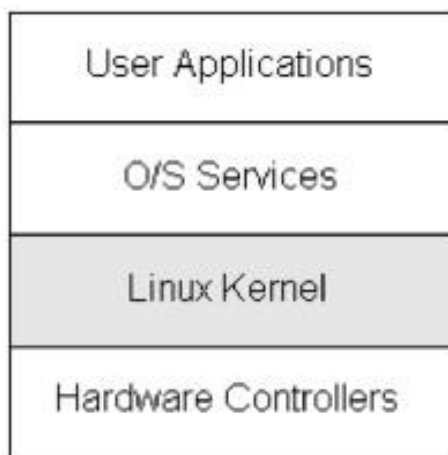




第0010讲 引导内存分配器原理



零声学院讲师: Vico老师



一、bootmem分配器

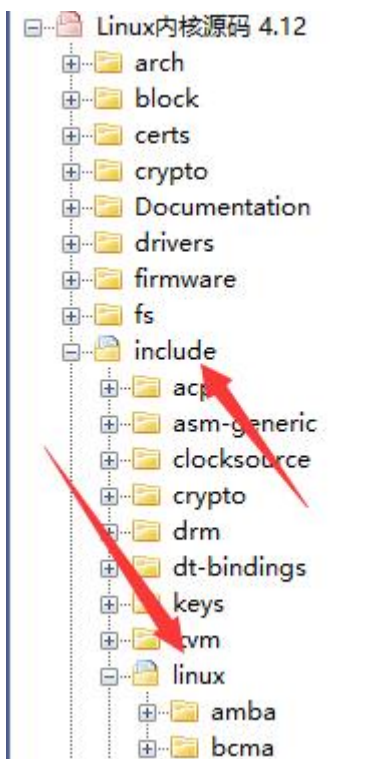
二、memblock分配器



一、bootmem分配器

在内核初始化的过程中需要分配内存，内核提供临时的引导内存分配器，在页分配器和块分配器初始化完成之后，把空闲的物理页交给页分配器管理，丢弃引导内存分配器。

bootmem分配器定义的数据结构，内核源码如下：

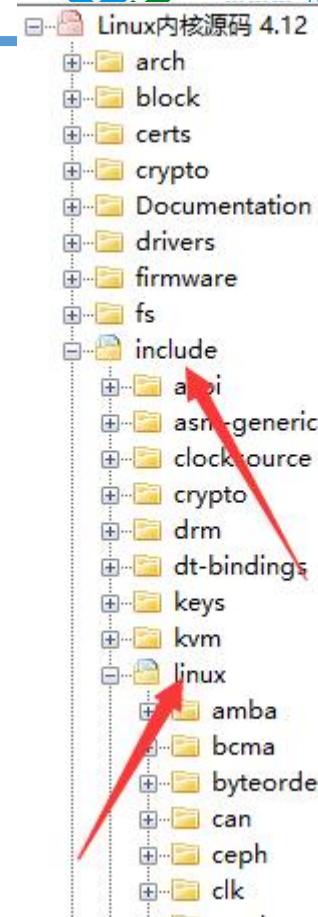


```
33 typedef struct bootmem_data {
34     unsigned long node_min_pfn;
35     unsigned long node_low_pfn;
36     void *node_bootmem_map;
37     unsigned long last_end_off;
38     unsigned long hint_idx;
39     struct list_head list;
40 } bootmem_data_t;
```



其中每个内存节点有一个bootmem_data实例：

```
621 struct bootmem_data;
622 typedef struct pglist_data {
623     struct zone node_zones[MAX_NR_ZONES]; // 内存区域数组
624     struct zonelist node_zonelists[MAX_ZONELISTS]; // 备用区域列表
625     int nr_zones; // 该节点包含的内存区域数量
626 #ifdef CONFIG_FLAT_NODE_MEM_MAP // 除了稀疏内存模型以外
627     struct page *node_mem_map; // 页描述符数组
628 #ifdef CONFIG_PAGE_EXTENSION
629     struct page_ext *node_page_ext; // 页的扩展属性
630 #endif
631 #endif
632 #ifndef CONFIG_NO_BOOTMEM
633     struct bootmem_data *bdata;
634 #endif
}
```



bootmem分配器提供分配内存/释放内存（alloc_bootmem/free_bootmem）。

ARM64架构的内核已不使用bootmem分配器，但其它处理器架构还在使用

bootmem分配器。

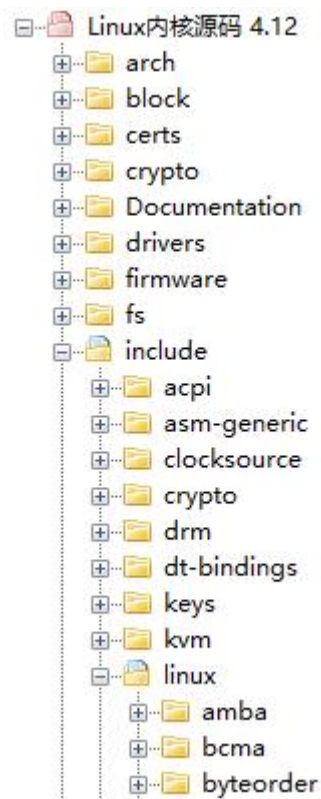


二、memblock分配器



1、数据结构

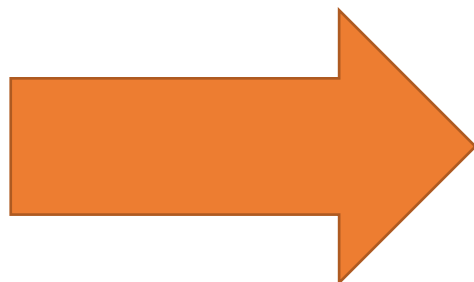
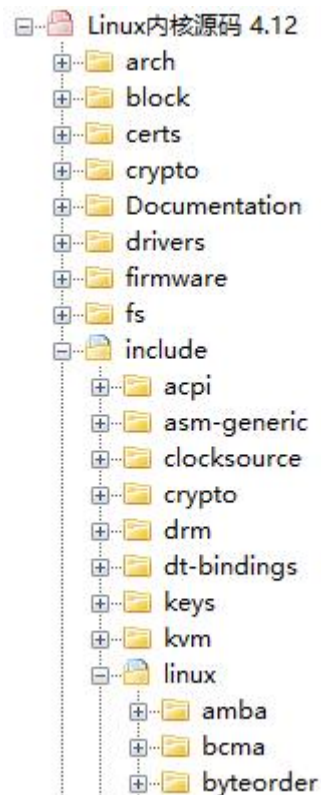
memblock分配器的数据结构如下:



```
48 struct memblock {
49     bool bottom_up; /* is bottom up direction? */
50     phys_addr_t current_limit;
51     struct memblock_type memory;
52     struct memblock_type reserved;
53 #ifdef CONFIG_HAVE_MEMBLOCK_PHYS_MAP
54     struct memblock_type physmem;
55 #endif
56 };
```



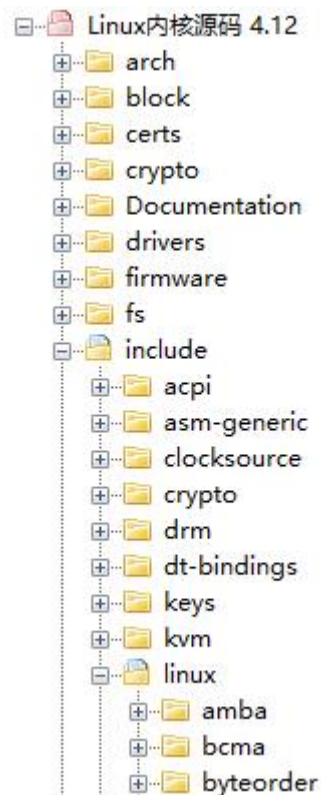

内存块类型的数据结构如下:



```
40 struct memblock_type {  
41     unsigned long cnt; /* number of regions */  
42     unsigned long max; /* size of the allocated array */  
43     phys_addr_t total_size; /* size of all regions */  
44     struct memblock_region *regions;  
45     char *name;  
46 };  
47
```

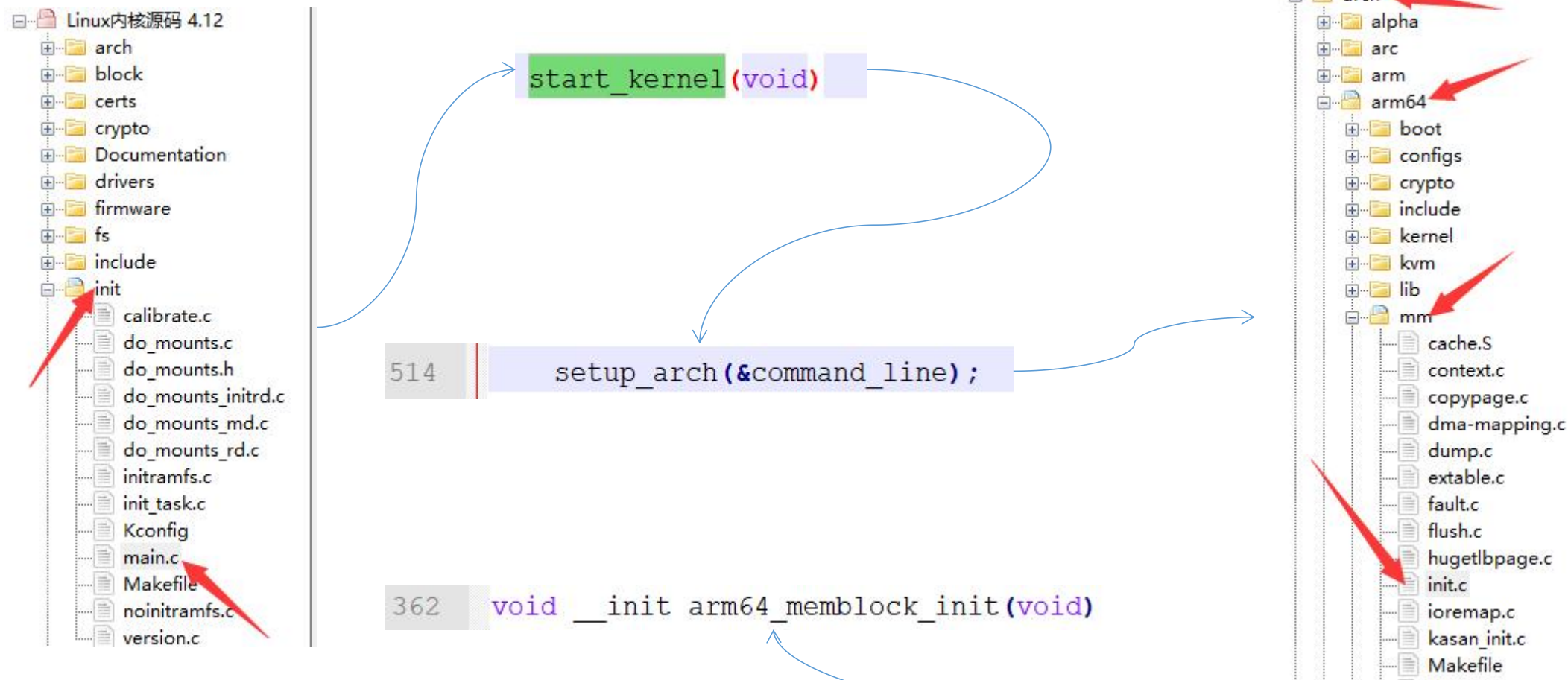


内存块区域的数据结构:



```
22
23  /* Definition of memblock flags. */
24  enum {
25      MEMBLOCK_NONE           = 0x0,  /* No special request */
26      MEMBLOCK_HOTPLUG       = 0x1,  /* hotpluggable region */
27      MEMBLOCK_MIRROR        = 0x2,  /* mirrored region */
28      MEMBLOCK_NOMAP         = 0x4,  /* don't add to kernel direct mapping */
29  };
30
31  struct memblock_region {
32      phys_addr_t base;
33      phys_addr_t size;
34      unsigned long flags;
35  #ifdef CONFIG_HAVE_MEMBLOCK_NODE_MAP
36      int nid;
37  #endif
38  };
```

2. ARM64内核初始化memblock分配器流程





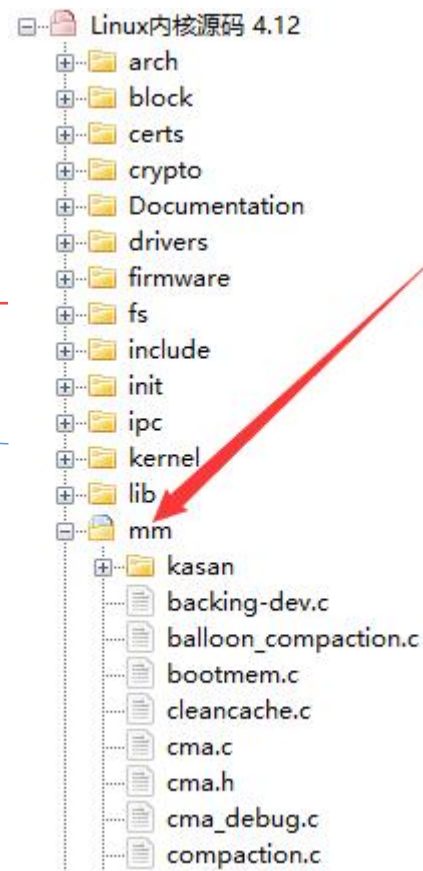
3、mемblock分配器编程接口

mемblock_add: 添加新的内存块区域到memblock.memory中;

mемblock_remove: 删除内存块区域;

mемblock_alloc: 分配内存;

mемblock_free: 释放内存。





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