NOM

mprotect - Contrôler les autorisations d'accès à une partie de la mémoire.

SYNOPSIS

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

int mprotect(const void *addr, size_t *len, int prot);
```

DESCRIPTION

mprotect contrôle la manière d'accéder à une portion de la mémoire. Si un accès interdit se produit, le programme reçoit un signal SIGSEGV.

prot est un OU binaire (|) entre les valeurs suivantes

PROT_NONE
On ne peut pas accéder du tout à la zone de mémoire.

PROT_READ
On peut lire la zone de mémoire.

PROT_WRITE
On peut écrire dans la zone de mémoire.

PROT_EXEC
La zone de mémoire peut contenir du code exécutable.

La nouvelle protection remplace toute autre protection précédente. Par exemple, si la mémoire a été marquée précédemment PROT_READ, et si l'on appelle mprotect avec PROT_WRITE, la zone concernée ne sera plus lisible.

VALEUR RENVOYÉE

mprotect renvoie 0 si il réussit, ou -1 s'il échoue, auquel cas errno contient le code d'erreur.

ERREURS

EINVAL addr n'est pas un pointeur valide, ou ce n'est pas un multiple de PAGESIZE.

EFAULT La mémoire n'est pas accessible.

EACCES L'accès spécifié n'est pas possible sur ce type de mémoire. Ceci se produit par exemple si vous utilisez mmap(2) pour représenter un fichier en lecture-seule en mémoire, et si vous demandez de marquer cette zone avec PROT_WRITE.

ENOMEM Pas assez de mémoire pour le noyau

EXEMPLE D'UTILISATION

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/mman.h>

#include <limits.h> /* pour avoir PAGESIZE */

#ifndef PAGESIZE
  #define PAGESIZE 4096
#endif
```

```
int
main(void)
{
    char *p;
    char c;

    /* Allocation d'un buffer, protection
       par défaut PROT_READ|PROT_WRITE. */
    p = malloc(1024 + PAGESIZE - 1);
    if (!p) {
        perror("Impossible d'allouer suffisamment de mémoire");
        exit(errno);
    }

    /*
     * Aligner p sur un multiple de PAGESIZE (que l'on suppose être
     * une puissance de 2)
     */
    p = (char *) (((int) p + PAGESIZE-1) & ~(PAGESIZE-1));

    c = p[666];         /* lecture ok */
    p[666] = 42;        /* ecriture ok */

    /* Buffer marqué en lecture-seule */
    if (mprotect(p, 1024, PROT_READ)) {
        perror("Impossible d'utiliser mprotect");
        exit(errno);
    }

    c = p[666];         /* lecture ok */
    p[666] = 42;        /* écriture, fin du programme avec SIGSEGV */

    exit(0);
}
```

CONFORMITÉ

SVr4, POSIX.1b (anciennement POSIX.4). SVr4 définit un code d'erreur EAGAIN supplémentaire. Les conditions d'erreur SVr4 ne correspondent pas tout à fait à celles de Linux. POSIX.1B précise que mprotect ne peut être utilisé que sur des zones de mémoire obtenues avec mmap(2).

VOIR AUSSI

mmap(2)

TRADUCTION

Christophe Blaess, 1997.