

NOM

mmap, munmap - Établir / supprimer une projection en mémoire (map / unmap) des fichiers ou des périphériques.

SYNOPSIS

```
#include <unistd.h>
#include <sys/mman.h>

#ifdef _POSIX_MAPPED_FILES

void * mmap(void *start, size_t length, int prot , int flags, int fd,
off_t offset);

int munmap(void *start, size_t length);

#endif
```

DESCRIPTION

La fonction mmap demande la projection en mémoire de length octets commençant à la position offset depuis un fichier (ou un autre objet) indiqué par le descripteur fd, de préférence à l'adresse pointée par start. Cette adresse n'est qu'une préférence, généralement 0. La véritable adresse où l'objet est projeté est renvoyée par la fonction mmap, et n'est jamais nul.

L'argument prot indique la protection que l'on désire pour cette zone de mémoire, et ne doit pas entrer en conflit avec le mode d'ouverture du fichier. Il s'agit soit de PROT_NONE (le contenu de la mémoire est inaccessible) soit d'un OU binaire entre les constantes suivantes :

PROT_EXEC

On peut exécuter du code dans la zone mémoire.

PROT_READ

On peut lire le contenu de la zone mémoire

PROT_WRITE

On peut écrire dans la zone mémoire.

Le paramètre flags indique le type de fichier projeté, les options de projection, et si les modifications faites sur la portion projetée sont privées ou doivent être partagées avec les autres références. Les options sont :

MAP_FIXED

N'utiliser que l'adresse indiquée. Si c'est impossible, mmap échouera. Si MAP_FIXED est spécifié start doit être un multiple de la longueur de page. Il est déconseillé d'utiliser cette option.

MAP_SHARED

Partager la projection avec tout autre processus utilisant l'objet. L'écriture dans la zone est équivalente à une écriture dans le fichier. En revanche ce dernier n'est pas nécessairement mis à jour tant qu'on n'a pas appelé msync(2) ou munmap(2).

MAP_PRIVATE

Créer une projection privée, utilisant la méthode de copie à l'écriture. L'écriture dans la zone ne modifie pas le fichier. Il n'est pas précisé si les changements effectués dans le fichier après l'appel mmap seront visibles.

Vous devez indiquer soit MAP_SHARED, soit MAP_PRIVATE.

Les trois attributs ci-dessus sont décrits dans POSIX.1b (anciennement POSIX.4) et SUSv2. Linux propose également des attributs non standards :

MAP_DENYWRITE

Cet attribut est ignoré. (Autrefois, une tentative d'écriture dans le fichier sous-jacent échouait avec l'erreur ETXTBUSY. Mais ceci permettait des attaques de déni de service).

MAP_EXECUTABLE

Cet attribut est ignoré.

MAP_NORESERVE

(Utilisé conjointement à MAP_PRIVATE). Ne pas réserver d'espace de swap pour les pages de cette projection. Une telle réservation garantit que l'on puisse modifier les zones soumises à une copie-en-écriture. Sans réservation, on peut recevoir un signal SIGSEGV durant une écriture, s'il n'y a plus de place disponible.

MAP_LOCKED

Les pages projetées ne seront pas transférées dans le swap.

MAP_GROWSDOWN

Utilisé pour les piles. Indique au système de gestion de la mémoire virtuelle que la projection doit s'étendre en croissant vers le bas de la mémoire.

MAP_ANONYMOUS

La projection n'est supportée par aucun fichier. Les arguments fd et offset sont ignorés. Cet attribut, utilisé en conjonction de MAP_SHARE est implémenté depuis Linux 2.4.

MAP_ANON

Alias de MAP_ANONYMOUS. Déconseillé.

MAP_FILE

Attribut pour compatibilité. Ignoré.

Certains systèmes utilisent les attributs supplémentaires MAP_AUTOGROW, MAP_AUTORESRV, MAP_COPY, and MAP_LOCAL.

fd doit être un descripteur de fichier valide, sauf si on utilise MAP_ANONYMOUS, auquel cas cet argument est ignoré.

L'argument offset doit normalement être un multiple de la taille de page renvoyée par l'appel getpagesize(2).

La mémoire obtenue par mmap est préservée lors d'un fork(2), avec les mêmes attributs.

La projection doit avoir une taille multiple de celle des pages. Pour un fichier dont la longueur n'est pas un multiple de la taille de page, la mémoire restante est remplie de zéros lors de la projection, et les écritures dans cette zone n'affectent pas le fichier. Les effets de la modification de la taille du fichier sous-jacent sur les pages correspondant aux zones ajoutées ou supprimées ne sont pas précisés. L'appel-système munmap détruit la projection dans la zone de mémoire spécifiée, et s'arrange pour que toute référence ultérieure à cette zone mémoire déclenche une erreur d'adressage. La projection est aussi automatiquement détruite lorsque le processus se termine. À l'inverse, la fermeture du descripteur de fichier ne supprime pas la projection.

L'adresse start doit être un multiple de la taille de page. Tous les pages contenant une partie de l'intervalle indiquées sont libérées, et tout accès ultérieur déclenche SIGSEGV. Aucune erreur n'est détectée si l'intervalle indiqué ne contient pas de page projetée.

Pour les projections supportées par un fichier, le champ st_atime du fichier peut être mis à jour à tout moment entre l'appel mmap() et le munmap() correspondant. Le premier accès dans la page projetée mettra le champ à jour si ce n'a pas été déjà fait.

Les champs st_ctime et st_mtime pour un fichier projeté avec PROT_WRITE et MAP_SHARED seront mis à jour après une écriture dans la région projetée, et avant l'éventuel msync() suivant avec attribut MS_SYNC ou MS_ASYNC.

VALEUR RENVOYÉE

mmap renvoie un pointeur sur la zone de mémoire, s'il réussit. En cas d'échec il retourne MAP_FAILED (-1) et errno contient le code d'erreur.

munmap renvoie 0 s'il réussit. En cas d'échec -1 est renvoyé et errno contient le code d'erreur (probablement EINVAL).

ERREURS

EBADF fd n'est pas un descripteur de fichier valide (et MAP_ANONYMOUS n'était pas précisé).

EACCES Le descripteur ne correspond pas à un fichier normal, ou on demande une projection privée MAP_PRIVATE mais fd n'est pas

ouvert en lecture, ou on demande une projection partagée MAP_SHARED avec protection PROT_WRITE, mais fd n'est pas ouvert en lecture et écriture (O_RDWR). Ou encore PROT_WRITE est demandé, mais le fichier est ouvert en ajout seulement.

EINVAL start ou length ou offset sont invalides. (par exemple : zone trop grande, ou non alignée sur une frontière de page).

ETXTBSY

MAP_DENYWRITE a été réclamé mais fd est ouvert en écriture

EAGAIN Le fichier est verrouillé, ou trop de pages ont été verrouillées en mémoire.

ENOMEM pas assez de mémoire, ou le nombre maximal de projection par processus a été dépassé.

ENODEV Le système de fichiers sous-jacent ne supporte pas la projection en mémoire.

L'accès à une zone de projection peut déclencher les signaux suivants :

SIGSEGV

Tentative d'écriture dans une zone en lecture seule

SIGBUS Tentative d'accès à une portion de la zone qui ne correspond pas au fichier (par exemple après la fin du fichier, y compris lorsqu'un autre fichier l'a tronqué).

CONFORMITÉ

SVr4, POSIX.1b (anciennement POSIX.4), BSD 4.4, SUSv2. SVr4 documente les codes d'erreur supplémentaires ENXIO et ENODEV. SUSv2 documente les codes d'erreur supplémentaires EMFILE et EOVERFLOW.

VOIR AUSSI

getpagesize(2), mremap(2), msync(2), shm_open(2), B.O. Gallmeister, POSIX.4, O'Reilly, pp. 119-124 et 365-369.

TRADUCTION

Christophe Blaess, 1997.