

Communication inter-processus

Introduction

Les processus ne peuvent pas communiquer directement car un seul processus est actif à un instant donné. Il est nécessaire pour cela d'utiliser l'équivalent d'une boîte aux lettres pour l'échange des messages.

Les mécanismes mis en œuvre sur les systèmes sont les : TUBES
FILES DE MESSAGES.

Les tubes (pipe)

Les tubes (pipe)

Moyen de communication entre deux processus s'exécutant sur une même machine

Propriétés :

- Fichiers particuliers (SGF)
- File de données (FIFO) → Fonctions de lecture et écriture bloquantes
- Lectures destructrices
- Lectures multiples: le 1er qui lit récupère les données

Les tubes (pipe)

Autres propriétés :

- tube vide et lecture → processus bloqué jusqu'à dépôt de données
- tube non vide et nombre de données à lire > données existantes → processus lit le nombre de car disponibles
- tube plein et écriture → processus bloqué jusqu'à lecture de données

Les tubes (pipe)

Les types de tubes :

- Tubes nommés (sur disque)
- Tubes anonymes (mémoire)

Les tubes nommés

Caractéristiques:

- fichier portant un nom
- création par la fonction `mknod()`
- ouverture par `open()`

Les tubes nommés

→ Création tube

mknod (nom_du_fichier, accès+S_IFIFO)

mkfifo (nom_fichier, accès)

- utilisation de la fonction de création d'un i-node particulier
- spécifier un nom de fichier
- donner les droits d'accès
- création d'un fichier fonctionnant en mode FIFO

→ Ouverture tube

desc = open(nom_du_fichier, mode)

- ouverture en lecture si mode = O_RDONLY
- ouverture en écriture si mode = O_WRONLY
- ouverture en maj si mode = O_RDWR
- ouverture bloquante / non bloquante mode = O_NDELAY

Les tubes nommés

→ lecture dans le tube

read (desc, buf, nb)

- si O_NDELAY à l'ouverture
 - retour code erreur si lecture dans un tube vide
- sinon
 - processus bloqué si tube vide, attente suffisamment de données à lire

→ écriture dans le tube

write (desc, buf, nb)

- si O_NDELAY à l'ouverture
 - retour code erreur si tube plein
- sinon
 - processus bloqué si tube plein, attente tube suffisamment vide pour écrire

Les tubes nommés

EXEMPLE DE PROGRAMMATION

```
/* Processus ecrivain */
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
main()
{  int tub;
   mkfifo ("fictub", S_IRUSR | S_IWUSR) /* création fichier
FIFO */
   tub = open("fictub",O_WRONLY) /* ouverture fichier */
write (tub,"0123456789",10); /* écriture dans fichier */
close (tub);
   exit(0);}

```

Les tubes nommés

EXEMPLE DE PROGRAMMATION

```
/* Processus lecteur */
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
main()
{  int tub;
   char buf[11];
   tub = open("fictub",O_RDONLY) /* ouverture fichier */
   read (tub,buf,10); /* lecture du fichier */
   printf("J'ai lu %s\n, buf);
   close (tub);
   exit(0); }
```

Les tubes anonymes

PROPRIETES :

- Structure sans nom
- Création en mémoire : Fonction `pipe()`
- Duplication des processus : Fonction `fork()`
- Deux descripteurs : Lecture et écriture
- Fonction d'accès : `read()`, `write()`, `close()`

Les tubes anonymes

Les fonctions de programmation

pipe (tab) → crée un tube

- retourne les n° des deux descripteurs dans un tableau
 - tab[0]: n° du descripteur de lecture: dl
 - tab[1]: n° du descripteur d'écriture: de

read (dl, buf, nb) → lecture dans un tube

- dl: n° descripteur lecture / • buf : zone de réception des octets
- nb : nombre d'octets à lire

write (de, buf, nb) – écriture dans un tube

- de: n° du descripteur écriture / • buf: zone d'émission des octets
- nb: nombre d'octets à écrire

Les tubes anonymes

Exemple de programmation

```
#include <stdio.h>

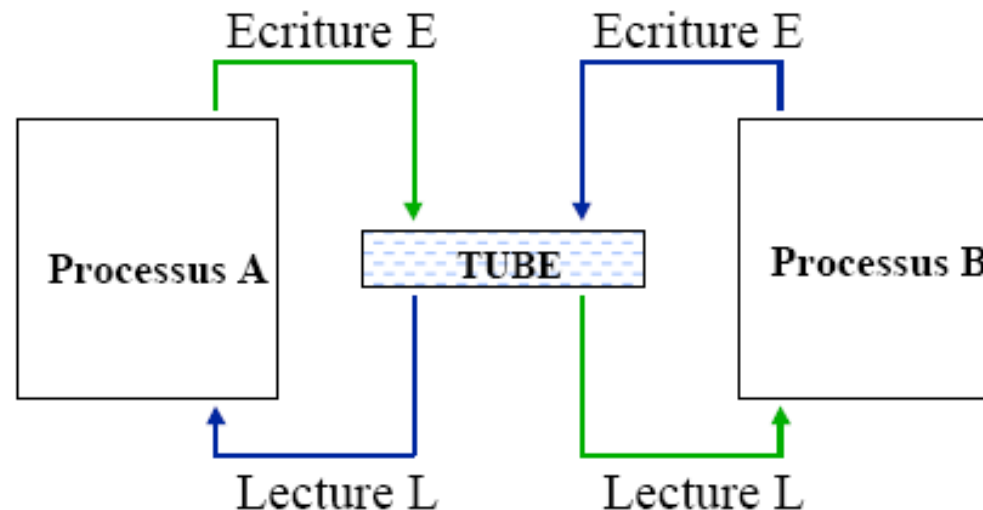
int pip[2]; /* descripteur de pipe */
char buf [6];

pere(){write (pip[1], "hello", 5); exit(0);} /* écriture pipe */
fils() {read (pip[0], buf, 5); printf("%s\n", buf); exit(0);} /* lecture
pipe */

main()
{ pipe(pip); /* creation pipe */
  switch (fork())
  {   case -1: perror("pipe"); exit(1);
      case 0: fils();
      default: pere();}
}
```

Les tubes anonymes

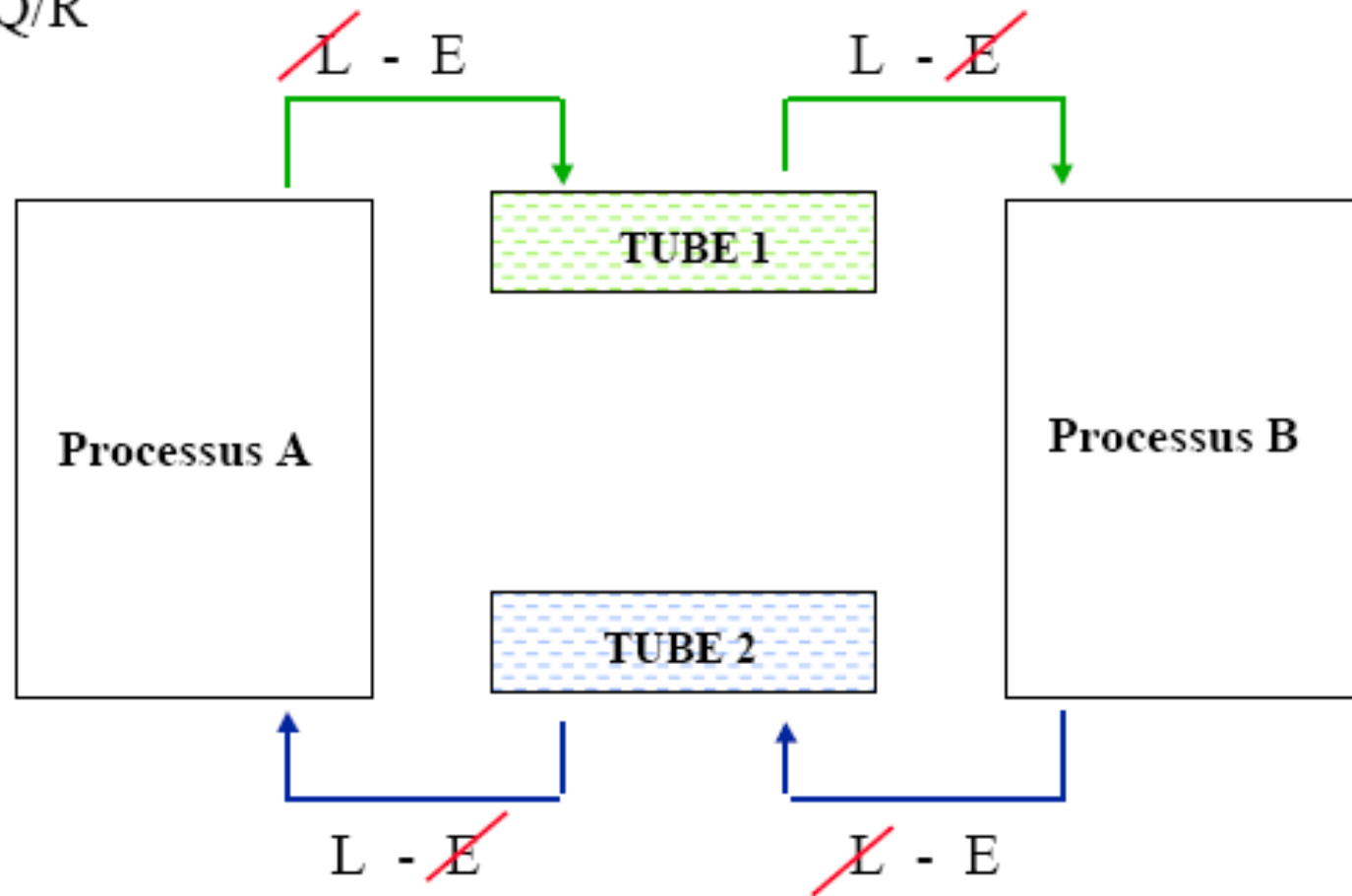
Synchronisation



- Soit: PA transmet à PB ou PB transmet à PA
- SI
- PA dépose et PA lit => PB bloqué
- PA dépose, PB dépose et PB lit => PB lit sa propre donnée

Les tubes anonymes

Modèle Q/R



Fermeture des descripteurs inutiles