mmap

#include <sys/mman.h>

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

- Réserve un segment de mémoire virtuelle, avec des droits d'accès donnés
  - **Partagé** entre différents processus ou **privé**
  - **couplé** à un fichier (contenu du fichier projeté en mémoire) ou **non**
- Retourne l'adresse de base de ce segment
- Si addr = NULL ➔ adresse choisie par le noyau
- length : taille du segment demandé
- prot : droits d'accès = PROT_NONE, PROT_EXEC, PROT_READ, PROT_WRITE
- flags :
  - MAP_SHARED : projection partagée. Les modifications de la projection sont visibles par les autres processus qui projettent le fichier, et sont appliquées à ce fichier
  - MAP_PRIVATE : projection privée. Les modifications de la projection ne sont pas visibles par les autres processus qui projettent le fichier, et ne sont pas appliquées à ce fichier
  - MAP_ANONYMOUS : Le segment n'est pas couplé à un fichier. Son contenu est initialisé à zéro. Les arguments *fd* et *offset* sont ignorés.
- fd : descripteur du fichier si couplage avec un fichier (non MAP_ANONYMOUS). Dans ce cas le fichier doit être ouvert au minimum en lecture
- offset : position à partir de laquelle les données du fichier sont projetées en mémoire

Exemples

1- **Un segment** de 4096 octets, en **lecture/écriture**, anonyme (sans ficher), et partagé

```
char* base = mmap (NULL, 4096, PROT_WRITE | PROT_READ, MAP_SHARED | MAP_ANONYMOUS, -1, 0);

if (base == MAP_FAILED) { printf ("mmap failed \n"); exit(1); }
```

2- **couplage privé** d'un fichier en lecture

```
struct stat statbuf;     //caractéristiques du fichier
if ((file_in = open (fichier, O_RDONLY)) == -1) { printf ("erreur …\n"); exit(1); }
if (fstat (file_in, &statbuf) < 0) { printf("erreur fstat"); exit(2); }
taille = statbuf.st_size;

base = mmap (NULL, taille, PROT_READ, MAP_PRIVATE, file_in, 0) ;

if (base == MAP_FAILED) { printf ("mmap failed \n"); exit(3); }
```

3- **couplage partagé** d'un fichier écriture

```
/* un fichier couplé avec mmap doit être ouvert en lecture ou en lecture/écriture*/
if ((file = open (fichier, O_RDWR | O_CREAT | O_TRUNC, 0644)) == -1) {
        printf("erreur …\n"); exit(1);
}
/* Il faut fixer la taille du fichier avant le couplage au minimum à la taille couverte par la projection en mémoire : lseek (taille-1) + write d'un octet */
/* Sinon, erreur bus au premier accès au segment mémoire */
char carqlq = 'x';
lseek (file, taille - 1, SEEK_SET);
write (file, &carqlq, 1);

 /* MAP_SHARED est nécessaire pour que les écritures soient visibles dans le fichier */
if ((base = mmap (NULL, taille, PROT_WRITE, MAP_SHARED, file, 0)) == (caddr_t) -1) {
        printf ("erreur mmap\n");   exit(2);
}
base[0]='a' ;  base[1]='b' ;  …
```

4- « **minichat » version tableau blanc**

- Un fichier partagé contenant les 20 derniers messages
- Projeté avec mmap dans chaque participant
- Pas besoin de serveur
- messages affichés toutes les secondes si de nouveaux messages sont arrivés ? penser à la variable dernier0