

第四讲物理内存管理：连续内存分配

第 1 节计算机体系结构和内存层次

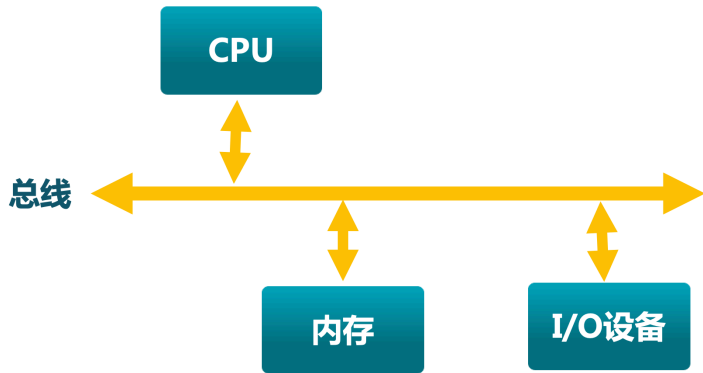
向勇、陈渝

清华大学计算机系

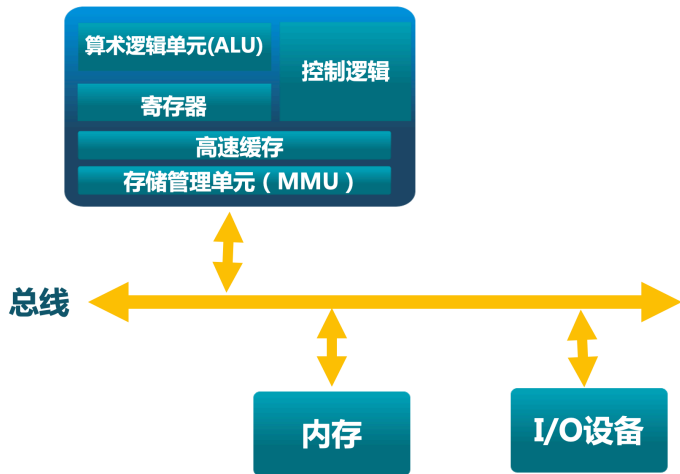
xyong,yuchen@tsinghua.edu.cn

2020 年 4 月 12 日

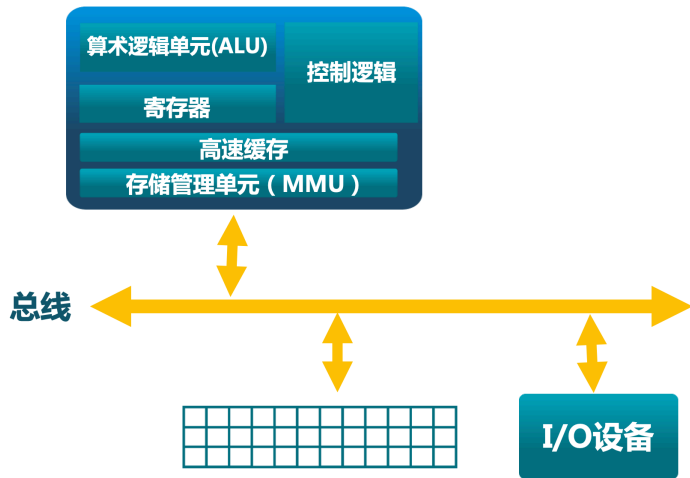
计算机体系结构



计算机体系结构



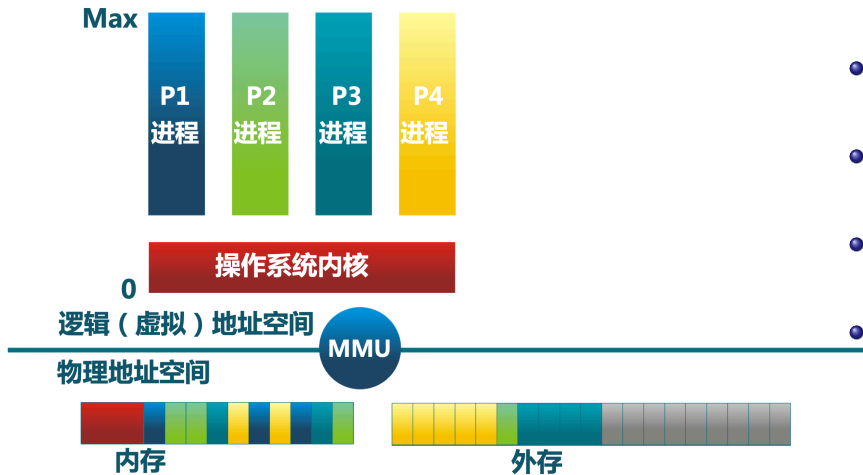
计算机体系结构



内存层次



操作系统的内存管理



- 抽象
 - 逻辑地址空间
- 保护
 - 独立地址空间
- 共享
 - 访问相同内存
- 虚拟化
 - 更大的地址空间

操作系统的内存管理方式

- 操作系统中采用的内存管理方式
 - 重定位 (relocation)
 - 分段 (segmentation)
 - 分页 (paging)
 - 虚拟存储 (virtual memory)
 - 目前多数系统 (如 Linux) 采用按需页式虚拟存储
- 实现高度依赖硬件
 - 与计算机存储架构紧耦合
 - MMU(内存管理单元): 处理 CPU 存储访问请求的硬件

第四讲物理内存管理：连续内存分配

第 2 节再看程序的地址空间

向勇、陈渝

清华大学计算机系

xyong,yuchen@tsinghua.edu.cn

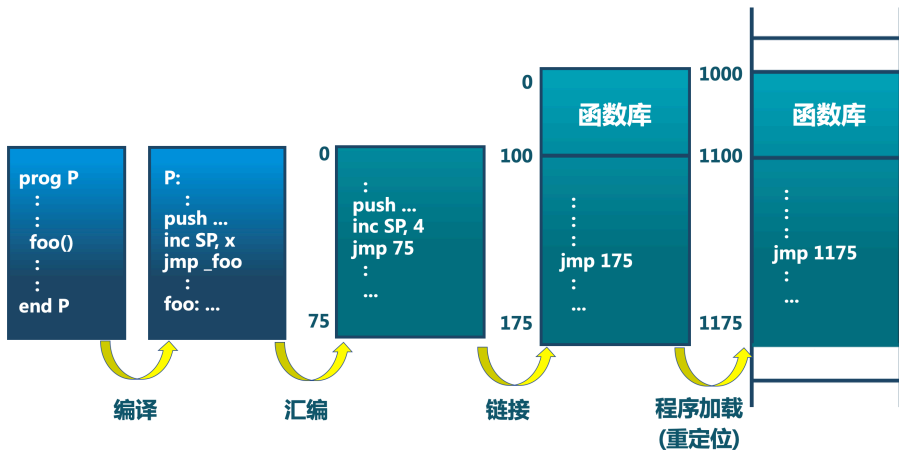
2020 年 4 月 12 日

地址空间定义



- 物理地址空间—硬件支持的地址空间
 - 起始地址 0, 直到 MAX_{sys}
- 逻辑地址空间—在 CPU 运行的进程看到的地址
 - 起始地址 0, 直到 MAX_{prog}

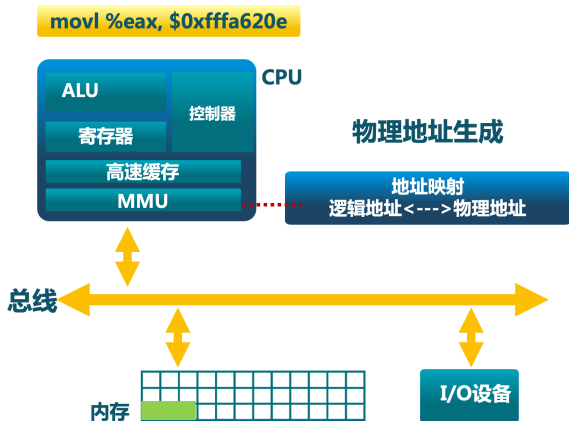
逻辑地址生成



地址生成时机和限制

- 编译时
 - 假设起始地址已知
 - 如果起始地址改变，必须重新编译
- 加载时
 - 如编译时起始位置未知，编译器需生成可重定位的代码 (relocatable code)
 - 加载时，生成绝对地址
- 执行时
 - 执行时代码可移动
 - 需地址转换 (映射) 硬件支持

地址生成过程



- CPU

- ALU: 需要逻辑地址的内存内容
- MMU: 进行逻辑地址和物理地址的转换
- CPU 控制逻辑: 给总线发送物理地址请求

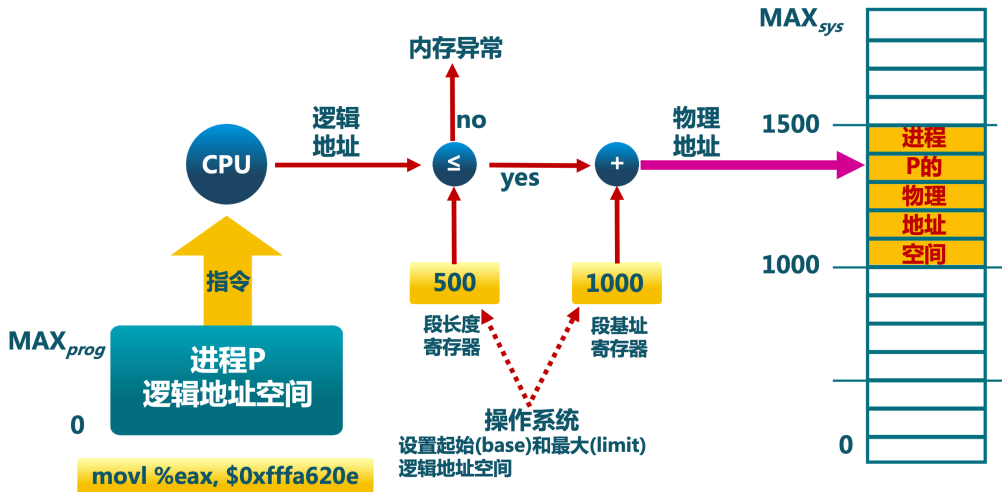
- 内存

- 发送物理地址的内容给 CPU
- 或接收 CPU 数据到物理地址

- 操作系统

- 建立逻辑地址 LA 和物理地址 PA 的映射

地址检查



第四讲物理内存管理：连续内存分配

第 3 节连续内存分配

向勇、陈渝

清华大学计算机系

xyong,yuchen@tsinghua.edu.cn

2020 年 4 月 12 日

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

第四讲物理内存管理：连续内存分配

第 4 节碎片整理

向勇、陈渝

清华大学计算机系

xyong,yuchen@tsinghua.edu.cn

2020 年 4 月 12 日

XXXX

XXXX

XXXX

第四讲物理内存管理：连续内存分配

第 5 节伙伴系统分配算法与实现

向勇、陈渝

清华大学计算机系

xyong,yuchen@tsinghua.edu.cn

2020 年 4 月 12 日

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

XXXX

第四讲物理内存管理：连续内存分配

第 6 节 SLAB 分配器

向勇、陈渝

清华大学计算机系

xyong,yuchen@tsinghua.edu.cn

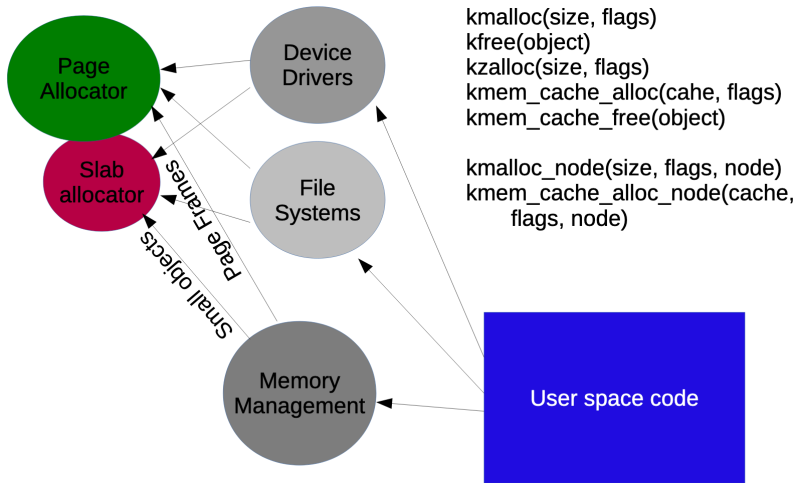
2020 年 4 月 12 日

rCore 中的物理内存管理

```
//Rust CODE
pub fn init(l: usize, r: usize)
pub fn init_allocator(l: usize, r: usize)
pub fn alloc_frame() -> Option<Frame>
pub fn alloc_frames(cnt: usize) -> Option<Frame>
pub fn dealloc_frame(f: Frame)
pub fn dealloc_frames(f: Frame, cnt: usize)
```

出处: [frame_allocator.rs](#)

与 SLAB 分配器相关的系统组成部件

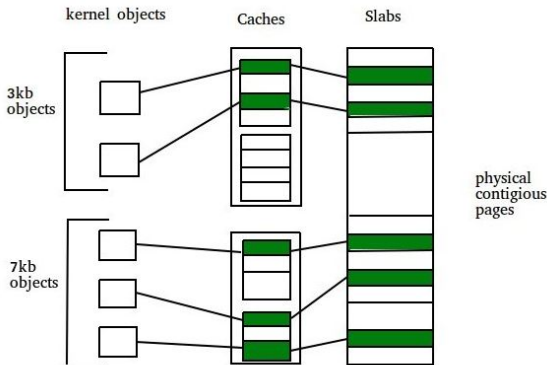


SLAB 分配器

SLAB 分配器源于 Solaris 2.4 的分配算法，工作于内存物理页分配算法之上，管理特定大小对象的缓存，进行快速高效的物理内存分配。

- 想解决的问题

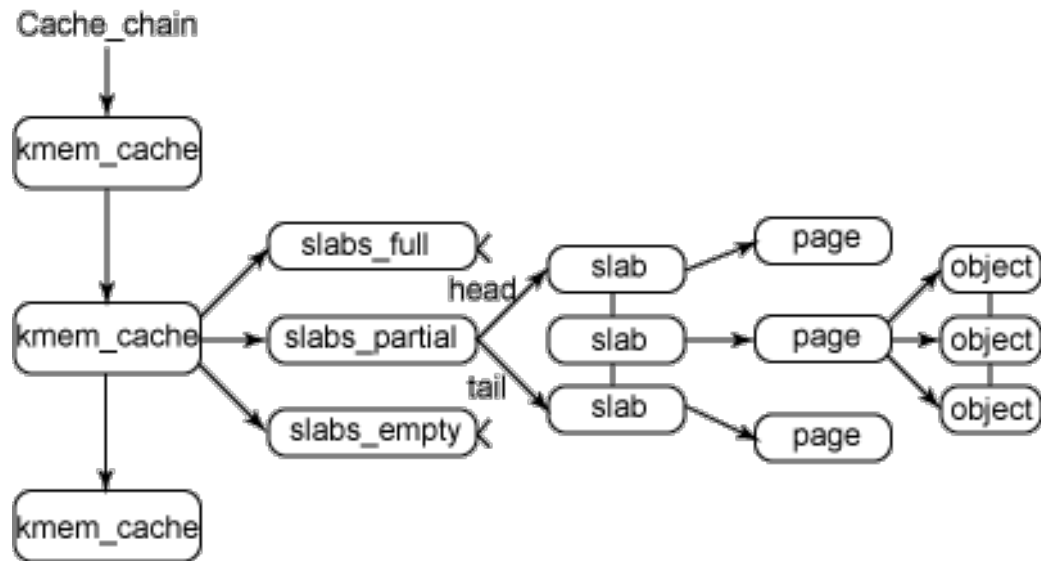
- 内核对象远小于页的大小
- 内核对象会被频繁的申请和释放
- 内核对象初始化时间超过分配和释放内存总时间



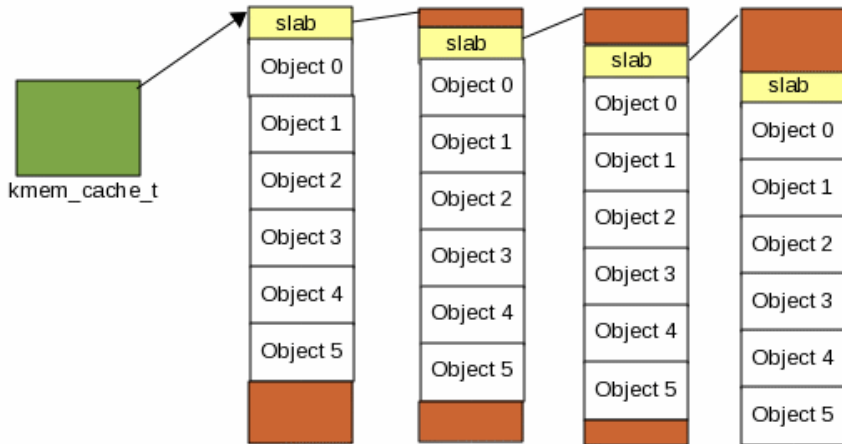
SLAB 分配器的特征

- 为每种使用的内核对象建立单独的缓冲区
- 按对象大小分组
- 两种 SLAB 对象状态：已分配或空闲
- 三类缓冲区队列：Full、Partial、Empty
- 优先从 Partial 队列中分配对象
- 缓冲区为每个处理器维护一个本地缓存

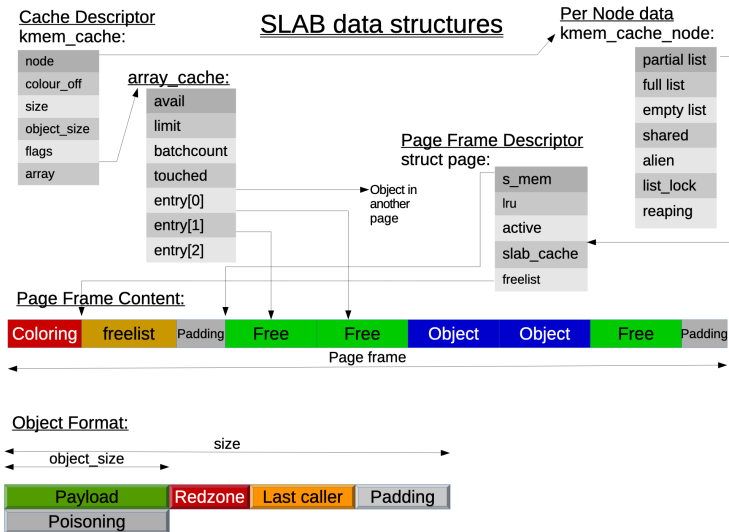
SLAB 分配器的结构



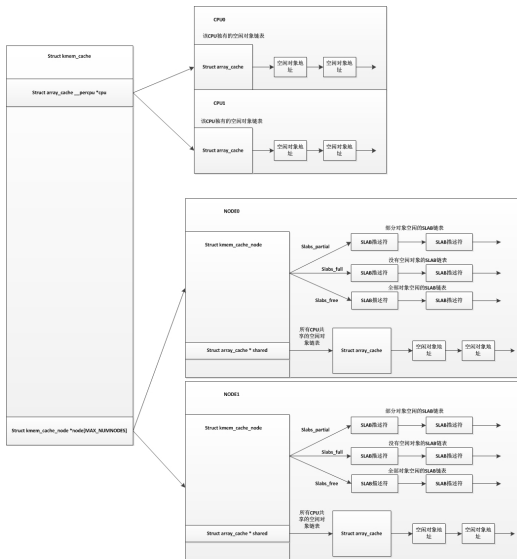
CPU 缓存着色与 SLAB



SLAB 的数据结构



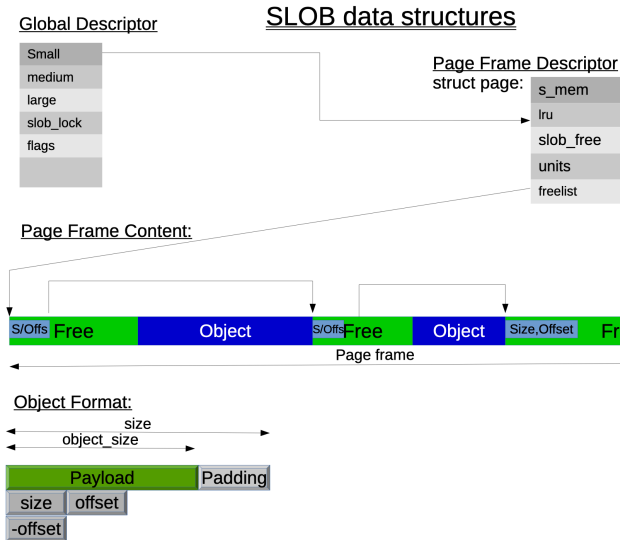
单个 SLAB 分配器结构



SLOB 分配器

- SLOB 分配器是针对嵌入式系统的 SLAB 简化版本
 - 没有本地 CPU 高速缓存和本地节点的概念
 - 只存在三个全局 partial free 链表
 - 链表按对象大小来划分

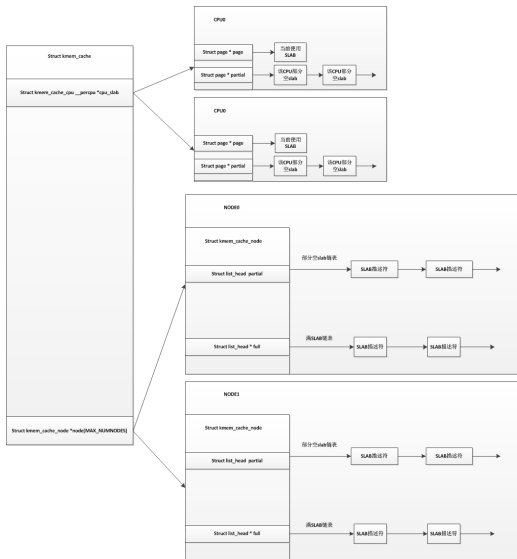
SLOB 分配器的数据结构



SLUB 分配器

- 目标
 - 简化设计理念
- 思路
 - 简化 SLAB 的结构：取消了大量的队列和相关开销
 - 一个 SLAB 是一组一个或多个页面，封装了固定大小的对象，内部没有元数据
 - 将元数据存储在页面相关的页结构
 - 没有单独的 Empty SLAB 队列

单个 SLUB 分配器结构



SLUB 分配器的数据结构

