L1实验报告

slow path: central list

大致实现

所有来自slab的内存请求或者大于32KB的内存请求会通过central list来分配内存, 中央链表只有一把大锁, 且地址按照64KB对齐.

对所有来自slab的内存请求, 直接返回一块大小为64KB的内存, 将中央链表的HEAD放在该块内存的首地址.

对所有大于32KB的内存请求, 会分配"请求内存大小 + 64KB"的内存, 以便64KB的地址对齐. 返回ptr指向请求分配的内存块, 多出来的64KB放在请求的内存块之前, 将中央链表的HEAD放在该64KB的首地址, 虽然有点浪费, 但是简化实现

fast path: slab

每个cpu有一把自己的锁, 和一组slab, slab的order从1到log(32KB), 用于分配小于32KB的内存, slab_alloc时不需要加锁, slab_free时需要加锁, 因为可能有多个cpu同时释放属于同一个cpu的内存.

slab同样有个HEAD, 放在central list的HEAD之后.

slab采用引用计数的方式, 当free_cnt = 0时, 请求一个新的slab; 当use_cnt = 0时, 释放该slab.