

定点数表示方法报告

18081811-高炜哲

程序 1:

```
gao@GAO-PC ~>Desktop>COD>Assignment>Assignment 1
> .\1.exe
数字a: a = 4294967295 = -1 = 0xffffffff
数字b: b = 2147483648 = -2147483648 = 0x80000000
数字c: c = 2147483648 = -2147483648 = 0x80000000
数字d: d = 2147483648 = -2147483648 = 0x80000000
数字d的内存地址: 0x64fe10
数字d的机器码 = 0x80000000
```

第一个程序主要目的是将输入的数字, 分别以无符号**十进制整数**, **有符号十进制整数**, **无符号十六进制整数**输出, 同时还输出了 d 在内存中的地址以及 d 的机器码.

a: a 为-1,在计算机中以补码储存为 1111 1111 1111 1111 1111 1111 1111 1111(0xffffffff), 所以当作无符号整数输出时则是 $2^{32}-1 = 4294967295$.

b: b 为 2147483648, 由于共 32 位且,若高 1 位作为符号位, 所以剩余 31 位实际上无法表示该数值,因此当以有符号整数输出时,溢出为-2147483648, 即 1000 0000 0000 0000 0000 0000 0000 0000(0x80000000), 作为无符号输出时则正常为 2147483648.

c: c 为-2147483648, 在计算机中以补码存储为 1000 0000 0000 0000 0000 0000 0000 0000 (0x80000000). 所以作为无符号整数输出时候为 $2^{31} = 2147483648$.

d: d 被定义为无符号整数-2147483648, 所以作为有符号数输出时正常表示为-2147483648, 作为无符号数输出时为 $2^{31} = 2147483648$.

程序 2:

```
gao@GAO-PC ~>Desktop>COD>Assignment>Assignment 1
> .\2.exe
c=a+b=1+2147483647=-2147483648
f=e+d=-1+-2147483648=2147483647
```

a+b: a 和 b 均为有符号 int 数，最高位 32 位为符号位，剩余 31 位最大数字表示为 $2^{31}-1 = 2147483647$ ，所以 2147483648 上溢，变为 -2147483648。

e+d: 于上述类似，有符号负数最小被表示为 -2147483648(1,000.....0)，无法表示

-2147483648-1 = -2147483649，同样上溢，变为 2147483647。

程序 3:

```
gao@GAO-PC ~>Desktop>COD>Assignment>Assignment 1
> .\3.exe
sizeof(char):1
sizeof(short):2
sizeof(int):4
sizeof(unsigned int):4
sizeof(long):4
sizeof(long long):8
sizeof(size_t):8
sizeof(void *):8
sizeof(iarray):20
sizeof(iarr):8
```

改程序输出了各种类型数据，以及所占的内存空间

Char 字符类型: 在任何机器中都占 1 字节 = 8 位，-128 ~ +127

Short 短整数类型: 占 2 字节 = 16 位，-32767 ~ +32768

Int 整型: -2147483648 ~ +2147483647

Unsigned int 无符号整数： 同样占 4 字节 = 32 位，0 ~ 4294967295

Long 长整型： 占 4 字节 = 32 位，与 int 表示范围相同

Long long 超长整型： 占 8 字节 = 64 位，-9223372036854775808 ~ +9223372036854775807

Size_t： size_t 类型定义在 cstdint 头文件中，该文件是 C 标准库的头文件 stddef.h 的 C++ 版 size_t 是标准 C 库中定义的，应为 unsigned int，其大小足以保证存储内存中对象的大小，所以位 8 字节 = 64 位

Void*： 指针存放一个内存地址，所以在 64 位系统中，为 8 字节 = 64 位

Iarray： iarray 是一个数组，其中存放了 5 个 int 类型的数字，所以 $5 \times 4 = 20$ 字节

Iarr： iarr 同样存放的是一个内存地址，所以同样为 8 字节，该指针指向数组 iarray