

内核共用一个页面目录

- PGD( 页面目录 )
- PGD( 页面目录 )
- PGD( 页面目录 )
- ( 目录项 ) 内核虚拟

- 页表
- ( 页表项 ) 内核虚拟
- ( 页表项 ) 内核虚拟
- 页表
- 页表

To:swp\_entry\_t  
如果页面不在内存

- 页面
- 页内地址

当页面中的内容来自一个文件时，index 表示页面在文件中的序号，当页面的内容被换出到交换设备上，但还保留内容作为缓冲时，index 指明了页面的去向。

struct page \*mem\_map

- struct page [0]
- .list
- .lru
- .mapping
- .index;
- .next\_hash
- .count
- struct page [1]
- struct page []
- struct page []
- struct page []
- struct page []
- struct page []
- struct page []

表示 NUMA 中的一个存储节点

- struct pglist\_data
- .node\_mem\_map
- .node\_zones[0]
- .offset;
- .free\_area[MAX\_ORDER]
- .free\_list
- .map
- .inactive\_clean\_list
- \*zone\_pgdat
- .node\_zones [1]
- .node\_zonelists[ ]
- \* zones [.]
- .gfp\_mask
- .node\_zonelists[ ]
- \*node\_next

struct pglist\_data

一个 node\_zonelists 对应一种内存分配策略

- active\_list
- inactive\_dirty\_list

每个线程有自己单独的 pgd, 指针中是虚拟地址，在内核态下转成物理地址存放到 CR3 寄存器 ( 内核态下虚拟地址和物理地址存在线性对应关系 )  
asm volatile("movl %0,%%cr3": : "r" ( \_\_pa(next->pgd)) );

- struct task\_struct
- .mm

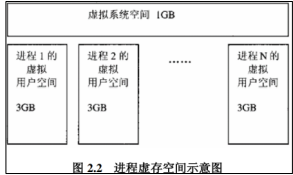
- struct mm\_struct
- .mmap
- .mmap\_avl
- .mmap\_cache
- .pgd
- .rss

- struct vm\_area\_struct
- .vm\_mm
- .vm\_start
- .vm\_end
- .vm\_next
- .vm\_ops

- struct vm\_operations\_struct
- (.\*open)()
- (.\*close)()
- (.\*nopage)()

- 虚拟存储空间
- 堆栈
- 内核空间 > 3G
- 堆栈
- 用户空间
- BSS
- 数据段
- 代码段

- 物理内存
- 0 物理地址



ZONE\_NORMAL  
ZONE\_DMA

- struct address\_space
- .clean\_pages
- .dirty\_pages
- .locked\_pages

- swapper\_space
- struct address\_space
- .clean\_pages
- .dirty\_pages
- .locked\_pages
- .a\_ops

管理所有可交换内存页面，所有换出的页面都在其中一个队列上

- swap\_aops
- address\_space\_operations

- swap\_info[MAX\_SWAPFILES]
- struct swap\_info\_struct
- .swap\_map
- .max
- .pages;
- .lowest\_bit
- .highest\_bit
- .cluster\_next

- swap\_list
- struct swap\_list\_t

- swap\_map[0]
- swap\_map[1]

每个数组成员对应一个盘上页面，swap\_map[0] 用于记录页面信息，其他成员表示页面引用计数

- swp\_entry\_t
- .offset // 下标
- .type // 下标

- 交换设备 1
- 盘上页面 1
- 盘上页面 2
- 交换设备 2
- 盘上页面
- 盘上页面 2