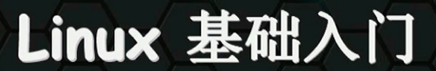
操作系统和linux的基础知识



CPU是运算器 和控制器组成

CPU为什么执行运算 ---Intel和摩托罗拉的CPU的运算器的功能一样么？

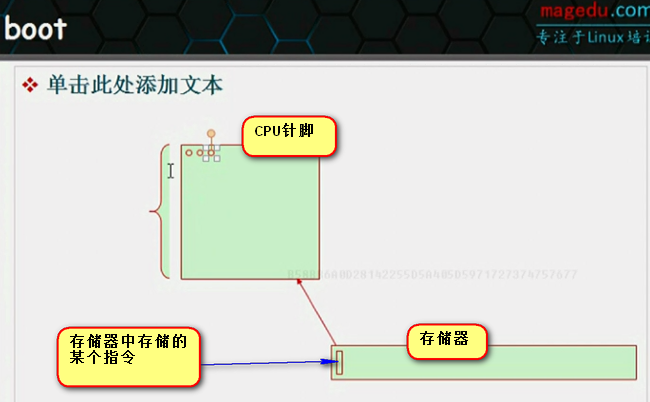
CPU内部有众多的电子芯片 晶体管 纳米

晶体管做到纳米级别 里面很多门电路 ---- 多个门电路组合到一起 实现的功能是不一样的

CPU就是一个芯片 有很多针脚 ---- 每一个针脚的功能不一样 最为基础的功能 ---- 为了让CPU完成自己期望的运算 ----- **指挥调用某一个针脚的功能**

任何一个计算芯片 必须有一个**基本的载入或者boot的过程** ---- 无论怎么进行

比如与CPU从内存中特定地址的位置 取出一个数据 或者一个控制指令 --- CPU开机自动获取 --- 这里面取出存储在存储器中的指令 --- 这个指令就是能够调用CPU某个针脚的基本功能的指令



每一个CPU提供的功能 都通过指令来实现 多个指令联合起来 就叫指令集

同一个CPU 为了完成不同的功能 里面有不同的芯片组成

完成多媒体处理 和完成整数运算 两回事

完全两个不同的芯片来完成

同一个CPU内部 不同的芯片组 ---- 所以 同一个CPU里面提供很多的指令集

每一个CPU厂商提供的指令集也是不一样的

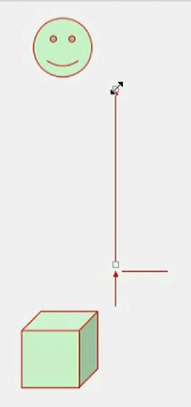
其实从本质上 CPU只能运行二进制格式的指令

二进制格式编程 这样的能力很少

所以 这种提供的编程接口 太少

------- 所以编程语言有高级语言 低级语言

一种编程语言 离机器越近 就是低级语言 离人类越近 就是高级语言

 我们使用哪种语言写的程序 性能好？低级语言

PHP JSP C语言 都算是高级语言

真正的低级语言 就是机器语言 --- 过于底层 所以每一个芯片 再生产完成时候 都会附带接口 V码编程接口 汇编语言 ----- 写大型程序很难 思考非常复杂的问题 非常困难

高级语言 人能看懂 但是计算机看不懂

所以 人能看懂的 都要转换成低级语言

==========

从机器 到人之间 有三层 机器语言 汇编语言 高级语言

高级语言 写完的程序 要经过这么两次之后 才能运行

---- 首先转换成芯片支持的汇编 ==== 第一阶段

---- 转换成对机器指令的调用 ==== 第二阶段

**对于C语言来讲 第一个阶段称为编译** ----- Intel x86 另一个是arm的CPU 他们的底层指令 汇编接口不一样 但是支持同一种高级语言   
 ---- 这个取决于底层的CPU的芯片是什么格式的

第二步 – 汇编语言 变成机器代码的过程 是汇编的过程 ---- 汇编是由高级语言转换成目标代码 成为object code 然后再转换成汇编指令 是这样的过程

编排的操作 就是程序 --- 所以 程序员就是干这件事的

非常复杂的问题 不同厂商的指令集不一样

存储大小不一样 ----- 同一个程序 要告诉 你的存储需要多大

------ 这样 挑战太大 因为写一个程序要考虑我底层有多少资源可以使用 告诉多大内存可以使用 ---- 这个是相当麻烦的

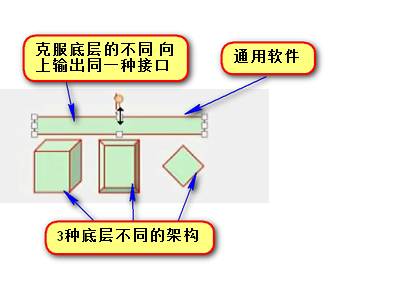
**要到达到灵活使用的目的 就是要消除底层差异**

要进行多个国家的旅行 8个国家 需要自己学 或者找8个翻译 ----太麻烦

如果可以找一个翻译 会8国语言 只需要进行一个翻译交互 就可以完成底层的交互 ---- 有一个翻译来应付所有的底层不同

----- 为了避免每一个程序员写程序的时候 来应付底层的不同 但是 向上输出接口是统一的 ----- 这就是**通用软件形式**

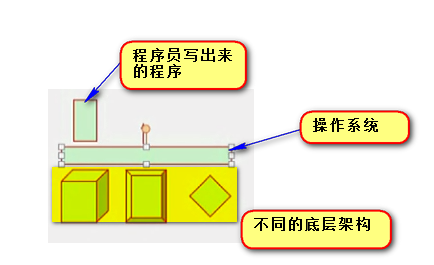
下面有两种机器结构 提供一个层次 向上只说一种语言

 这个通用软件 就是操作系统

提到的linux Windows android ios都是这么一种通用程序

Linux可以装在x86 arm 都可以

底层谁生产的CPU 都可以对程序员都是透明的 ---- 对程序员的写程序 都是透明的 ----- 所以 这个时候 程序员仅仅需要面向OS写程序即可



所以 操作系统通常是一个**软件程序** ---- 将底层提供的各种计算能力 给抽象成统一接口的这么一种程序 无论底层是什么样子的CPU 都可以计算能力的CPU

所以 程序员写程序 就大大的简化了

这就是 计算机拿来 先安装操作系统 再安装应用软件 因为这些软件都是面向操作系统编程的

===== 脱离了操作系统 也可以运行程序 这样就要面向硬件写程序

这就是什么是OS

OS Operating System

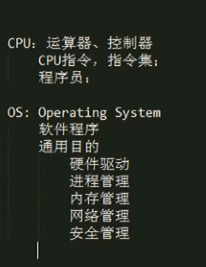
底层功能：让硬件工作起来 --- 硬件驱动

把CPU的时间 分配给众多程序员 应用 所以 就是时间片切割 和分配 ---- 进程管理

将有限的内存资源分配给同时运行的程序 ---- 内存管理

和其他主机进行网络通讯 ---- 网络管理

还包括一些安全管理



所以 有一台打印机 让这台打印机工作起来 --- 只要在OS上面装这个驱动程序来实现 但是要真的打印 该怎么打印 要通过一个特定的应用程序来完成

此时 不再面向硬件写程序 面向操作系统来写程序 所以 操作系统本身就是一个虚拟机

所有的操作过程 都是面向OS 不是面向计算机硬件

操作系统将计算机接口虚拟出来

计算机有很多指令 是特权指令 --- 拥有管理功能

普通指令 --- 普通功能 基本操作

----- **注意 其中 我们程序员面向OS写的应用程序 都只能调用的是普通指令 只有操作系统才能调用特权指令**  
 --- 如何实现

整个CPU实现的指令集 分成了四个环 0 1 2 3四个环 所有的特权指令环0上 所有的普通指令都在环3上面 ---- 历史原因 1 2上面这两个环没有指令运行

所以 OS运行的时候 可以运行环0和环3上面的指令 但是 普通的应用程序只能运行环3上面的普通指令

22:42 第一天第二讲