静态联编和动态联编

多态

- 多态是面向对象程序设计语言中数据抽象和继承之外的第三个基本特征。
- 多态性(Polymorphism)提供接口与具体实现之间的另一层隔离，从而将"what"和"how"分离开来。多态性改善了代码的可读性和组织性，同时也使创建的程序具有可扩展性，项目不仅在最初创建时期可以扩展，而且当项目在需要有新的功能时也能扩展。
- C++ 支持编译时多态(静态多态)和运行时多态(动态多态)，运算符重载和函数重载就是编译时多态，而派生类和虚函数实现运行时多态。
- 静态多态和动态多态的区别就是函数地址是早绑定(静态联编)还是晚绑定(动态联编)。

静态联编和动态联编

- 程序调用函数时，将使用哪个可执行代码块呢？编译器负责这个问题。
- 在C语言中，这非常简单，因为每个函数名都对应一个不同的函数。在 C++ 中，由于函数重载的缘故，这项任务更复杂。编译器必须查看函数参数以及函数名才能确定使用哪个函数。
- 将源代码中的函数调用解释为执行特定的函数代码块被称为函数名联编（binding）。
- 如果函数的调用，在编译阶段就可以确定函数的调用地址，并产生代码，就是静态联编（static binding），就是说地址是早绑定的，又称为早期联编（early binding）。
- 而如果函数的调用地址不能在编译期间确定，而需要在运行时才能决定，就是动态联编（dynamic binding），就是说地址是晚绑定，又称为晚期联编（late binding）。

Thanks