
Subject: Re: [PATCH v2 06/13] slab: Add kmem_cache_gfp_flags() helper function.
Posted by [Glauber Costa](#) on Sun, 11 Mar 2012 10:53:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 03/10/2012 12:39 AM, Suleiman Souhlal wrote:

```
> This function returns the gfp flags that are always applied to
> allocations of a kmem_cache.
>
> Signed-off-by: Suleiman Souhlal<suleiman@google.com>
> ---
> include/linux/slab_def.h | 6 ++++++
> include/linux/slob_def.h | 6 ++++++
> include/linux/slub_def.h | 6 ++++++
> 3 files changed, 18 insertions(+), 0 deletions(-)
>
> diff --git a/include/linux/slab_def.h b/include/linux/slab_def.h
> index fbd1117..25f9a6a 100644
> --- a/include/linux/slab_def.h
> +++ b/include/linux/slab_def.h
> @@ -159,6 +159,12 @@ found:
>  return __kmalloc(size, flags);
>  }
>
> +static inline gfp_t
> +kmem_cache_gfp_flags(struct kmem_cache *cachep)
> +{
> + return cachep->gfpflags;
> +}
> +
> #ifdef CONFIG_NUMA
> extern void *__kmalloc_node(size_t size, gfp_t flags, int node);
> extern void *kmem_cache_alloc_node(struct kmem_cache *, gfp_t flags, int node);
> diff --git a/include/linux/slob_def.h b/include/linux/slob_def.h
> index 0ec00b3..3fa527d 100644
> --- a/include/linux/slob_def.h
> +++ b/include/linux/slob_def.h
> @@ -34,4 +34,10 @@ static __always_inline void *__kmalloc(size_t size, gfp_t flags)
>  return kmalloc(size, flags);
>  }
>
> +static inline gfp_t
> +kmem_cache_gfp_flags(struct kmem_cache *cachep)
> +{
> + return 0;
> +}
> +
> #endif /* __LINUX_SLOB_DEF_H */
> diff --git a/include/linux/slub_def.h b/include/linux/slub_def.h
```

```
> index a32bcfd..5911d81 100644
> --- a/include/linux/slub_def.h
> +++ b/include/linux/slub_def.h
> @@ -313,4 +313,10 @@ static __always_inline void *kmalloc_node(size_t size, gfp_t flags, int
node)
> }
> #endif
>
> +static inline gfp_t
> +kmem_cache_gfp_flags(struct kmem_cache *cachep)
> +{
> + return cachep->allocflags;
> +}
> +
> +
>
> #endif /* _LINUX_SLUB_DEF_H */
```

Why is this needed? Can't the caller just call
mem_cgroup_get_kmem_cache(cachep, flags | cachep->allocflags) ?