

AN04001

手持机中数据格式与 PC 的对比

作者：技术支持部 张 焯

一、概述：

手持机编程使用了 C 或 C++ 语言甚至可能是汇编语言，在对这些语言进行编译时，我们使用了 GCC 的工具，该工具对各种数据的定义（主要指尺寸方面）与 PC 中使用的其他工具（如 TURBO C, VC++, BORLAND C 等等）不尽相同，而手持机使用的 68000 类 CPU 与 PC 的 x86 类 CPU 在数据存放的方法（主要指高低字节的顺序方面）也正好相反。基于以上原因，用户手持机与 PC 机交换数据时要特别小心。

二、已知的一些不同点：

1、数据类型的尺寸：

C 中的基本类型有：

`char, int, double, long, short, float`

(以及 `signed, unsigned` 修饰符)

- 1) `char, short, long, double, float` 在两种机器中的尺寸定义是一样的，即 `char` 1 字节，`short` 2 字节，`long` 4 字节，`double` 8 字节，`float` 4 字节。
- 2) `int` 是省缺的整型数类型，在不同的编译系统中是不一样的，有些编译系统将省缺 `int` 按 `short` 处理，有些按 `long` 处理。所以建议用户不要使用 `int` 这样不明确的定义，而是使用 `short` 或 `long` 这样明确的定义，这样在不同的编译系统之间易于移植以及不同机器间交换数据。

- 3) `long double` 不是 IEEE 754 中规定的浮点类型，但在不同系统中其尺寸是不同的。ANSI C 中其长度是 16 个字节，在 SPARC/POWERPC 中它是 16 个字节，但在手持机 GCC 中它是 12 字节，在 PC 中它是 10 字节。所以这些数据要在手持机与 PC 或工作站之间交换是非常困难的，用户最好采用其他的方法。
- 4) 相同的结构变量，在不同的系统中因为“数据对齐”的原因，其尺寸也可以是不同的，有些机器是按双字节对齐的，有些是按 4 字节对齐的。

2、基本类型数据的存放顺序：

手持机的 68000 类 CPU 是高字节在先的，PC 的 x86 型 CPU 正好相反。比如一个长整型数 `0x12345678`，在手持机中存放顺序是（地址由小至大）：`0x12,0x34,0x56,0x78`，而在 PC 中是：`0x78,0x56,0x34,0x12`。

所以一个基本类型数据，如果长度大于 1，又需要在手持机与 PC 间交换，则需要程序处理这种情况，将顺序反序后才可以。

数组等其他非基本数据，其数组元素存放的顺序，两种机器是相同的，即元素 0 在元素 1 的前面，元素 2 在元素 3 的前面，但元素内的基本类型数据的存放如前述是相反的。

字符串因其他基本类型是 `char`，长度是 1 字节，故在不同机器间存放的顺序是一样的。

三、 实验：

如果用户需要了解不同系统的数据尺寸以及存放顺序，可以采用参考如下程序片断来实验：

```
{
    float fl;
    double db;
    long double ldb;
    unsigned char *tptr, tcnt;
    fl = db = ldb = 1.0;
    for (tcnt = 0, tptr = &fl; tcnt < sizeof(fl); tcnt ++, tptr++)
        printf(tstr, "%02X ", *tptr);
    printf("\n");
}
```

```
for (tcnt = 0, tptr = &db; tcnt < sizeof(db); tcnt ++, tptr++)
    printf(tstr,"%02X ", *tptr);
printf("\n");
for (tcnt = 0, tptr = &ldb; tcnt < sizeof(ldb); tcnt ++, tptr++)
    printf(tstr,"%02X ", *tptr);
printf("\n");
}
```

四、参考资料

- 1.<<MC998 DEVELOPER'S MANUAL V3.2>>
- 2.<<MC2002 DEVELOPER'S MANUAL V1.1>>