Programmation Système et Réseau

Communication Inter-Processus (IPC)

La mémoire partagée

Équipe pédagogique

CY Tech

Bibliographie - Sitographie

- Slides de Juan Ángel Lorenzo del Castillo, Seytkamal Medetov et Son Vu
- Linux : Programmation système et réseau de Joëlle Delacroix Dunod
- Système d'exploitation de Andrew Tanenbaum, Pearson Education

Programmation Système et Réseau

Communication Inter-Processus (IPC)

La mémoire partagée

Équipe pédagogique

CY Tech

Bibliographie - Sitographie

- Slides de Juan Ángel Lorenzo del Castillo, Seytkamal Medetov et Son Vu
- Linux : Programmation système et réseau de Joëlle Delacroix Dunod
- Système d'exploitation de Andrew Tanenbaum, Pearson Education
- Slides de A. Silberschatz, P. B. Galvin et G. Gagne
- Slides de A. Frank - P. Weisberg (Introduction to Operating System)
- Slides H. Bourzoufi , Arnaud Lewandowski, François Bourdon, Joelle Delacroix, Mirian Halfeld-Ferrari, A. B. Dragut
- `https://www.lri.fr/~anab/teaching/DevLog/cours4-threads.pdf`
- `http://cours.polymtl.ca/inf2610/documentation/notes/chap4.pdf`

Bibliographie - Sitographie

- Les files de messages IPC : http://supertos.free.fr/supertos.php\?page=37
- La fonction ftok() : http://manpagesfr.free.fr/man/man3/ftok.3.html et http://supertos.free.fr/supertos.php\?page=36
- Manuel de msgctl() : http://manpagesfr.free.fr/man/man2/msgctl.2.html#lbAB
- Communication Interprocessus : https://mtodorovic.developpez.com/linux/programmation-avancee/\?page=page_5
- Inter Processus Communications (I.P.C.) : http://www-igm.univ-mlv.fr/~dr/NCS/node197.html

Les IPC (Inter Processus Communication)

- Fournit des facilités de communication entre processus sur une même machine

Les IPC (Inter Processus Communication)

- Fournit des facilités de communication entre processus sur une même machine
- Externes au SGF

Les IPC (Inter Processus Communication)

- Fournit des facilités de communication entre processus sur une même machine
- Externes au SGF
- 3 fonctions : Files de messages, segments de mémoire partagée, sémaphores

Les IPC (Inter Processus Communication)

- Fournit des facilités de communication entre processus sur une même machine
- Externes au SGF
- 3 fonctions : Files de messages, segments de mémoire partagée, sémaphores
- Identifiés et manipulés à travers une clé numérique

Les IPC (Inter Processus Communication)

- Fournit des facilités de communication entre processus sur une même machine
- Externes au SGF
- 3 fonctions : Files de messages, segments de mémoire partagée, sémaphores
- Identifiés et manipulés à travers une clé numérique
- Interface #include <sys/ipc.h> #include <sys/shm.h> #include <sys/types.h>

Les IPC (Inter Processus Communication)

- Fournit des facilités de communication entre processus sur une même machine
- Externes au SGF
- 3 fonctions : Files de messages, segments de mémoire partagée, sémaphores
- Identifiés et manipulés à travers une clé numérique
- Interface #include <sys/ipc.h> #include <sys/shm.h> #include <sys/types.h>
- Deux commandes : ipcs (voir un IPC) et ipcrm (supprimer un IPC)

Mémoire partagée

La mémoire partagée ou SHM

Mémoire partagée

- Zone de mémoire commune à plusieurs processus

Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création

Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création
  - shmid

Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création
  - shmid
- Attachement de cette zone par les processus utilisateurs

Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création
  - shmid
- Attachement de cette zone par les processus utilisateurs
- Données non typées

Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création
  - shmid
- Attachement de cette zone par les processus utilisateurs
- Données non typées
- Aucune filiation exigée

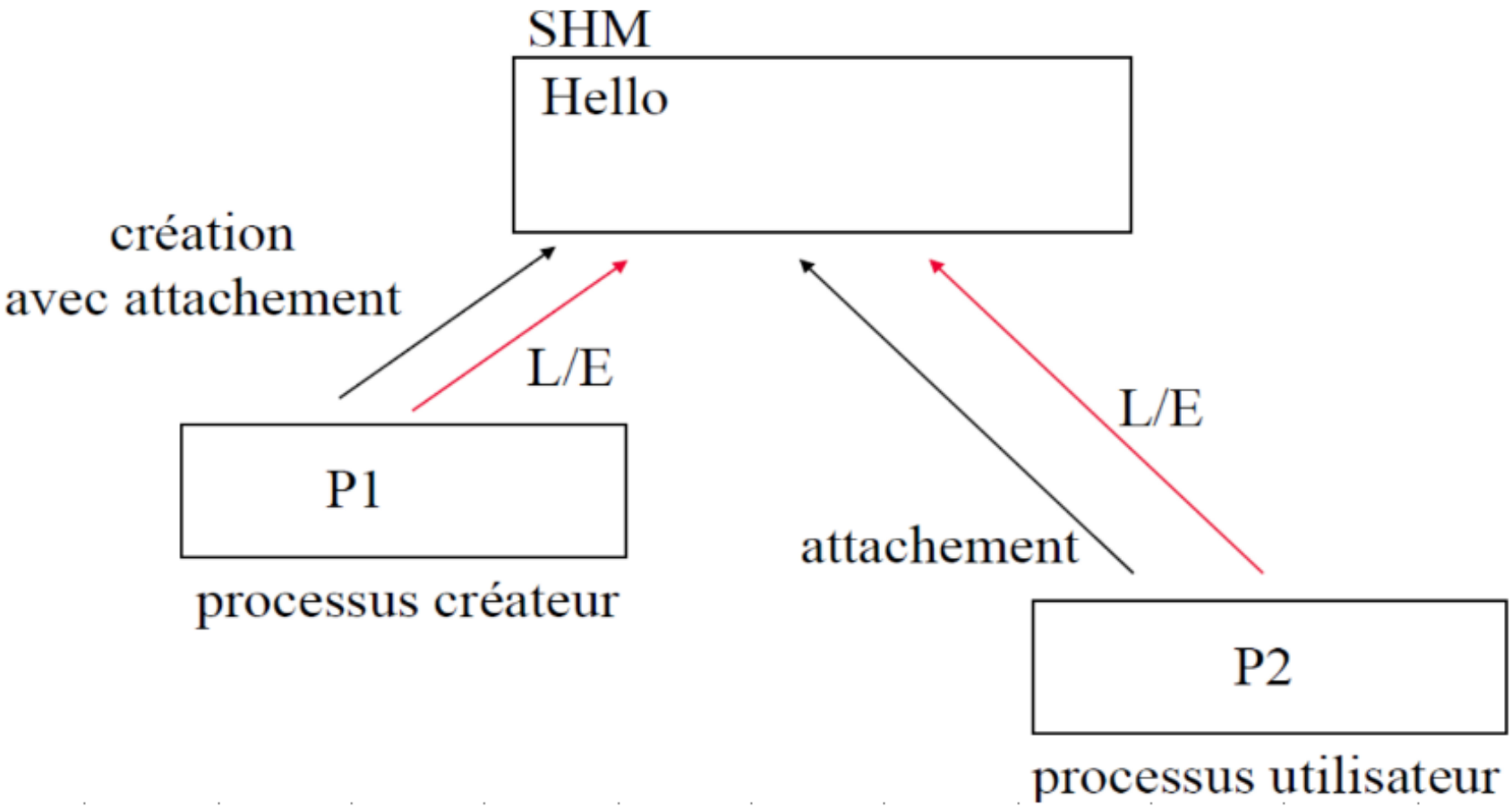
Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création
  - shmid
- Attachement de cette zone par les processus utilisateurs
- Données non typées
- Aucune filiation exigée
- Lecture non destructrice : zone de mémoire

Mémoire partagée

- Zone de mémoire commune à plusieurs processus
- Identificateur SHM : entier fourni par le système à la création
  - shmid
- Attachement de cette zone par les processus utilisateurs
- Données non typées
- Aucune filiation exigée
- Lecture non destructrice : zone de mémoire
- Structure associée : fichier shm.h

Mémoire partagée



Mémoire partagée

- Création SHM avec une clé

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus
  - Fournir shmid

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus
  - Fournir shmid
  - Récupération pointeur début zone SHM

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus
  - Fournir shmid
  - Récupération pointeur début zone SHM
- Lecture ou écriture

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus
  - Fournir shmid
  - Récupération pointeur début zone SHM
- Lecture ou écriture
  - Accès mémoire

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus
  - Fournir shmid
  - Récupération pointeur début zone SHM
- Lecture ou écriture
  - Accès mémoire
- Détachement SHM par chaque processus

Mémoire partagée

- Création SHM avec une clé
  - Récupération de l'identificateur shmid
  - Autorisation d'accès par le créateur
- Attachement SHM par un processus
  - Fournir shmid
  - Récupération pointeur début zone SHM
- Lecture ou écriture
  - Accès mémoire
- Détachement SHM par chaque processus
- Libération SHM par le processus créateur

Mémoire partagée

- Accès via shmget()
  - Création d'une zone de mémoire partagée
  - Accès à une zone déjà existante
- Prototype : int shmget(key_t cle, int taille, int option)
- Retourne l'identifiant shmid de la zone sinon -1 et errno est :
  - EACCESS : problème de droits
  - EEXIT : une IPC a déjà ce numéro
  - EIDRM : IPC détruite
  - EINVAL : paramètres invalides
  - ENOMEM : table des IPCs saturée
  - ENOENT : pas de mémoire partagée à la clé spécifiée
  - EFAULT : paramètre incorrects

Mémoire partagée

Création :

- `int shmget(cle,taille,IPC_CREAT | IPC_EXCL | 0660);`
  - Retourne le `shmid`, sinon `-1`.
  - `IPC_EXCL` permet l'exclusivité (erreur si clé existe déjà)
- Sans clé :
  - Utilisation de `IPC_PRIVATE` comme clé

Récupération d'un `shmid` :

- `shmget(cle, 0);`

Contrôle :

- `int shmctl(int shmid, int op, struct shmid_ds *buf)`
  - Destruction d'une SHM : `shmctl(cle, IPC_RMID, NULL);`
  - `IPC_STAT` lecture de la structure dans buf
  - `IPC_SET` modification de la structure à partir de buf

Mémoire partagée

Attachement :

- `void *shmat(int shmid,const void *shmadd,int option);`

La SHM est attachée au segment de données du processus à l'adresse spécifiée par `*shmadd` :

- `*shmadd` = 0 : la région est attachée à la première adresse disponible
- `*shmadd` != 0 et :
  - `option` contient `SHM_RND` : attachée à l'adresse (`shmadd` modulo `SHMLBA`)
  - `option` ne contient pas `SHM_RND` : attachée à l'adresse `shmadd`

Lecture seule possible avec l'option `SHM_RDONLY`

Retourne l'adresse de la région partagée en cas de succès, sinon -1 et `errno` est :

- `EACESS` : Problème de droits
- `EINVAL` : Clé ou adresse invalide
- `EIDRM` : Ressource détruite
- `ENOMEM` : La mémoire est insuffisante

Un processus fils hérite de toutes les zones mémoires partagées attachées à l'espace d'adressage de son père

Mémoire partagée

Détachement :

- `void *shmdt(const void *shmadd);`
  - La région de mémoire détachée devient inaccessible pour le processus appelant.
  - La terminaison d'un processus entraîne le détachement de toutes les régions qu'il avait préalablement attachées à son espace d'adressage.

Mémoire partagée

```c
#include <sys/types.h>
#include <sys/ipc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/shm.h>
#define CLE 217          // cle de la SHM
int main()
{ char* mem; // pointeur sur la SHM
  int shmid; // id de la SHM
  shmid = shmget((key_t)CLE, 1000, IPC_CREAT|0750);
      //creation SHM
  mem = shmat(shmid,NULL,0); //ecriture SHM
  strcpy(mem,"Vive la mémoire partagée");
  exit(0); }
```

Mémoire partagée

```c
#include <sys/types.h>
#include <sys/ipc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/shm.h>
#define CLE 217          // cle de la SHM
int main()
{ char* mem; // pointeur sur la SHM
  int shmid; // id de la SHM
  shmid = shmget((key_t)CLE, 0, 0); //recup id SHM
  mem = shmat(shmid,NULL,0); //attachement SHM
  printf("Lu dans SHM : %s\n", mem); //lecture SHM
  shmdt(mem); // detachement SHM
  shmctl(shmid, IPC_RMID, NULL); //destruction SHM
exit(0); }
```

Mémoire partagée

Pour executer :
Ouvrir un terminal et lancer :
>./ecrivainSHM
puis
> ipcs
Ouvrir un autre terminal et lancer :
>./lecteurSHM
Puis
>ipcs