[SYS1] Systèmes d'exploitation (3)

Par Rhaeven, relecture par Find3r

Mapping mémoire

void *mmap(void *addr, size_t sz, int prot, int flag, int fd, off_t offset); : Faire des mapping mémoire

- addr = NULL : Le kernel choisit lui-même ou conçoit le mapping (le plus portable)
- size : Taille quelconque réalignée au multiple de 4096 (Taille de page par défaut du systeme) suivant
- prot : RD | WR
- offset : Doit être aligné. Écrit le fichier fd à partir de l'offset dans la page
- flag : MAP_PRIVATE (copie privée) | MAP_SHARED (changements propagés sur le file system) | MAP_ANONYMOUS (non POSIX, alloue de la mémoire et non un fichier)
- fd : -1 avec MAP_ANONYMOUS
- Valeur de retour : MAP_FAILED ((void*)-1)

> /dev/zero : En écriture n'écrit rien, en lecture lit les n bytes demandés (tous à 0)

munmap(addr, sz);
mprotect(addr, sz, prot); : Change les permissions sur les mappings

void *mremap(void *addr, size_t old_size, size_t new_size, int flag, ... /*void *new_addr*/); : Étendre ou réduire un mapping existant

> Garde la mémoire physique, crée une nouvelle zone d'adresse virtuelle qui pointe sur l'ancien bloc de m"moire physique

On veut pouvoir allouer de la mémoire dans la page (*allocateur à grains fins*)

Algorithmes d'allocation

First fit

- On alloue un bloc (avec des métadonnées associées : premier bloc alloué de la page, pointeur vers page suivante, taille de la page)
- Dans ce bloc, on alloue d'autres blocs (avec metadonnées associées : bloc libre?, taille du bloc, bloc suivant, bloc précédent), doublement chaînes entre eux

> Problème : Recherche de place libre et suppression : compléxite lineaire

> Problème : Grosses métadonnées

Solution : On alloue plusieurs pages, contenant chacune des allocations de taille fixe identique (**Buckets**)

Page begin

Retrouver le début de la page quand on a un pointeur random sur un bloc random dans cette page :

- Réaligner le pointeur sur la taille de la page (on redescend au multiple de 4096 (PAGE_SIZE) inf"rieur)

> Cast en uintptr_t
> addr & 0xFFFFF000

Métadonnées

On stocke les métadonnées à l'exterieur, qui pointe sur notre bloc de taille 4K, et sur la prochaine métadonnée

> Pas de temps constant sur la suppression

-> On utilise une hashmap