

怎么强制不让结构体对齐

结构体对齐：方便CPU查找；因为 cpu的查找，为了提高效率，不是一个一个字节来的

模板T 可以有 默认类型

<template T=char> 然后,当在函数中 无法自动推导出函数类型时，会使用此默认类型！ 但是 函数传的参数的默认初始值，不会作为模板参数推导依据

```
<template T,template U=char>
void fun(T=100,U=100){
    cout<<U<<endl;
    cout<<T<<endl;           // 报错 因为T无法自动推导出来类型，且没给出 默认参数 （形参 100不能
}
```

assert 断言 （相当于强制判断语句，若此处断言结果为假，则程序退出）

如： assert(size>0); (程序运行到此处时， size若《0，则退出)

静态断言：不同环境下断言结果不同，需要等程序跑在一个确定的环境/平台下，才可确定：

assert(sizeof(long long)==8); 用来判断程序是否在64位环境下运行，若不是则退出程序

throw 抛出异常 noexcept 在函数后面修饰，表明此函数不会出错，若出错，则终止程序运行

*** throw ***; 抛出异常 try {执行逻辑} catch (抓住异常信息) {执行逻辑} catch(抓住抛出的异常变量){执行逻辑}

`noexcept` 后可跟参数

`noexcept(true)` 表明函数不会异常

`noexcept(false)` 表明函数可能异常