

L'allocateur slab

Xavier Deguillard
—
xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

L'allocateur slab

Xavier Deguillard – xavier@lse.epita.fr

LSE

26 octobre 2011

1 Introduction

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

Un allocateur doit être :

- rapide,
- avec le moins de fragmentation possible

Dans le cas du kernel, la fragmentation est un élément prépondérant.

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

Fonctionnement d'un allocateur généraliste

2 Object-caching

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

La durée de vie d'un objet peut se résumer à :

```
obj = alloc ();  
init (obj );  
use (obj );  
destroy (obj );  
free (obj );
```

Si la fonction alloc utilise un pool d'objet du même type, la phase d'initialisation et de destruction peut être évitée, puisque les objets du pool seront déjà correctement initialisés.

Plutôt que de faire toutes les allocations dans la même zone de mémoire (le tas), l'allocateur slab va construire un cache par type d'objet :

```
struct kmem_cache* kmem_cache_create(  
    char* name,  
    size_t size,  
    int align,  
    void (*constructor)(void*, size_t),  
    void (*destructor)(void*, size_t));
```

3 Slab

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

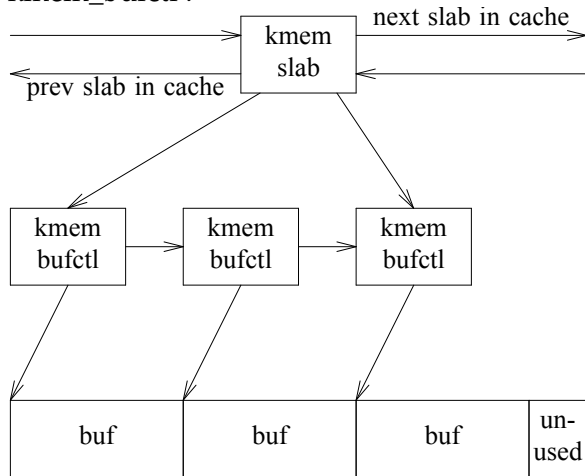
Slab

Allocation

Libération

Pour aller plus loin

Chaque cache est découpé slab, un slab étant une ou plusieurs pages virtuelles contiguës. Chaque slab maintient une liste de bloc libre représenté par un `kmem_bufctl` :



L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

4 Allocation

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

```
void* kmem_cache_alloc(struct kmem_cache* cache,
                      int flags);
```

- ① Trouver un slab avec un bloc libre,
- ② Prendre le premier kmem_bufctl,
- ③ Renvoyer le bloc pointé par le kmem_bufctl.

Le bloc retourné à l'utilisateur est déjà initialisé.

5 Libération

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

```
void kmem_cache_free(struct kmem_cache* cache,
                    void* buf);
```

- 1 Trouver le `kmem_bufctl` associé au bloc,
- 2 Placer le `kmem_bufctl` dans la liste des blocs libre.

6 Pour aller plus loin

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

L'allocateur slab

Xavier Deguillard

xavier@lse.epita.fr

Introduction

Object-caching

Slab

Allocation

Libération

Pour aller plus loin

Papier original

The Slab Allocator : An Object-Caching Kernel Memory Allocator, Jeff Bonwick