

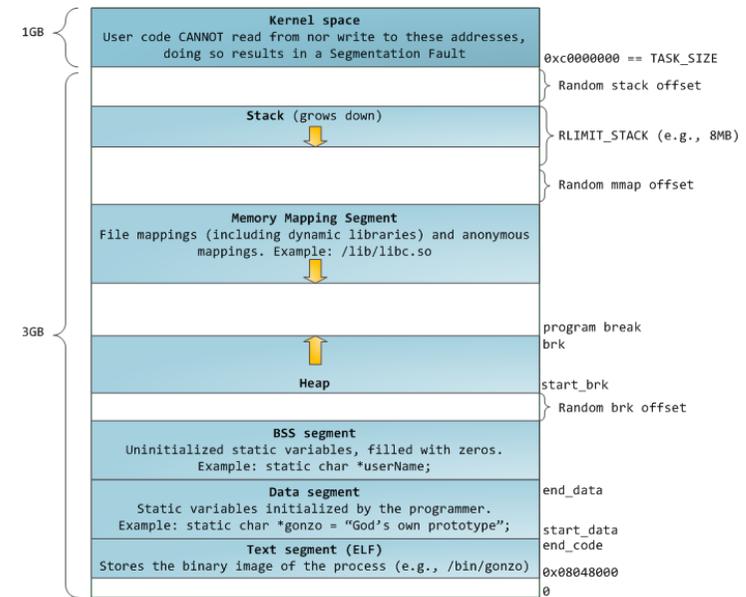
C. Raïevsky



Département Informatique

2019

Position des segments de mémoire partagé



Création d'un segment partagé (1/2)

```
int shm_open(const char *name, int oflag, mode_t mode);
```

- ▶ Crée un objet noyau représentant un segment de mémoire
- ▶ Retourne un descripteur de fichier correspondant à cet objet

name

- ▶ Nom de l'objet
- ▶ Peut être utilisé par des processus sans rapport
- ▶ doit commencé par '/' pour être portable

Création d'un segment partagé (2/2)

```
int shm_open(const char *name, int oflag, mode_t mode);
```

oflag

- ▶ O_RDONLY : lecture seule
- ▶ O_RDWR : lecture écriture
- ▶ O_CREAT : création s'il n'existe pas
- ▶ O_EXCL : création, erreur s'il existe
- ▶ O_TRUNC : création, effacement s'il existe

mode → voir open

Mapping dans l'espace d'adressage du processus (1/3)

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

addr

- ▶ Souhait d'adresse pour le début du mapping
- ▶ NULL pour laisser le noyau choisir (valeur la plus portable)

length

- ▶ Taille du segment
- ▶ Seule cette partie du segment sera initialisée

offset

- ▶ Décalage en début d'objet (segment mémoire, fichier)
- ▶ Doit être aligné sur une page
(taille des pages : `sysconf(_SC_PAGE_SIZE)`)

5 / 9

Mapping dans l'espace d'adressage du processus (2/3)

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

prot

- ▶ PROT_EXEC : droits d'exécution
- ▶ PROT_READ : droits en lecture
- ▶ PROT_WRITE : droits en écriture
- ▶ PROT_NONE : aucun droit

Les droits demandés doivent être compatibles avec ceux utilisés lors du `shm_open`

fd

- ▶ descripteur de fichier retourné par `shm_open`
- ▶ (Même fonctionnement si ce descripteur décrit un fichier ordinaire)

6 / 9

Mapping dans l'espace d'adressage du processus (3/3)

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

flags

- ▶ MAP_SHARED : le segment mémoire est partagé, les autres processus ont accès aux modifications
- ▶ MAP_PRIVATE : les modifications apportées au segment ne sont pas visibles par les autres processus
- ▶ MAP_POPULATE : prefault pages

Valeur de retour

- ▶ Adresse mémoire du mapping
- ▶ Dans l'espace d'adressage virtuel du processus

7 / 9

Accès

- ▶ `mmap` retourne un pointeur
- ▶ Les données peuvent être modifiées comme avec n'importe quel pointeur
- ▶ Un soin particulier doit être porté aux accès concurrents.

8 / 9

Durée de vie, fermeture

L'objet noyau créé par `shm_open`

- ▶ Survie au processus qui l'a créé
- ▶ Est détruit par un `shm_unlink`
 - ▶ **Si** plus aucun processus n'y accède
- ▶ Est détruit par un redémarrage