

红色边框为一级知识点：熟悉

橙色边框为二级知识点：掌握

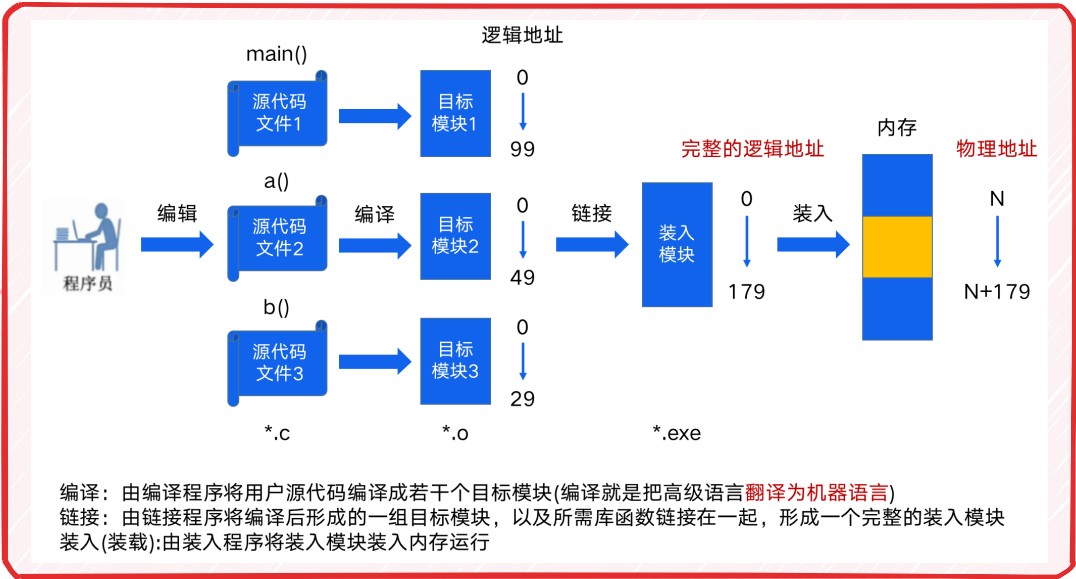
第三章 内存管理

内存管理的基本原理和要求

程序进入内存

最初的故事（哈工大）

- 创建进程时，首先在内存中找一块空闲的内存
- 将空闲地址（基地址）找到并赋值给PCB
- 程序载入到空闲内存后，PC置好初始地址然后执行，每次执行指令都要进行地址翻译
- 上下文切换时，PCB中的基址就进入基址寄存器



链接（概念见图片）

- 静态链接：程序运行前，将各模块及库函数链接成一个完整的可执行程序
- 装入时动态链接：将用户源程序编译后得到的目标模块，装入内存时，采用边装入边链接的方式
- 运行时动态链接：对于某些目标模块的链接，程序执行中需要才会对其链接

装入（概念见图片）

- 绝对装入：装入时按实际内存地址，将程序和数据装入内存
- 可重定位装入（静态重定位）：装入时对目标程序中指令和数据修改地址为物理地址，放入内存（作业必须一次性全部装入，运行时不能移动，不能申请内存空间）
- 动态运行时装入（动态重定位）：装入内存后不立即把地址转换，只有当程序执行时才进行地址翻译（运行时重定位，基址（重定位）寄存器）

重定位（哈工大）
修改程序中相对地址（逻辑地址）为绝对地址（物理地址）

时机 （对应装入三种）

- 编译时（只能放固定地址，逻辑地址=物理地址）
- 载入（装入）时（载入时修改地址，慢但灵活）：程序载入后需要移动，引入运行时重定位
- 运行时重定位：执行每条指令都要从逻辑地址算出物理地址（地址翻译），逻辑地址+基地址（从PCB中取）= 物理地址

逻辑（虚拟）地址空间和物理地址空间

- 物理地址空间：内存中物理单元的集合，进程在运行时执行指令和访问数据，都要通过物理地址从主存中存取
- 逻辑（虚拟）地址空间：即相对地址，链接程序依次按照各个模块的相对地址构成统一的从0单元开始编址的逻辑地址空间
实现分段与分页的关键所在，为了实现段页式引入，下文详述

内存保护

- 方案一：CPU设置上下限寄存器，存放用户作业在主存中的上限和下限，每当CPU访问地址，和寄存器比较，是否越界
- 方案二：内存管理机构动态将逻辑地址与界地址寄存器进行比较，若未越界，则加上重定位寄存器的值映射为物理地址（逻辑地址+基地址=物理地址）

可重入程序（纯代码，不可修改，非临界资源）通过减少对换数量改善系统性能

By: （b站）分享笔记的好人儿

个人感觉这章某道是讲的不太好的，仅仅分别介绍了几种机制，没有串在一起，多进程的问题记得是有考到，但某道好像只讲了段的共享

操作系统作为一个系统的东西，一定要把各个章节内容揉和在一起，章节内知识点揉和在一起，所以相较于计算机组成原理，可能操作系统看额外课程的提升会大一些（王道是最贴合408考纲，这是必听的）

重点考点：

- 1.编辑、编译、链接、装入（顺序、链接和装入的地址变化、链接和装入的三种不同方式）
- 2.分段分页、多级页表快表、段页式、请区分页
- 3.内存分区（动态分区分配的分区链接方式）
- 4.页面置换算法
- 5.分配策略、调页相关小点和抖动、工作集