Lecture 14：知识 - 数据的对齐和大小端存放方式

大端方式与小端方式

概念：大端方式与小端方式

- 记忆：
  - 大端高存低；
    - 大端方式时，低位地址存储高位内容（大端内容）
  - 小端低存低；
    - 小端方式时，低位地址存储低位内容（小端内容）
- 概念：
  - 现代计算机基本使用**字节**编址，即：每个地址编号中，存放 `1` 字节；
  - 在存储数据时，不同类型的数据占用的字节数不同，数据从低位到高位可以按照两种方式排列；
  - 1. 从左到右排列 `->` **大端**方式：先存储**高位**字节，再存储低位字节，字中字节顺序和原序列**相同**；
  - 2. 从右到左排列 `->` **小端**方式：先存储**低位**字节，再存储高位字节，字中字节顺序和原序列**相反**；
  - 因此通常用最低有效字节（`LSB`）和最高有效字节（`MSB`）来分别表示数据的低位和高位；
- 注意：
  - 阅读以小端方式存储的机器代码时，要注意字节是按相反顺序显示的；
  - 小端方式时，字节间反序，字节内顺序；
  - 大端方式时：字节间顺序，字节内顺序；
- 分析：
  - 假设变量 `i` 地址为 `0800 H` ，字节 `01H、23H、45H、67H`
  - 两种排列方式：

	0800H	0801H	0802H	0803H	
大端方式	…	01H	23H	45H	67H	…

	0800H	0801H	0802H	0803H	
小端方式	…	67H	45H	23H	01H	…

- 分析：
  - 题目：

记得续更哦 同步微信：Wang02277

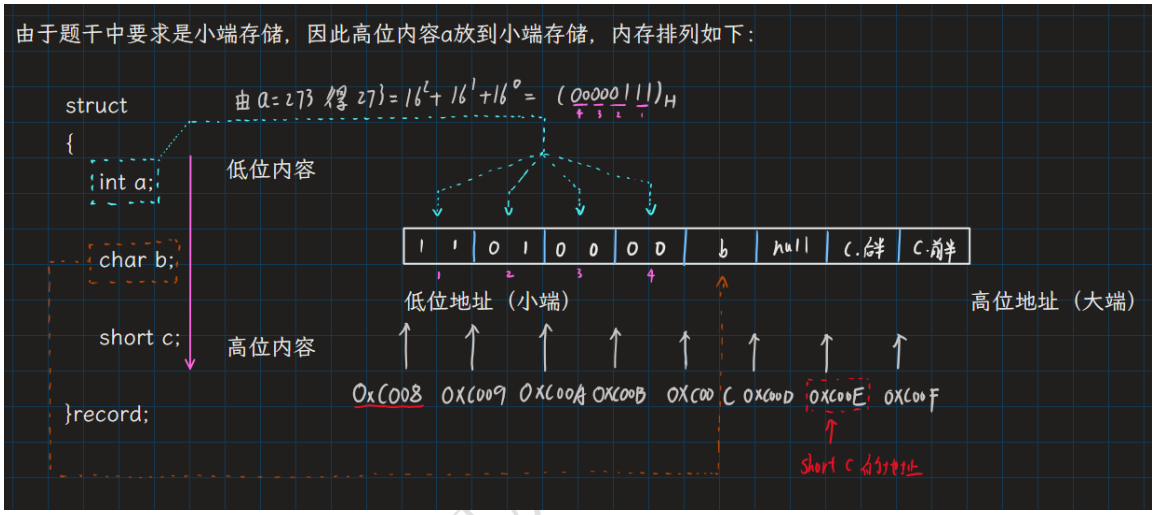
- 【2012 统考真题】某计算机存储器按字节编址，采用小端方式存放数据。假定编译器规定 int 型和 short 型长度分别为 32 位和 16 位，并且数据按边界对齐存储。某 C 语言程序段如下：

```
struct{
int         a;
char        b;
short       c;
}record;
record.a = 273;
```

若 record 变量的首地址为 0xC008，地址 0xC008 中的内容及 record.c 的地址分别为（ ）。

A. 0x00, 0xC00D　B. 0x00, 0xC00E　C. 0x11, 0xC00D　D. 0x11, 0xC00E

  - 解析：



数据对齐存储

概念：数据对齐存储

- 原因：
  - 现代计算机都是按字节编址的，假设字长为 `32` 位，数据按边界对齐方式存放；
  - 要求其存储地址是自身大小的整数倍，半字地址一定是 2 的整数倍，字地址一定是 4 的整数倍；
  - 这样无论所取的数据是字节、半字还是字，均可一次访存取出；
  - 当所存数据不满足上述要求时，可通过填充空白字节使其符合要求。这样做虽然会浪费一些存储空间，但可以提高存取数据的速度；
  - 当数据不按边界对齐方式存储时，半字长或字长的数据可能在两个存储字中，此时需要两次访存，并对高低字节的位置进行调整后才能得到所需数据，从而影响了系统的效率；
- 两个要求：
  - 1. 每个成员按照其类型大小对其：`char` 对齐值 = 1、`short` 对齐值 = 2、`int` 对齐值 = 4（单位都为字节）
  - 2. `struct` 的长度必须是**成员中最大对齐值的整数倍**，不够时需要补空字节；
- 图示：

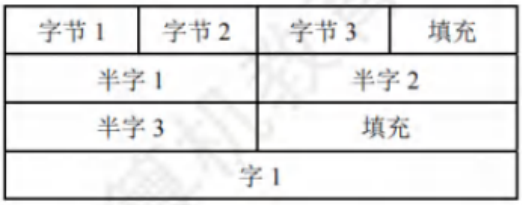


图 2.13　按边界对齐方式存储

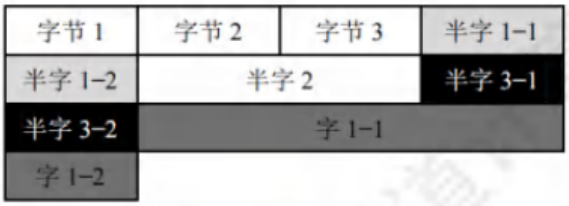


图 2.14　按边界不对齐方式存储

获取笔记持续更新可加微信：WandD2277