

第 0019 讲 11kzalloc&kcallolc 案例实战分析

一、kzalloc&kcallolc 函数及功能

1、kzalloc()函数

该函数是内核中动态内存分配之一，与 `kmalloc` 函数类似，区别在于它所分配的内存空间初始化为 0，它也是基于 slab 分配在物理上连续的实际内存。使用时可以通过指定标志位来指定你所分配内存的类型和属性。

`kzalloc()`函数原型如下：

```
include > linux > C slab.h > ...
670  /**
671   * kzalloc - allocate memory. The memory is set to zero.
672   * @size: how many bytes of memory are required.
673   * @flags: the type of memory to allocate (see kmalloc).
674   */
675  static inline void *kzalloc(size_t size, gfp_t flags)
676  {
677      return kmalloc(size, flags | __GFP_ZERO);
678  }
679
```

size: 指定要分配的内存字节个数；

flags: 分配标志。

返回值: 返回分配内存对象的引用，即内存对象的起始地址。

配套函数: 释放内存对象 `kzfree()`（防止内存泄漏）。

2、kcalloc()函数

该函数是内核中动态内存分配之一，与 `kzalloc` 函数类似，但有一个额外的参数是用来指定所分配内存中每个元素的大小，同时 `kcalloc` 也会将所有分配的内存空间初始化为 0。

`kcalloc()`函数原型如下：

```
include > linux > C slab.h > kcalloc(size_t, size_t, gfp_t)
608 /**
609  * kcalloc - allocate memory for an array. The memory is set to zero.
610  * @n: number of elements.
611  * @size: element size.
612  * @flags: the type of memory to allocate (see kmalloc).
613  */
614 static inline void *kcalloc(size_t n, size_t size, gfp_t flags)
615 {
616     return kmalloc_array(n, size, flags | __GFP_ZERO);
617 }
618
```

n: 数组中元素的个数；

size: 每个元素对应的内存对象大小；

flags: 分配标志。

返回值: 返回所分配的内存对象数组的引用。

配套函数: 释放内存对象 `kfree()`（防止内存泄漏）。

二、实战案例分析

1:kzalloc()函数实例

```
root@ubuntu: /home/vico/Desktop/kzallocdemo
File Edit View Search Terminal Help
root@ubuntu:/home/vico/Desktop/kzallocdemo# ls
kzallocdemo.c      kzallocdemo.mod.c  Makefile
kzallocdemo.ko     kzallocdemo.mod.o  modules.order
kzallocdemo.mod    kzallocdemo.o      Module.symvers
root@ubuntu:/home/vico/Desktop/kzallocdemo# insmod kzallocdemo.ko
root@ubuntu:/home/vico/Desktop/kzallocdemo# dmesg -c
[ 6484.983387] Prompt:kzalloc() function allocation successfully, address = 0xf
fff9ccf9b648000
[ 6484.983388] Prompt:The content of memory_pointer+4 is : 0
root@ubuntu:/home/vico/Desktop/kzallocdemo# rmmod kzallocdemo.ko
root@ubuntu:/home/vico/Desktop/kzallocdemo# dmesg -c
[ 6517.529203] Prompt:kzfree() function successfully free memory.
[ 6517.529204] Prompt:Normal exit of kernel module.
root@ubuntu:/home/vico/Desktop/kzallocdemo#
```

2: kcalloc()函数实例

```
root@ubuntu: /home/vico/Desktop/kcallocdemo
File Edit View Search Terminal Help
root@ubuntu:/home/vico/Desktop/kcallocdemo# ls
kcallocdemo.c  Makefile
root@ubuntu:/home/vico/Desktop/kcallocdemo# make
make -C /usr/src/linux-headers-5.4.0-150-generic M=/home/vico/Desktop/kcallo
cdemo modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-150-generic'
CC [M] /home/vico/Desktop/kcallocdemo/kcallocdemo.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/vico/Desktop/kcallocdemo/kcallocdemo.mod.o
LD [M] /home/vico/Desktop/kcallocdemo/kcallocdemo.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-150-generic'
root@ubuntu:/home/vico/Desktop/kcallocdemo# insmod kcallocdemo.ko
root@ubuntu:/home/vico/Desktop/kcallocdemo# dmesg -c
[ 7529.839558] Prompt:kcalloc() allocation address = 0xffff9ccf9cbce000
[ 7529.839560] Prompt:ksize() calculates the size of memory space = 8192
[ 7529.839561] Prompt:the content of memory_pointer+10 = 0
root@ubuntu:/home/vico/Desktop/kcallocdemo# rmmod kcallocdemo.ko
root@ubuntu:/home/vico/Desktop/kcallocdemo# dmesg -c
[ 7553.359873] Prompt:kfree() was successfully released.
[ 7553.359874] Prompt:Normal exit of kernel module.
root@ubuntu:/home/vico/Desktop/kcallocdemo#
```