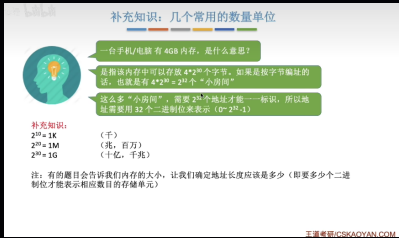


内存基础

内存概念

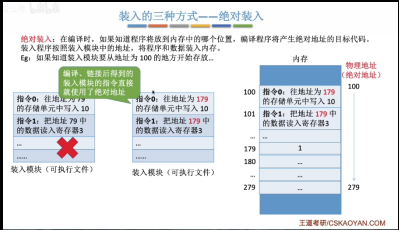
- 定义
  - 可以存放数据，执行前先放到内存才能被CPU处理，缓和CPU与硬盘之间的速度矛盾
- 概念
  - 内存中有存储单元 —— 存储单元 存放数据的最小单元
  - 内存地址从0地址开始，每个地址对应一个存储单元
  - 计算机按字节编址 每个存储单元大小为1字节 —— 8个2进制位
  - 计算机按字变址 —— 字长为16位计算机 每个字大小就是16位
- 数量单位
  - K M G
  - B 字节 b 比特位



指令的工作原理

- 图解 1
  - @计算机组成原理
- 图解 2
  - 逻辑地址 —— 相对地址 相对于进程起始地址的地址
  - 物理地址 —— 实际在内存中的地址
  - 逻辑地址转入物理地址

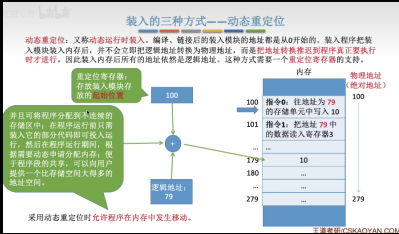
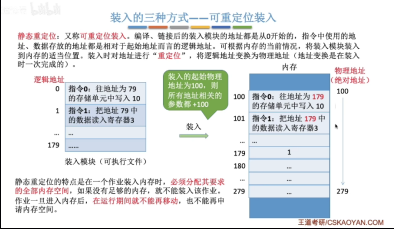
编译时，知道程序放入内存中的那个位置。编译时，将产生绝对地址的目标代码。装入程序按照装入模块中的地址，将程序和数据装入内存



灵活性差 只适用于单道程序环境

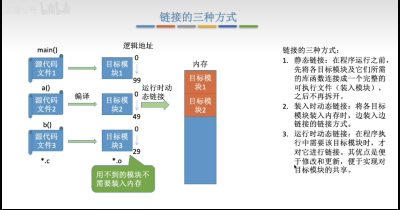
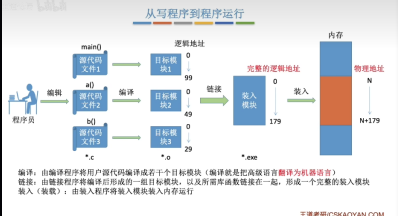
地址转换问题

- 绝对装入 (编译时)
- 重定位装入 (装入内存时)
  - 装入后进行重定位，逻辑转入物理地址 ——
  - 要求:
    - 必须分配其要求的全部内存空间
    - 一次性装入全部
    - 内存空间连续
    - 运行期间不可再移动
- 动态重定位
  - 存入逻辑地址，地址转换推迟到程序真正执行时进行 —— 装入后依旧是逻辑地址，需要重定位寄存器得到 —— 重定位寄存器存入装入模块存放的起始位置支持
  - 图解 ——
  - 很方便于移动
  - 可分配到不连续存储区

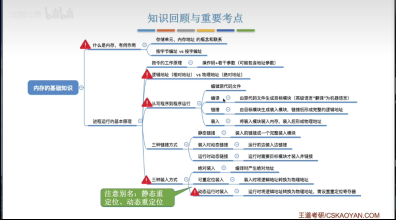


从写程序到程序运行

- 源代码文件到执行 ——
- 链接的方式
  - 静态链接 —— 运行前就链接成功
  - 装入时动态链接 —— 边装入边链接
  - 运行时动态链接 —— 执行到该目标模块才会链接 —— 便于修改与更新
- 链接的三种方式 ——



回顾



要点

- 三种装入方式
  - 绝对装入
  - 静态重定位装入
  - 动态重定位装入
- 三种连接方式
  - 静态链接
  - 装入时链接
  - 运行时链接