0D 00 00 00 00 04 6E 61 6D 65 06 05 64 61 76 69 64　　　　　　对饮数据
����□name□□david

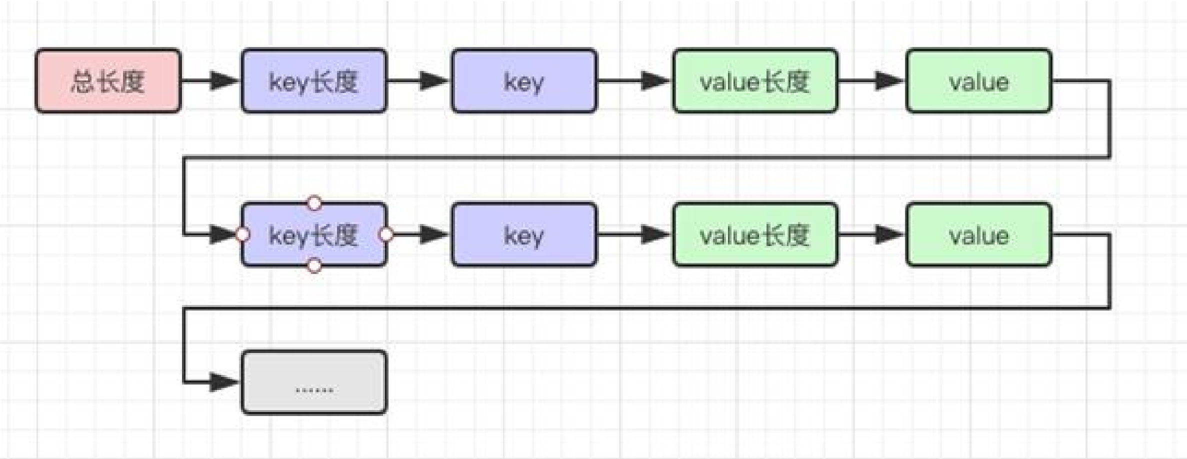
三、Protobuf协议

1、 什么是protobuf协议

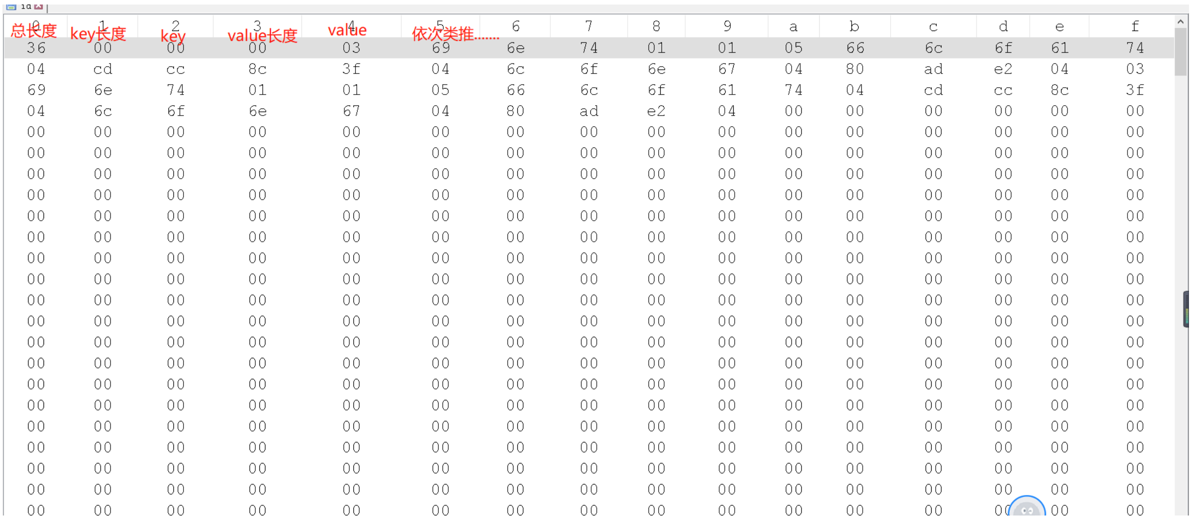
protobuf 是google开源的一个序列化框架，类似xml，json，最大的特点是基于二进制，比传统的XML表示同样一段内容要短小得多。

MMKV正式基于protobuf协议进行数据存储，存储方式为增量更新，也就是不需要每次修改数据都要重新将所有数据写入文件了。

2、数据结构



protobuf是二进制存储格式，第一位代表的是key和value的总长度，后面是key长度->key， value长度->value。。。。。 依次排列，可以用二进制查看工具来看一下：



```
0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
36 00 00 00 03 69 6e 74 01 01 05 66 6c 6f 61 74
04 cd cc 8c 3f 04 6c 6f 6e 67 04 80 ad e2 04 03
69 6e 74 01 01 05 66 6c 6f 61 74 04 cd cc 8c 3f
04 6c 6f 6e 67 04 80 ad e2 04 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

3、写入方式

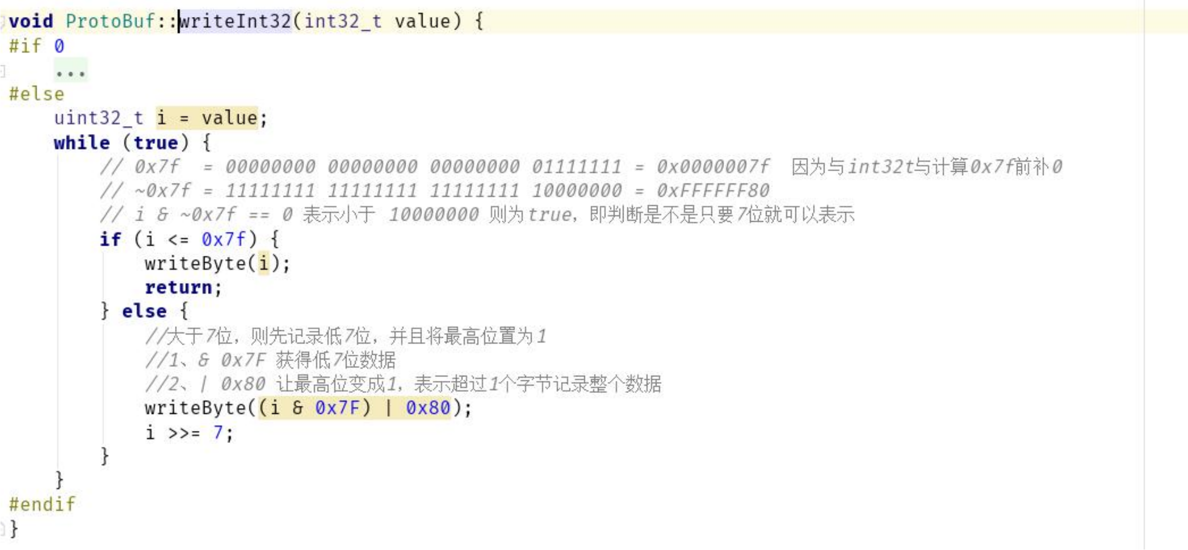
为了方便理解，这里拿**整型编码**举例：

1个字节保存7位数据，第1位为标志位

> 如果写入的数据 <= 0x7f 那么使用7位即1个字节表示，完成编码

> 如果写入的数据 > 0x7f 那么先记录低7位数据，并将最高位设为1，继续执行判断

16进制	0x7f
10进制	127
2进制	0111 1111



```
void ProtoBuf::writeInt32(int32_t value) {
#if 0
    ...
#else
    uint32_t i = value;
    while (true) {
        // 0x7f  = 00000000 00000000 00000000 01111111 = 0x0000007f  因为与int32t与计算0x7f前补0
        // ~0x7f = 11111111 11111111 11111111 10000000 = 0xFFFFFF80
        // i & ~0x7f == 0 表示小于 10000000 则为true，即判断是不是只要7位就可以表示
        if (i <= 0x7f) {
            writeByte(i);
            return;
        } else {
            //大于7位，则先记录低7位，并且将最高位置为1
            //1、& 0x7F 获得低7位数据
            //2、| 0x80 让最高位变成1，表示超过1个字节记录整个数据
            writeByte((i & 0x7F) | 0x80);
            i >>= 7;
        }
    }
#endif
}
```

以上代码就是整型的写入方式。

4、举例

为了更好的说明编码和解码的原理，我们拿一个整型来进行计算： 318242

编码

首先将318242进行二进制转换

1. 编码318242 -》

0100 1101 1011 0010 0010 （步1）

1. 大于0x7f，取最低7位，最高位补1：

1010 0010 ------------》写出到文件

1. 将步1的数据右移7位

0000 0000 1001 1011 0110 （步2）

1. 再取低7位，最高位补1

1011 0110 ------------》写出到文件

1. 再将步2的数据右移7位

0000 0000 0000 0001 0011（步3）

1. 再取低7位，最高位补1

1001 0011 ------------》写出到文件

1. 再将步3的数据右移7位

0000 0000 0000 0000 0000 （步4）

1. 再取低7位，因为这7位小于0x7f，不需要补1，直接写出

0000 0000 -----------》写出到文件

经过以上8个步骤，protobuf就为318242编码完成了。

解码

拿到上面写入的几组数据：

1. 1010 0010
2. 1011 0110
3. 1001 0011
4. 0000 0000

然后从后面往前拼接：（注意一下，因为前3个数据只有后7位是有效数据，拼接时要去掉首位）

1）将4拼接到3前面：

0000 0000 001 0011

2）再将上面的数据拼接到2前面：

0000 0000 001 0011 011 0110

3）再拼接到1前面：

0000 0000 001 0011 011 0110 010 0010

去除无效位：

0 001 0011 011 0110 010 0010

这样就得到了原二进制码（0100 1101 1011 0010 0010）。