字节在后。