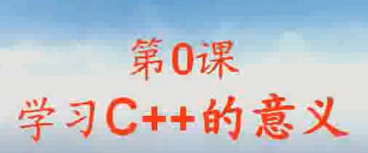
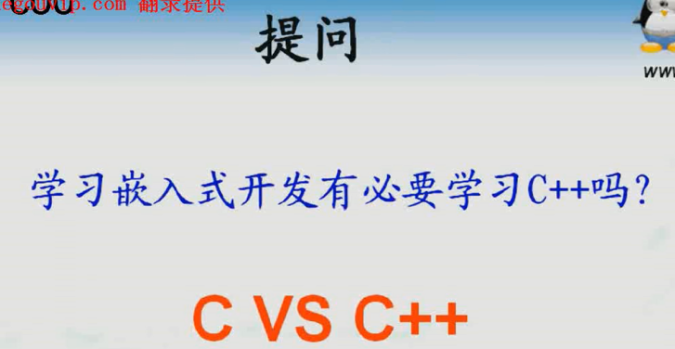
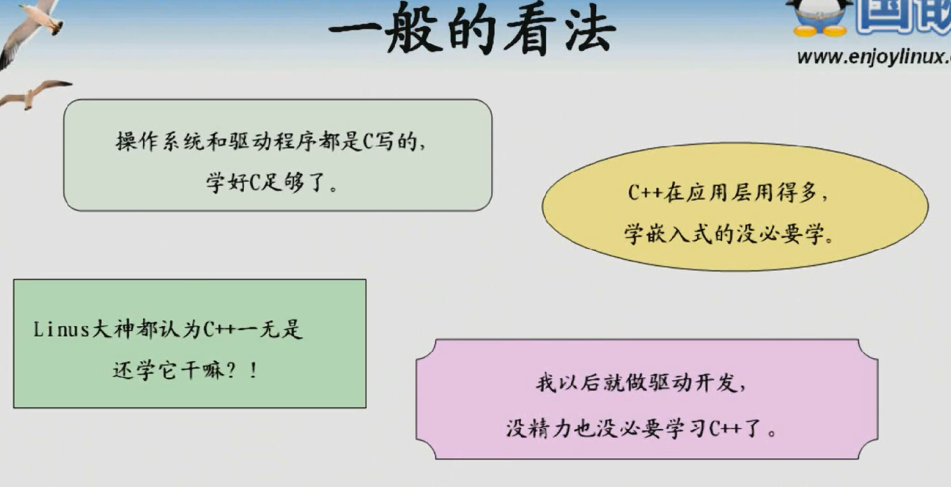
第0课 - 学习c++的意义

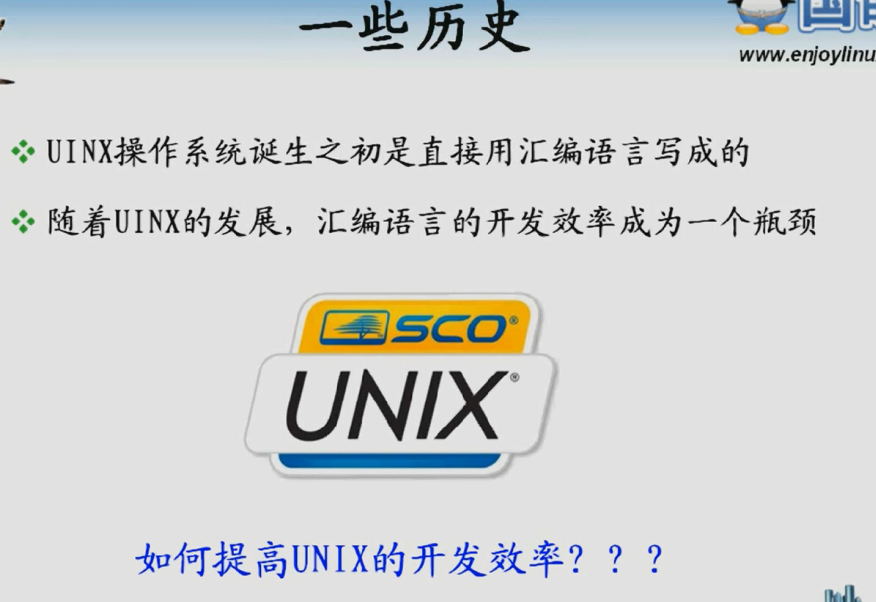


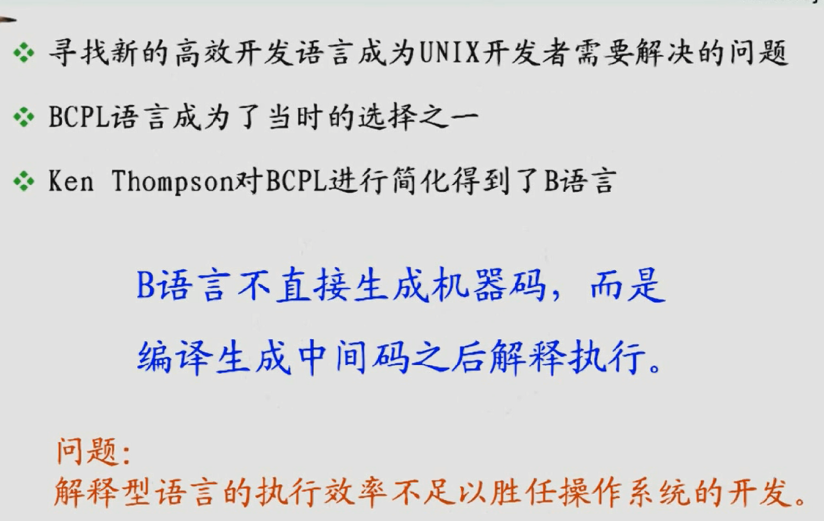


很多人认为嵌入式学习里面 c和c++是对立的



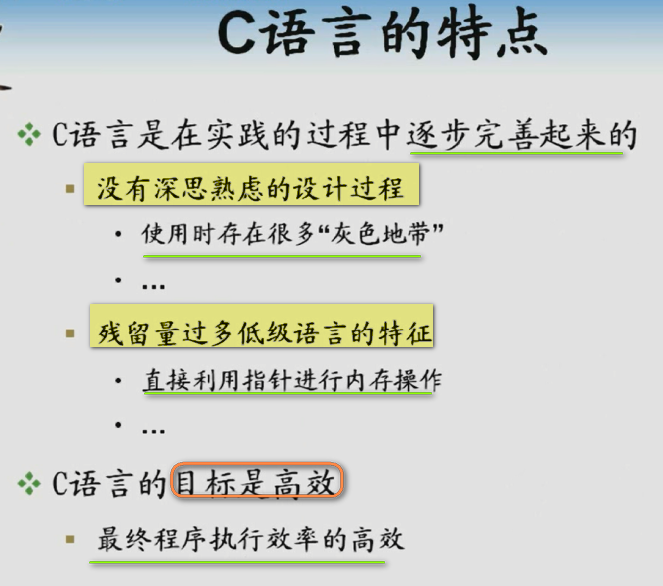
嵌入式系统 也是应用级的系统 --- 出厂之前 软件和硬件牢牢结合的 ----- 最终用户无法修改最终的软件和硬件 所以叫嵌入式





中间码解释执行 这个是java语言的执行原理 ---- 解释语言的效率无法胜任操作系统的开发

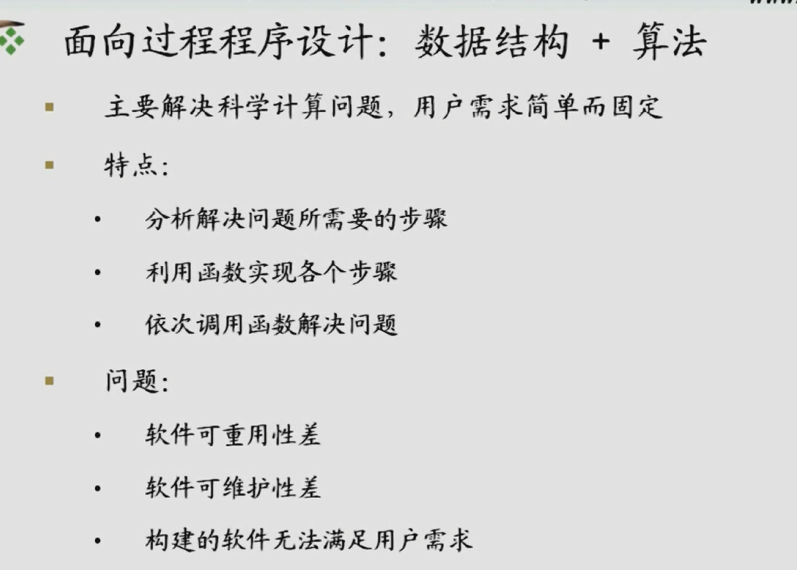




C语法没有深思熟虑 没想到以后会广泛被使用 只是为了实现Unix操作系统 所以 C语言高效就好 ---- C语言的目标就是高效

从这段历史 得出一个结论 ---- C语言天生为操作系统的开发存在的 C就适合开发操作系统

有了OS 终端用户也无法解决问题

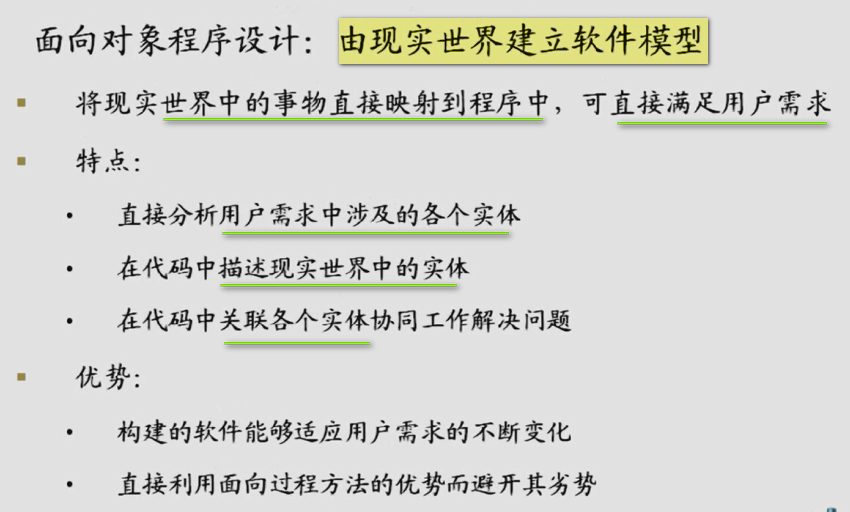


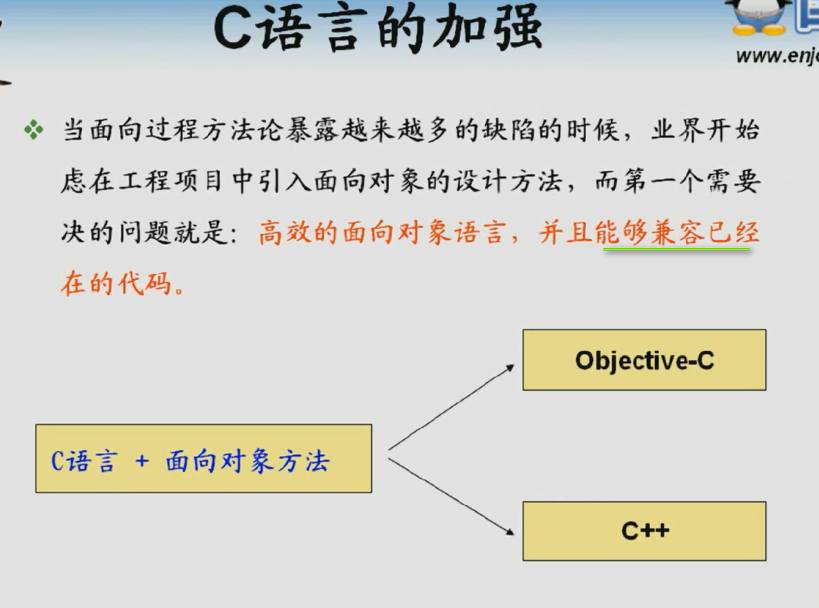
OPP是C的方法论 ---- 就是数据结构 + 算法

**解决科学计算 有一个特点 就是 问题逐步分解 自上而下 分解为一个个步骤 只需要写函数 依次调用 然后问题就能解决**

但是 软件领域 和计算机是发展的 后期就是软件可重用性差

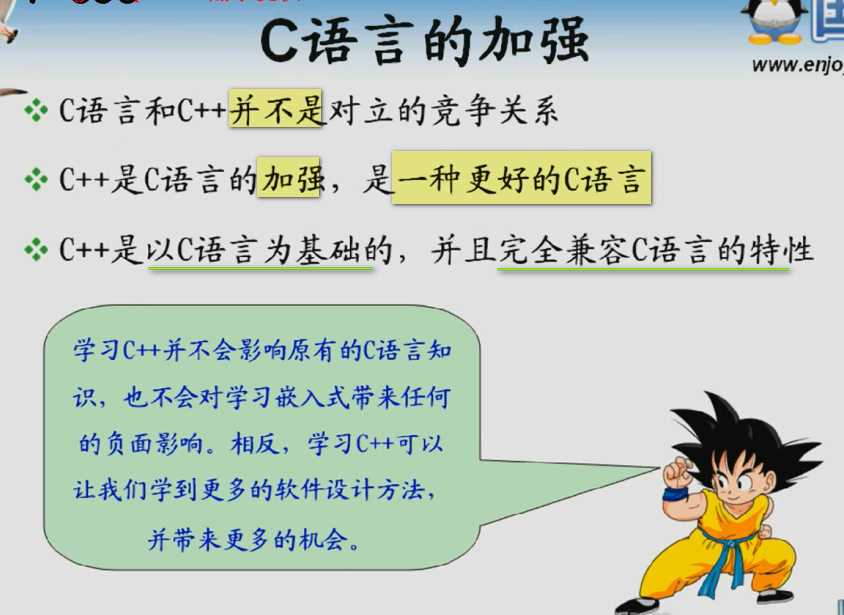
这时候 软件没有办法再发展了 ----- 也没法继续更新了 --- 软件领域发展受到了阻碍



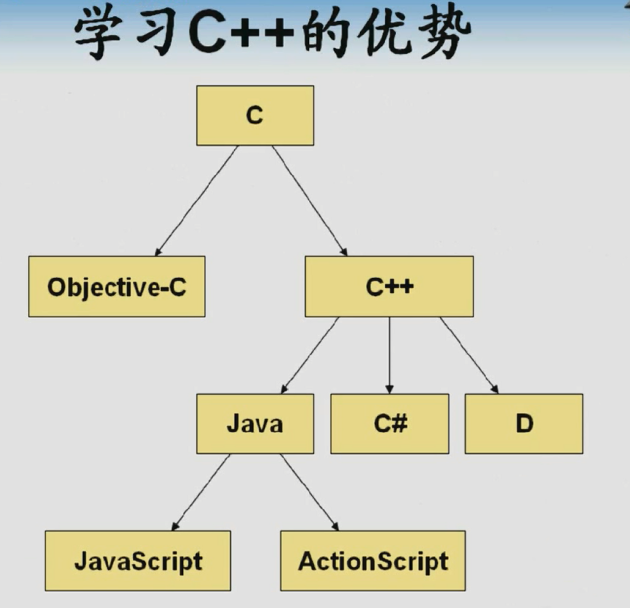


在已有的C语言的基础上 加上面向对象的方法 打造一门新的语言 来满足高效和 兼容现有的代码

这样有了两条路 ----- Objective-C 苹果公司采用的 另一种是C++



会了C语言 就会了一半的C++



C++使用的过程中 有了新的继承者 Java C#

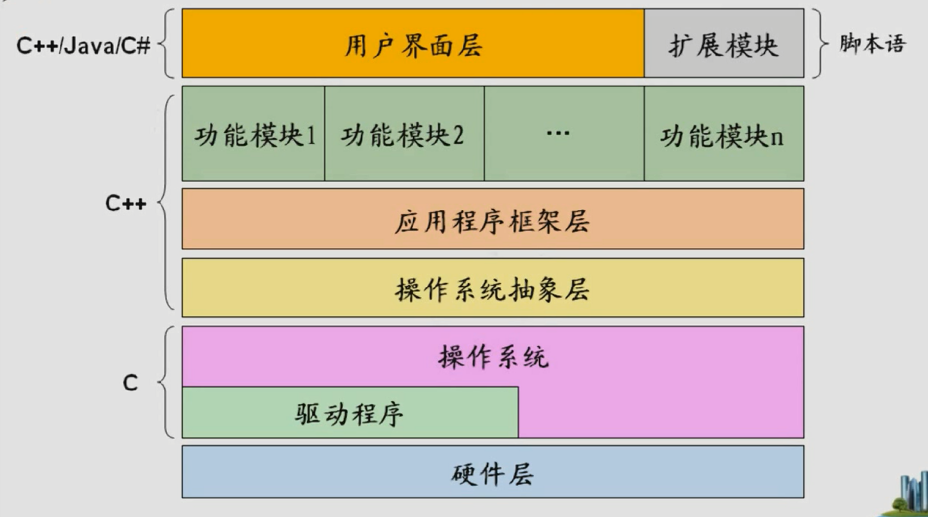
Java是C++语言的子集 --- 当年java语言的设计者 认为C++太复杂了 C++每次编程 就用了几个关键的特性 为什么不把特性抽象出来 成立一门新的语言呢？ 所以 java就出来了 比C++简单

C#是微软推出的 ------ 为了和Java抗衡 两者诞生的目的一样

要会C++ 最多学的快的 一个星期就可以会java和C# ---java和C#都是解释型语言 效率不够好

所以C++继续向前发展 生成了D语言 ----- D非常高效 不是中间码 而是 直接编译成可运行的码

现代软件架构图



最下面是硬件层



