page

zone：将page划分为不同的区域，即zone

ZONE\_DMA、ZONE\_CMA、movable

页的分配：

alloc\_pages、alloc\_page、\_\_get\_free\_pages、\_\_get\_free\_page

页的释放：

free\_page、free\_pages

内核内存的分配和释放：

kmalloc、kfree

kmalloc：

rb = kmalloc(sizeof(\*rb) + len, GFP\_KERNEL);

if (rb == NULL)

return -ENOMEM;

kmalloc的参数：gfp\_mask标志

标志分为三类：行为修饰符、区修饰符、类型

行为修饰符：内核如何分配所需的内存，例如在中断服务程序中分配内存不能睡眠。

区修饰符：从哪个区域ZONE中来分配内存。

类型：行为修饰符和区修饰符的组合。

vmalloc：虚拟地址连续，物理地址不一定连续

kmalloc分配的物理地址一定连续

buddy：对页框的管理，如果有2^n个连续的页框空闲，就进行合并。

slab：buddy以页为单位进行管理，而程序实际使用通常以字节为单位，因此slab分配器专为小内存分配而生。

slab类似于一个链表，存储了所有的有关信息。

Linux进程在虚拟地址空间执行，每一个进程都拥有自己独立的地址空间。

32位: 0 ~ 2^32 - 1

64位: 0 ~ 2 ^64-1

使用虚拟内存的作用：

1. 资源的虚拟化：

在使用虚拟内存的系统中，进程不用关心到底有多少物理内存可用，或者有哪些物理内存已经被其他进程所使用。虚拟内存占用有效的物理内存，将其转化为无限的或至少是丰富的资源。

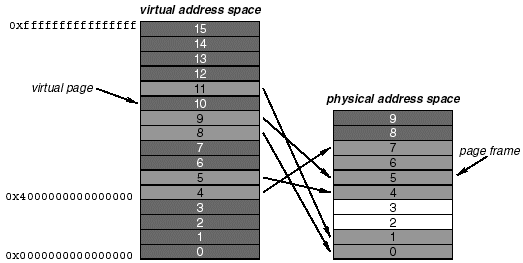
1. 信息的隔离：

由于每个进程都运行在自己独立的地址空间，因此一个进程无法读取另一个进程的私有数据。这大大增强了系统的安全性，它降低了一个进程偷取另一个进程私有数据的风险。

1. 错误隔离：

具有自己虚拟地址空间的进程不能覆盖彼此的内存。这大大降低了一个进程出错引发其他进程出错的风险，当一个进程出错后，错误仅仅局限在这个出错的进程内部，它不会导致整个系统挂掉。

1. Linux的虚拟内存系统以及相关概念，虚拟内存系统的软硬件组成



虚拟地址空间被划分成等大小的页，页大小通常为2的n次方，Linux的页大小通常为4，8，16或者64Kb。

虚拟地址 = 页号 + 页内偏移

物理地址空间通常也被划分成相同大小的页，每一页物理地址空间被称为页帧（page frame）。

1. Linux虚拟地址空间
2. Linux Page Tables
3. Linux对TLB的管理（Translation Lookaside Buffer，一个用来加速虚拟地址访问的硬件结构）
4. Linux的缺页异常处理机制
5. 内存一致性，以及Linux如何保证进程访问时能读到正确的值
6. Linux地址空间的切换（进程切换时的必要过程）
7. 总结