

Linux认证辅导:Linux内核中的循环链表结构Linux认证考试

PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Linux_E8_AE

[_A4_E8_AF_c103_644857.htm](https://www.100test.com/kao_ti2020/644/2021_2022_Linux_E8_AE_A4_E8_AF_c103_644857.htm) 注：文章中引用的代码来源

于LXR，所分析的内核版本是v2.6.31。linux内核通过定义list_head以及对于list_head上的一组操作实现对不同类型的循环链表的同类操作，这种做法避免了对于不同数据类型的循环链表定义重复的操作函数，使代码得到了充分的使用，

是一种十分有效的编程方法。list_head的定义：

```
19 struct  
list_head { 20 struct list_head *next, *prev. 21 }.
```

接着我们来看任意一种数据结构的循环链表（如图1），链表的每个节点中加入了一个list_head类型的变量，节点的其他变量任意。（注意：每个指针所指向的位置不是节点数据的起始位置，而

是list_head类型变量的开始地址。）图1通过这样一种实现方式建立的链表，节点都是通过list_head类型的变量相连接的，

那么我们如何由list_head类型得指针得到中间某个节点类型的指针呢？我们来看这样一个操作：list_entry(p,t,m)，其中t是链表的节点类型，m是节点内list_head类型的变量名，p是指向该变量的指针，该操作用于从list_head指针得到指向链表节点的指针。

```
334 #define list_entry(ptr, type, member) \ 335  
container_of(ptr, type, member) 650 #define container_of(ptr, type,
```

```
member) ({ \ 651 const typeof( ((type *)0)-&.(TYPE
```

```
*)0)-&.MEMBER) /*计算出变量在结构中的偏移量（以字节为单位）*/
```

100Test 下载频道开通，各类考试题目直接下载。

详细请访问 www.100test.com