Linux SWAP原理

# Linux地址空间

## 物理地址空间（PA）

Linux物理地址空间表示的是一块内存区域实际存访的地址，地址从首地址0开始算起。

Linux物理地址空间的分配可以通过查看/proc/iomem进行查看。

## 内核线性地址空间（VA）

Kernel模块使用的地址空间，64位系统下通常只是PA+一个偏移实现（32位系统下可能会通过映射管理）。地址转换通过MMU自动转换。

下面是PA到VA转化的内核实现

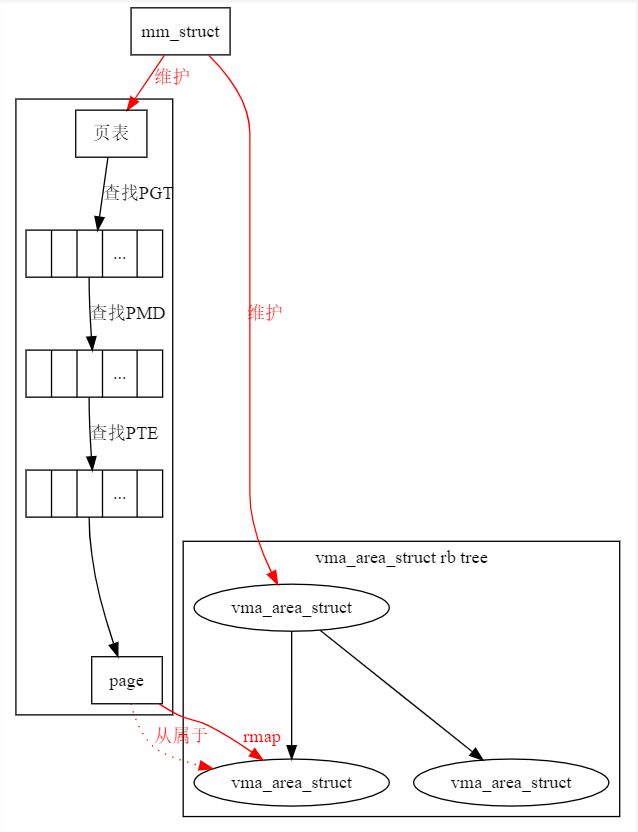
1. #define \_\_PAGE\_OFFSET           \_AC(0xffff880000000000, UL) //x86\_64
3. #define PAGE\_OFFSET     ((unsigned long)\_\_PAGE\_OFFSET)
4. //将物理地址转换为内核线性空间地址
5. #define \_\_va(x)         ((void \*)((unsigned long)(x)+PAGE\_OFFSET))

## 进程虚拟地址空间

**进程**虚拟地址空间通过mm\_struct管理.

每一个进程包含一个mm\_struct，每一个mm\_struct里面将进程地址空间分为不同的vma\_area\_struct，每一个vma\_area表示进程中申请的一块地址空间（可能是栈空间，代码地址，mmap生成的堆地址），每一个vma\_area维护一个页中断函数用来处理页中断，vma\_area\_struct通过红黑树组织。

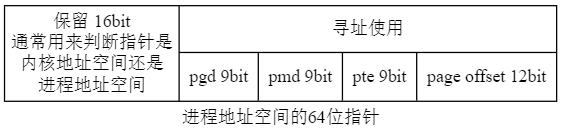
mm\_struct维护一个页表，在vma\_area\_struct触发页中断是建立映射(map)和反映射(rmap)。



# 页表映射

x86\_64操作系统当中，指针大小通常为64位。

## 指针结构

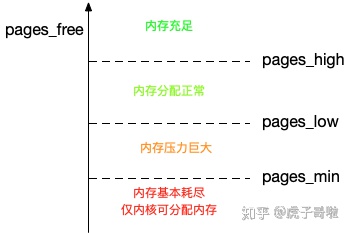


1. 为什么页表引索为9bit？因为每个页表索引节点为一页（4096bytes），每个页可以储存4096/8=512个指针，为2的9次方，所以每个引索是9bit。
2. 为什么page offset是12位？因为4096（页的默认大小）是2的12次方。
3. 为什么前面保留16位？首先48位已经可以储存256T数据，暂时够用。16位可以用来区分这个指针类型是内核地址空间还是进程地址空间。

# SWAP

Linux系统中swap操作是通过内核线程kswapd进行的。

## 阈值

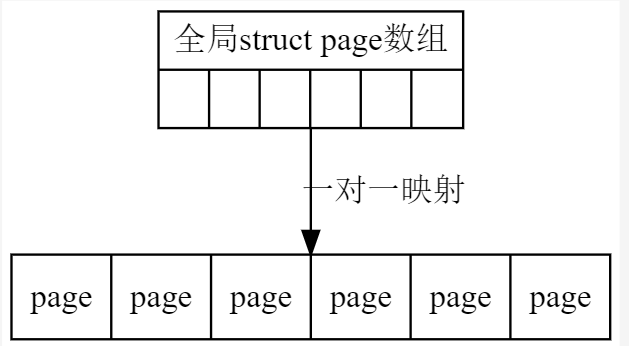


kswapd会定期扫描内存剩余。其中定义了如图三个宏来表示当前内存使用的吃紧程度。当剩余内存低于low时，kswapd开始回收内存（将部分内存放到硬盘上），直到内存剩余的量最终大于pages\_low为止。

当内存剩余空间小于min时，进程不可分配内存使用，只有内核才可以分配内存使用。

## 页数据维护

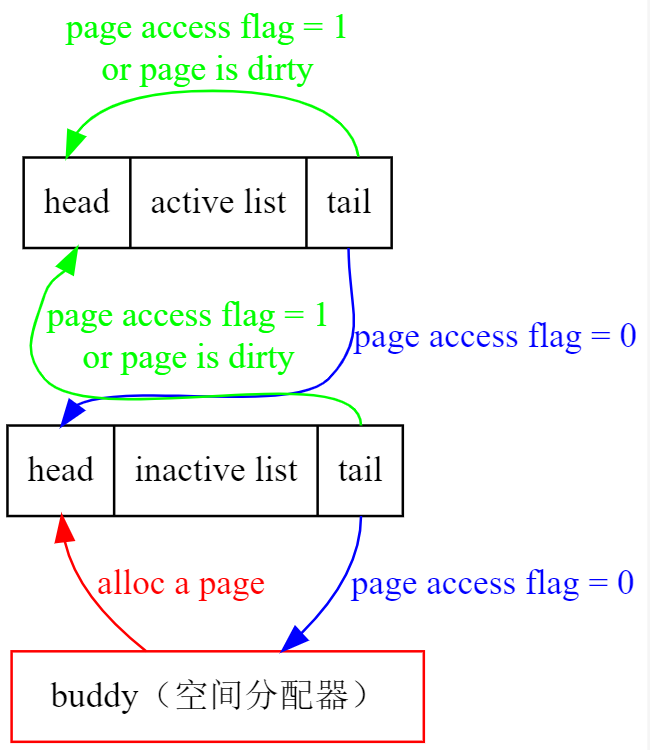
在Linux内核初始化阶段，将所有的物理地址按照页划分，为页分配一个page frame number（pfn = 物理地址/4096）。将所有页的元数据存放在一个全局指针的位置，通过pfn引索。。



该数组在Linux中通过struct page \*mem\_map实现。

## LRU链表

通过使用mem\_map中page的元数据中的LRU链表头，将所有page元数据放入LRU链表中。



在Linux中LRU操作包含两个LRU链表。分别是Active list(AL)和Inactive list(IL)。**这两个链存储的是已经分配的page元数据。**

1. 每当buddy分配器分配页面给用户时，将page加入IL头部。
2. 每次swap操作首先检测AL尾部节点对应元数据中的**PAGE\_ACCESS\_FLAG**。如果为0则放入IL，反之放入AL头部。

然后检测IL尾部节点，如果PAGE\_ACCESS\_FLAG为0，将节点回收。反之放入AL当中。

1. 为什么使用AL和IL实现LRU，不用传统的LRU list呢？
2. 首先，传统LRU list要求每次对page进行访问时候更新LRU链表，但是操作系统并不能做到记录每一次页面的访问（因为有TLB的存在）。
3. 其次，操作系统提供了PAGE\_ACCESS\_FLAG，会在页表查询到相应页面时进行设定。因为有TLB的存在，每次清除PAGE\_ACCESS\_FLAG都必须flush TLB。Linux通过使用IL结合AL，实现类似定时检测PAGE\_ACCESS\_FLAG的功能，尽量减少flush TLB次数。

## 页面回收

页面回收时，首先通过page元数据中的rmap遍历所有涉及的mm\_struct，清空对应的PTE项，flush TLB，即可。