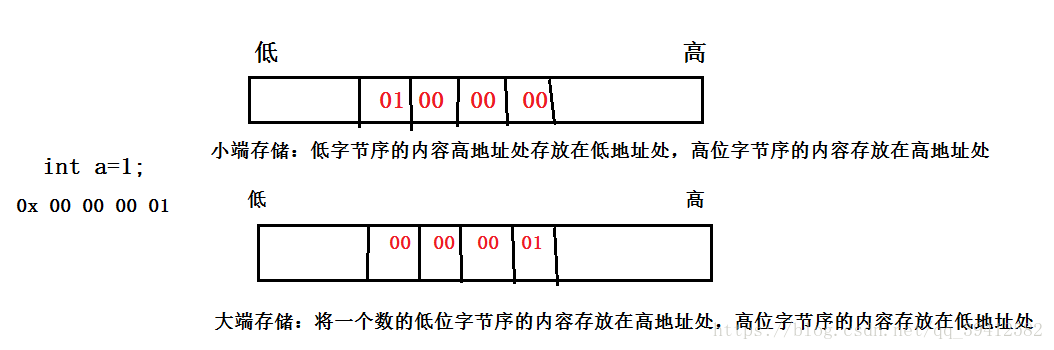
**大小端**

1. **大小端名字的由来及发展**
2. 在乔纳森·斯威夫特的著名讽刺小说《格列夫游记》中，小人国内部分裂成Big-endian和Little-endian两派，区别在于一派要求从鸡蛋的大头把鸡蛋打破，另一派要求从鸡蛋的小头把鸡蛋打破。斯威夫特借以讽刺英国的政党之争，在计算机工业中指数据储存顺序的分歧。
3. 后来计算机通信发展起来后，遇到一个问题就是：在串口等串行通信中，一次只能发送1个字节。这时候我要发送一个int类型的数就遇到一个问题。int类型有4个字节，我是按照：byte0 byte1 byte2 byte3这样的顺序发送，还是按照byte3 byte2 byte1 byte0这样的顺序发送。规则就是发送方和接收方必须按照同样的字节顺序来通信，否则就会出现错误。这就叫通信系统中的大小端模式。这是大小端这个词和计算机挂钩的最早问题。
4. 现在我们讲的这个大小端模式，更多是指计算机存储系统的大小端。在计算机系统中，我们是以字节为单位存放数据的，每个地址单元都对应着一个字节，一个字节为8bit。但在C语言中存在不同的数据类型，占用的字节数也各不相同，那么就存在怎样存放多个字节的问题，因此就出现了大端存储模式和小端存储模式。
5. 现实的情况就是：有些CPU公司用大端（譬如C51单片机）；有些CPU用小端（譬如ARM）。（大部分是用小端模式，大端模式的不算多）。于是乎我们写代码时，当不知道当前环境是用大端模式还是小端模式时就需要用代码来检测当前系统的大小端。
6. **大小端定义**

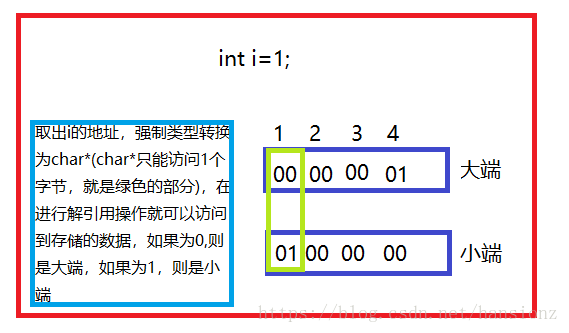
* 大端存储模式：是指数据的低位字节序保存在内存的高地址中，而数据的高位字节序保存在内存的低地址中
* 小端存储模式：是指数据的低位字节序保存在内存的低地址中，而数据的高位字节序保存在内存的高地址中



**3. 怎么设计一个程序来判断当前机器是大端存储还是小端存储**

①方法一

算法思路：



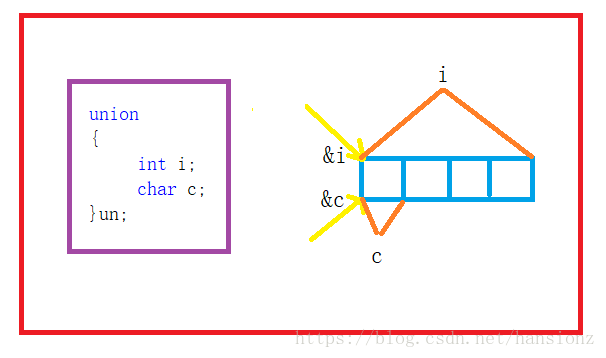
参考代码：



②方法二

算法思路：定义一个联合体，根据联合体成员起始地址相同巧妙的访问到了第一个字节地址的数据。

联合体类型特点：



参考代码：

