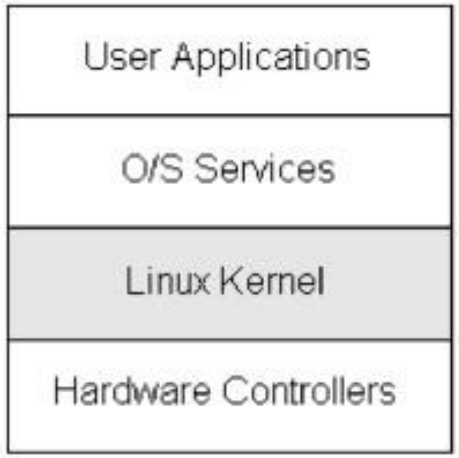
第013讲 不连续页分配器及页表



零声学院讲师：Vico老师

一、不连续页分配器

二、页表

一、不连续页分配器

1、系统接口

不连续页分配器所提供接口如下：

```
void *vmalloc(unsigned long size);
void vfree (const void * addr);

void *vmap(struct type **pages,unsigned int count,unsigned long flags,pgprot_t prot);
void vunmap(const void *addr);
```

内核还提供接口：

```
void *kvmalloc(size_t size,gfp_t flags);
void kvfree (const void * addr);
```

2、内核源码数据结构

- 每个虚拟内存区域对应一个vmap_area实例；
- 每个vmap_area实例关联着一个vm_struct实例。

vmalloc.h 2

include > linux > vmalloc.h > vm_struct > caller

```
43  struct vmap_area {
44      unsigned long va_start;
45      unsigned long va_end;
46      unsigned long flags;
47      struct rb_node rb_node;
48      struct list_head list;
49      struct llist_node purge_list;
50      struct vm_struct *vm;
51      struct rcu_head rcu_head;
52  };
```

include > linux > vmalloc.h > vmap_area > flags

```
32  struct vm_struct {
33      struct vm_struct    *next;
34      void                *addr;
35      unsigned long       size;
36      unsigned long       flags;
37      struct page         **pages;
38      unsigned int        nr_pages;
39      phys_addr_t         phys_addr;
40      const void          *caller;
41  };
```

3、技术原理

vmalloc虚拟地址空间的范围是（VMALLOC_START,VMALLOC_END），每种处理器架构都需要定义这两个宏，比如：ARM64架构定义的宏如下：

arch > arm64 > include > asm > C pgtable.h > ...

```
34
35  #define VMALLOC_START        (MODULES_END)
36  #define VMALLOC_END     (PAGE_OFFSET - PUD_SIZE - VMEMMAP_SIZE - SZ_64K)
37
```

二、页表

页表是一种特殊的数据结构，放在系统空间的页表区，存放逻辑页与物理页帧的对应关系。 每一个进程都拥有一个自己的页表，PCB表中有指针指向页表。

1、地址结构

逻辑地址、物理地址、逻辑地址空间、物理地址空间?

2、页表

页表用来把虚拟页映射到物理页，并且存放页的保护位，即访问权限。

3、ARM64处理器的页表

ARM64处理器把页表称为转换表，最多4级。ARM64处理器支持3种页长度，4KB、16KB和64KB。 页长度和虚拟地址的宽度决定了转换表的级数。

办学宗旨：一切只为渴望更优秀的你

办学愿景：让技术简单易懂