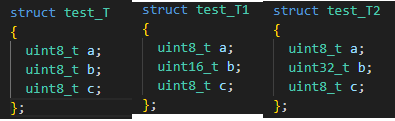
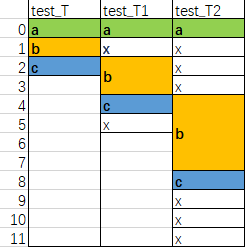
对齐的规则是,每个成员按其类型的对齐参数(通常是这个类型的大小)和指定对齐参数(这里是 n 字节)中较小的一个对齐，即：min( n, sizeof( item )) 。并且结构的长度必须为所用过的所有对齐参数的整数倍,不够就补空字节。

1、对于下面三个结构体，sizeof的大小分别为3、6、12。

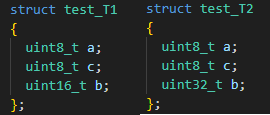


他们对于在内存中的排列如下所示

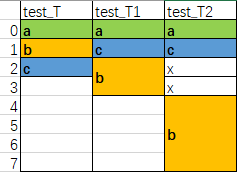


变量自身是几个字节，那么就要放到能整除的地址处。1字节可以在任意地址处；2字节则必须在0、2、4、。。。等被2整除的地址处。4字节则必须在0、4、8.。。等能被4整除的地址处。另外，结构体整体的大小要能被成员最大字节数整除。比如test\_T1最大成员字节数是2，所以结构体大小不能是5，必须是6，最后还得补一个空位。test\_T1最大成员字节数是4，所以结构体大小不能是9，最后必须补三个空位达到12才OK。

改变结构体成员的顺序可以减少空位。对于T1和T2，如果如下排列，则结构体大小变为4和8.



内存中的排列变为



在实际应用中如果忽略以上对齐问题，那么可能会使用错误的结构体大小，导致数据收发不匹配、错位等问题。所以遇到结构体第一时间就要考虑对齐的问题，如果不放心最好用sizeof测试一下，以保万无一失。