II - PARTIE: LES PRIMITIVES

Les tubes sans nom sont liaison unidirectionnelle de communication donc la taille maximale peut varie d’une version a une autre d’UNIX. Elle est approximativement égale a 4ko. Le fonctionnement de la communication sans nom repose sur trois primitives principales :

1. La primitive de création d’un tube : pipe()
2. La primitive de Lecture dans un Tube : read()
3. La primitive d’écriture dans un tube : Write()
4. - Primitive de création d’un tube : pipe()

Les tubes sont créés par la primitive pipe() qui est un appel système auquel on passe un tableau de 2 entiers. Les deux entiers sont appelés des descripteurs et mis en paramètre dans les deux positions du tableau. Ils peuvent être considérer comme une valeur entière que le système d’exploitation utilise pour accéder à un fichier.

* Un descripteur pour accès en lecture P[0]
* Un descripteur pour accès en écriture P[1]

Au retour de notre appel système, en cas d’échec le pipe nous renverra -1 sinon en cas de succès on aura 0 et tube aura été créé. Il est noté que seul note que seul le processus créateur du tube et ses descendants peuvent accéder au tube.

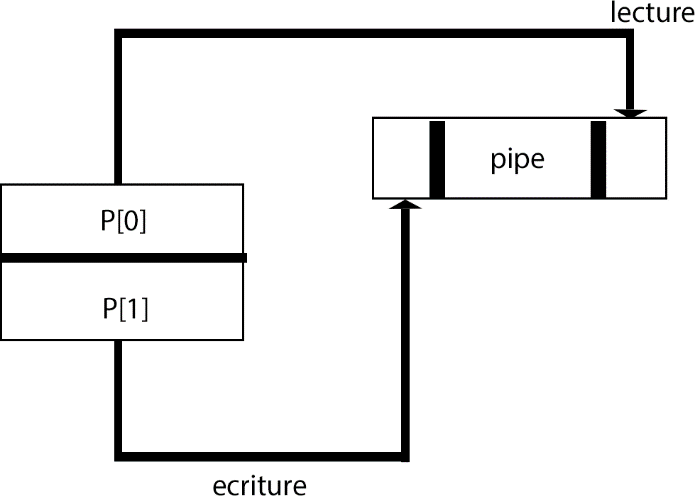
Exemple : Déclaration pour créer un tube :

#include <unistd.h>

int P[2] ;

Pipe(p) ;

Illustration : Pipe Lecture et Ecriture



1. – La primitive de Lecture dans un Tube : Read()

La primitive Read() permet de lire dans un fichier mais cette opération doit se faire de façon atomique. On utilise le descripteur p[0]. Les précautions à prendre voudrait qu’un processus ferme systématique les descripteurs donc il n’a pas besoin : on lit dans le tube et on ferme le descripteur d’écrire p[1]. Cela permet d’éviter des erreurs aboutissant à des situations d’interblocages (Deadlock).

Exemple Bout de programme « Code Complet juste en bas »

char contenu [100] ;

int p[2] ;

…..

Close(p[1]) ;

Read(p[0], contenu,20) ;

Commentaire : On demande ici la lecture de 20 caractères dans le tube p. les caractères lus sont rendus dans la zone de contenu.

1. – La primitive d’écriture dans le Tube :Write()

L’écriture dans le tube se fait en utilisant la primitive write() et le descripteur p[1] cette fois.

Exemple : Bout de programme « Code complet en dessous »

Char contenu[100] ;

Intp[2] ;

….

Close(p[0]) ;

Contenu = ‘’ communication unidirectionnelle ‘’ ;

Write(p[1] , contenu,20) ;

Commentaire : On fait une demande d’écriture de 20 caractères dans le tube de descripteur p[1.] la séquence a écrire est prise dans la zone de contenu. write renvoie le nombre d’octet écrit. Ici aussi on ferme le descripteur de lecture en close(p[0]) ;

1. – Code complet : Communication unidirectionnelle du processus vers le processus père.

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

int main(){

int p[2];

char contenu[100];

char \*texte = "communication unidirectionnelle";

pipe(p) // appel systeme pour creer un tube sans nom

if(fork()==0){

close(p[0]); // fermeture du descripteur de lecture

write(p[1] , texte, 20);

close(p[1]) // fermeture du descripteur d'ecrtiure

}else{

close(p[1]);

read(p[0] , contenu , 20);

close(p[0]);

}

return 0;

}