

结构体或者类会把成员对齐到该类型大小的整数地址，结构体总大小会是最大的基础类型的整数倍

举几个例子：

```
1 struct Struct {
2     int64_t a; // 8 bytes 0x7ffe909d7d70
3     int8_t b; // 1 byte 0x7ffe909d7d78
4     int16_t c; // 2 bytes 0x7ffe909d7d7a
5     // 总大小: 16 bytes
6 };
```

```
1 struct Struct {
2     int8_t b; // 1 byte 0x7ffc5007ce70
3     int64_t a; // 8 bytes 0x7ffc5007ce78
4     int16_t c; // 2 bytes 0x7ffc5007ce80
5     // 总大小: 24 bytes
6 };
```

```
1 struct ComplexStruct {
2     char a; // 对齐为1字节 0x7ffc5b472590
3     double b; // 对齐为8字节 0x7ffc5b472598
4     int c; // 对齐为4字节 0x7ffc5b4725a0
5     char d; // 对齐为1字节 0x7ffc5b4725a4
6     float e; // 对齐为4字节 0x7ffc5b4725a8
7     short f; // 对齐为2字节 0x7ffc5b4725ac
8     long long g; // 对齐为8字节 0x7ffc5b4725b0
9     // 总大小: 40 bytes
10 };
```