kmalloc vs vmalloc

2016-11-10 (http://jake.dothome.co.kr/kmalloc-vs-vmalloc/) Moon Young-il (http://jake.dothome.co.kr/author/admin/) Linux Kernel (http://jake.dothome.co.kr/category/linux/)

kmalloc vs vmalloc comparison

When allocating memory in the kernel, there are two APIs to consider between the required memory size and performance, and they should be used separately according to their characteristics.

- The allocation sizes are based on the most commonly used slub in the latest one, which is different units for slab, slub, and slob.

Features of kmalloc

Contiguous physical memory is allocated by mapping it to a contiguous virtual space.

- If the required memory is less than the maximum size of the kmalloc cache, the slub object is allocated from the kmalloc cache.
  - e.g. 4K pages: kmalloc cache (slub) max size=8K (2 pages)
- If the required memory is greater than the kmalloc maximum size, it should be allocated using the buddy system.
- Contiguous physical memory is allocated by mapping it to a contiguous virtual address space. (Available on DMA devices)
- It already uses pre-1:1 mapped lowmem (ZONE_DMA and ZONE_NORMAL) space, so it's faster than vmalloc().
- The disadvantage is that it is necessary to allocate a physically contiguous space, which makes it difficult to manage fragmentation in terms of page management
- If you use a GFP_ATOMIC flag, it doesn't slip, so it can also be used in interrupt handlers.

Features of vmalloc

Multiple fragmented and non-contiguous pages are collected and mapped to a contiguous virtual memory space and allocated.

- Unlike kmalloc(), it collects and manages multiple fragmented contiguous physical memory and maps them to separate spaces (vmalloc address space for kernel and user space address for userland), which increases the management factor and requires a TLB flush of each CPU, which makes it difficult to use when there are many CPU cores.
  - The kernel provides contiguous virtual address memory via the VMALLOC address space, which varies depending on the architecture.

        - arm64: About half of the kernel mapping space (location varies for different kernel build options)
        - arm: 240M (0xf000_0000 ~ 0xff00_0000)
    - For mapping, use the vmap() mapping function that uses the vmalloc address space space.
- If you need a large amount of kernel memory and expect pressure from the lack of lowmem, and you can afford to be a little slower, you may want to use vmalloc() to avoid fragmenting the page.
- In order to map non-contiguous blocks of physical addresses to contiguous virtual addresses, the existing kernel managed by adding each vma_area (a block of mapping ledgers) to the list. Although the kernel did not use a large number of vmap_area, it was changed in kernel 2.6 to manage using the red-black tree to prevent performance degradation by searching hundreds or thousands of vmp_area in large applications.
- It should not be used in interrupt handlers as it can slip.

For this reason, most kernel code uses kmalloc() more and vmalloc() for the following purposes:

- Data Structure for Swap Area
- Module Space Allocation
- Some device drivers, etc.

kzalloc() vs vzalloc()

- The kzalloc() function initializes the memory allocated by the kmalloc() function to zero.
- The vzalloc() function initializes the memory allocated by the vmalloc() function to zero.

consultation

- Slab Memory Allocator -1- (Structure) (http://jake.dothome.co.kr/slub/) | Qc
- Kmalloc vs Vmalloc (http://jake.dothome.co.kr/kmalloc-vs-vmalloc) | Sentence C – Current post
- Kmalloc (http://jake.dothome.co.kr/kmalloc) | Qc
- Vmalloc (http://jake.dothome.co.kr/vmalloc) | Qc
- Vmap((http://jake.dothome.co.kr/vmap)) | Qc
- GFP Flags (http://jake.dothome.co.kr/gfp) | Qc

LEAVE A COMMENT

Your email will not be published. Required fields are marked with *

Comments

name *

email *

Website

WRITE A COMMENT

‹ GFP Flags (http://jake.dothome.co.kr/gfp-flag/)

per-cpu -3-(dynamic allocation) › (http://jake.dothome.co.kr/per-cpu-dynamic/)