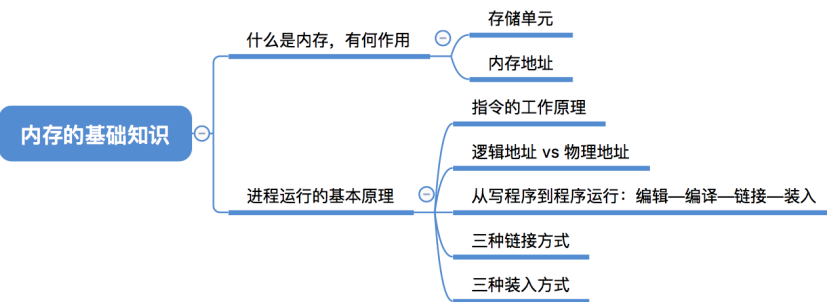


本节内容

## 内存的基础知识

王道考研/CSKAOYAN.COM

### 知识总览



王道考研/CSKAOYAN.COM

### 什么是内存? 有何作用?



新品 小米Max3 全网通版 4GB内存 黑色  
6.9"巨无霸全面屏, 5500mAh充电宝级大电量, 纤薄全

京东价 ¥1699.00 降价通知  
促销 赠品 0 × 1 (赠完即止)  
换购 购买1件可优惠换购热销商品 立即

增值业务 高价回收-卖了换钱 4G套餐18元  
配送至 北京海淀区三环以内 有货 支持  
由 京东 发货, 并提供售后服务, 11:10前下

重量 0.47kg

选择颜色 黑色 金色

王道考研/CSKAOYAN.COM

### 什么是内存? 有何作用?



联想(Lenovo)拯救者Y7000 15.6英寸游戏笔记本电脑(英特尔八代酷睿i5-8300H 8G : GTX1050Ti 黑)

信仰"Y"Logo灯效, 超窄边全面屏, TSX散热, CPU显卡性能全速发挥, 2T大硬盘, PCIe3.0 【拯救者新品6期白条】

京东秒杀  
秒杀价 ¥6199.00 [¥6699] 降价通知  
优惠券 满99减10  
促销 限购 仅购买1件时享受单价¥6199, 超出数量以结算价为准  
限制 此价格不与套装优惠同时享受  
换购 购买1件可优惠换购热销商品 立即换购 >>

配送至 北京海淀区五环到六环之间 有货 支持 99元免基础运费(20kg内) 京准达 211限时达

王道考研/CSKAOYAN.COM



方案：给内存的存储单元编地址



这么多“小房间”，需要  $2^{32}$  个地址才能一一标识，所以地址需要用 32 个二进制位来表示 ( $0 \sim 2^{32}-1$ )

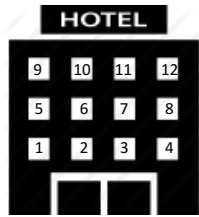
$2^{10} = 1\text{K}$  (千)  
 $2^{20} = 1\text{M}$  (兆, 百万)  
 $2^{30} = 1\text{G}$  (十亿, 千兆)

王道考研/CSKAOYAN.COM



## 逻辑地址 vs 物理地址

宿舍四个人一起出去旅行，四个人的学号尾号分别是 0、1、2、3。  
住酒店时酒店给你们安排了4个房号相连的房间。四个人按学号递增次序入住房间。比如 0、1、2、3 号同学分别住进了 5、6、7、8 号房间。  
四个人的编号 0、1、2、3 其实是一个“相对位置”，而各自入住的房间号是一个“绝对位置”。  
只要知道 0 号同学住的是房号为 N 的房间，那么 M 号同学的房号一定是  $N+M$ 。  
也就是说，只要知道各个同学的“相对位置”和“起始房号”，就一定可以算出所有同学的“绝对位置”



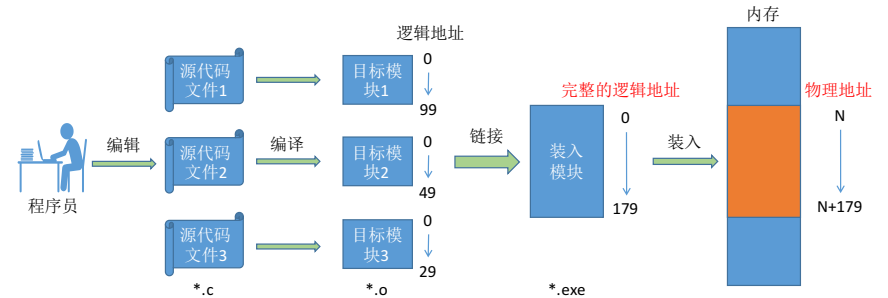
指令中的地址也可以采用这种思想。编译时产生的指令只关心“相对地址”，实际放入内存中时再想办法根据起始位置得到“绝对地址”。

Eg: 编译时只需确定变量 x 存放的相对地址是 100（也就是说相对于进程在内存中的起始地址而言的地址）。CPU 想要找到 x 在内存中的实际存放位置，只需要用进程的 起始地址+100 即可。

相对地址又称逻辑地址，绝对地址又称物理地址。

王道考研/CSKAOYAN.COM

## 从写程序到程序运行



编译：由编译程序将用户源代码编译成若干个目标模块（编译就是把高级语言翻译为机器语言）  
链接：由链接程序将编译后形成的一组目标模块，以及所需库函数链接在一起，形成一个完整的装入模块  
装入（装载）：由装入程序将装入模块装入内存运行

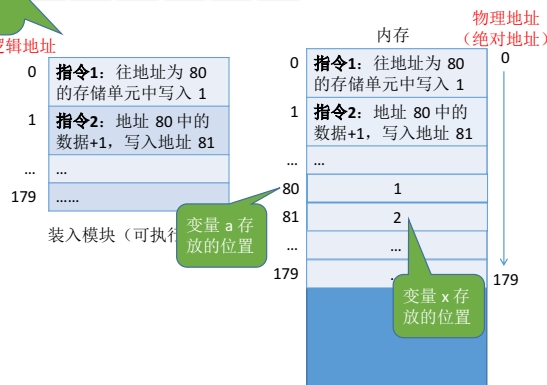
王道考研/CSKAOYAN.COM

## 装入模块装入内存

装入模块中的指令地址指的是“相对地址”，即：相对于开始地址而言的地址。  
相对地址又称逻辑地址。

C语言程序翻译成机器语言指令后：

```
int a = 1;
int x = a + 1;
.....
```



王道考研/CSKAOYAN.COM

## 装入模块装入内存

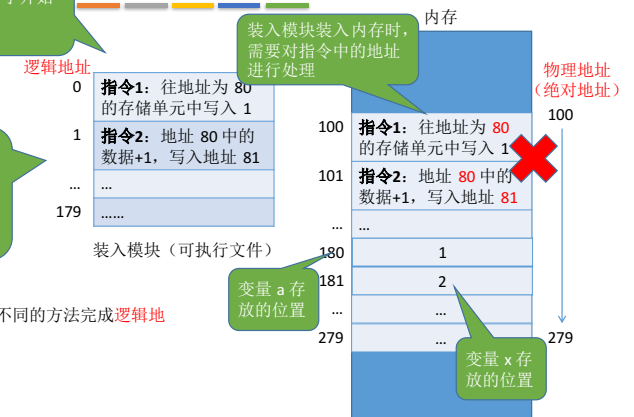
装入模块中的指令地址指的是“相对地址”，即：相对于开始地址而言的地址。  
相对地址又称逻辑地址。

C语言程序翻译成机器语言指令后：

```
int a = 1;
int x = a + 1;
.....
```

装入的三种方式（用三种不同的方法完成逻辑地址到物理地址的转换）：

1. 绝对装入
2. 静态重定位
3. 动态重定位

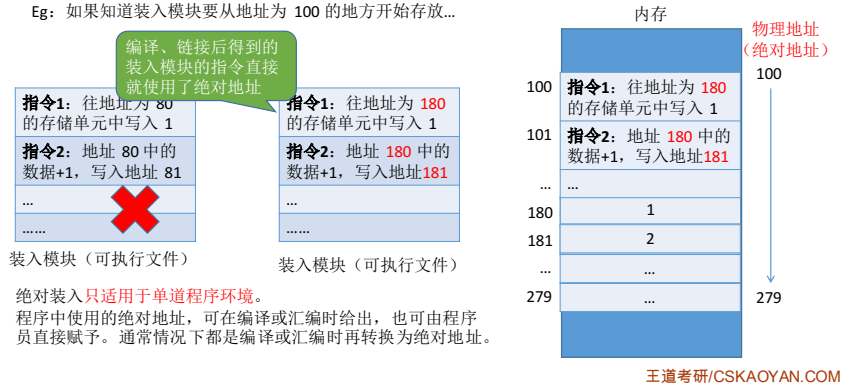


王道考研/CSKAOYAN.COM

## 装入的三种方式——绝对装入

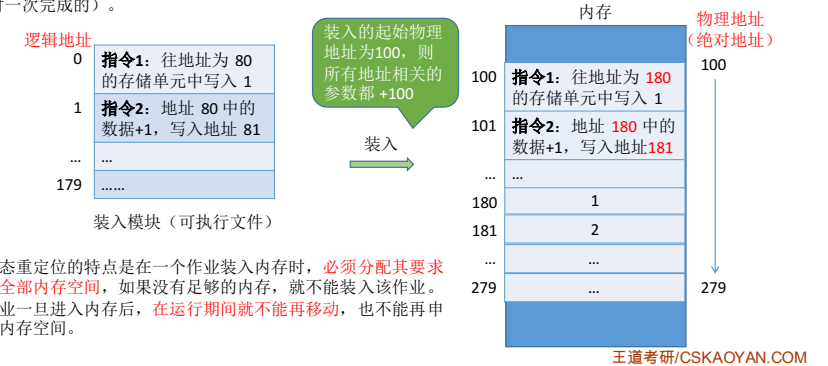
**绝对装入：**在编译时，如果知道程序将放到内存中的哪个位置，编译程序将产生绝对地址的目标代码。装入程序按照装入模块中的地址，将程序和数据装入内存。

Eg：如果知道装入模块要从地址为 100 的地方开始存放...



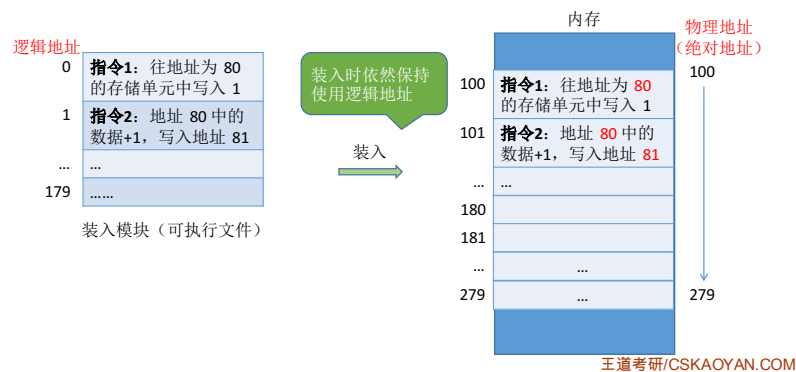
## 装入的三种方式——静态重定位

**静态重定位：**又称**可重定位装入**。编译、链接后的装入模块的地址都是从0开始的，指令中使用的地址、数据存放的地址都是相对于起始地址而言的逻辑地址。可根据内存的当前情况，将装入模块装入到内存的适当位置。装入时对地址进行“**重定位**”，将逻辑地址变换为物理地址（地址变换是在装入时一次完成的）。



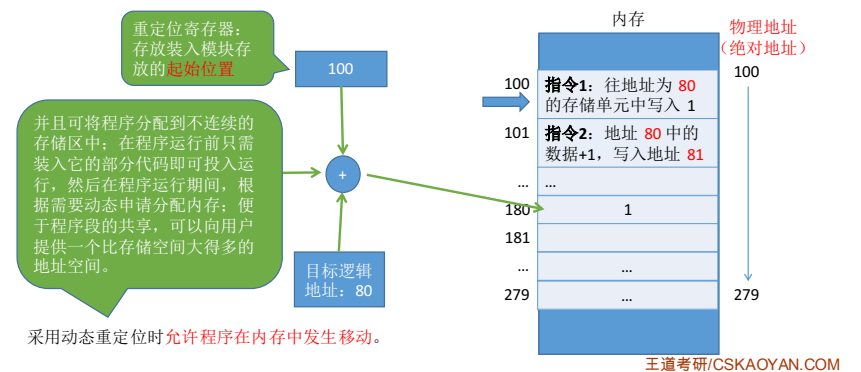
## 装入的三种方式——动态重定位

**动态重定位：**又称**动态运行时装入**。编译、链接后的装入模块的地址都是从0开始的。装入程序把装入模块装入内存后，并不会立即把逻辑地址转换为物理地址，而是把**地址转换推迟到程序真正要执行时才进行**。因此装入内存后所有的地址依然是逻辑地址。这种方式需要一个**重定位寄存器**的支持。

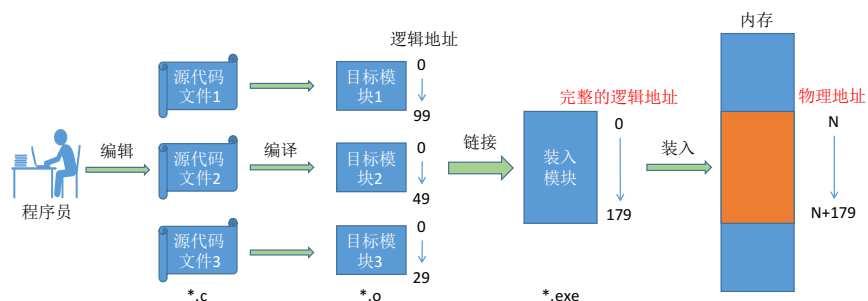


## 装入的三种方式——动态重定位

**动态重定位：**又称**动态运行时装入**。编译、链接后的装入模块的地址都是从0开始的。装入程序把装入模块装入内存后，并不会立即把逻辑地址转换为物理地址，而是把**地址转换推迟到程序真正要执行时才进行**。因此装入内存后所有的地址依然是逻辑地址。这种方式需要一个**重定位寄存器**的支持。



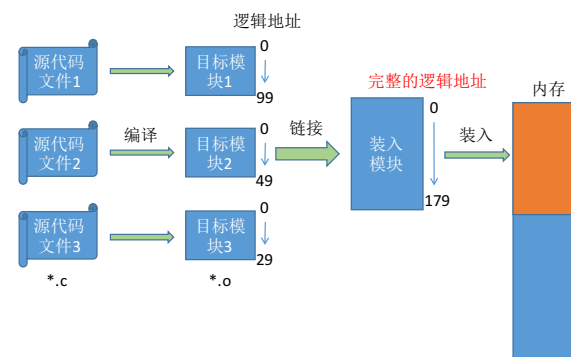
## 进程运行的基本原理



编译：由编译程序将用户源代码编译成若干个目标模块（编译就是把高级语言翻译为机器语言）  
链接：由链接程序将编译后形成的一组目标模块，以及所需库函数链接在一起，形成一个完整的装入模块  
装入（装载）：由装入程序将装入模块装入内存运行

王道考研/CSKAOYAN.COM

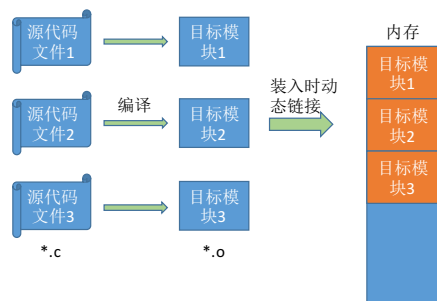
## 链接的三种方式



链接的三种方式：  
1. 静态链接：在程序运行之前，先将各目标模块及它们所需的库函数连接成一个完整的可执行文件（装入模块），之后不再拆开。

王道考研/CSKAOYAN.COM

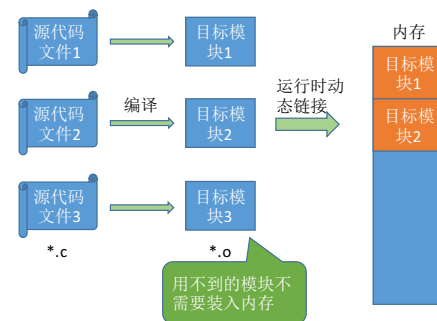
## 链接的三种方式



链接的三种方式：  
1. 静态链接：在程序运行之前，先将各目标模块及它们所需的库函数连接成一个完整的可执行文件（装入模块），之后不再拆开。  
2. 装入时动态链接：将各目标模块装入内存时，边装入边链接的连接方式。

王道考研/CSKAOYAN.COM

## 链接的三种方式



链接的三种方式：  
1. 静态链接：在程序运行之前，先将各目标模块及它们所需的库函数连接成一个完整的可执行文件（装入模块），之后不再拆开。  
2. 装入时动态链接：将各目标模块装入内存时，边装入边链接的连接方式。  
3. 运行时动态链接：在程序执行中需要该目标模块时，才对它进行链接。其优点是便于修改和更新，便于实现对目标模块的共享。

王道考研/CSKAOYAN.COM

## 知识回顾与重要考点

