

Communication par tubes sous Unix

Oum-El-Kheir.Aktouf@grenoble-inp.fr

Introduction

- Un tube est un fichier → appartient au SGF d'Unix
 - i-nœud, lecture, écriture, redirection, etc.
- Mécanismes de communication unidirectionnels
 - Une extrémité de lecture
 - Une extrémité d'écriture
 - 2 entrées dans la table des fichiers ouverts (1 en lecture, 1 en écriture)
- Objets de type fifo : lecture par un bout, écriture par l'autre
- Lecture destructrice
- Communication d'un flot continu de caractères (mode stream)
- Capacité finie

Introduction

- Nombre de lecteurs : nombre de descripteurs associés à l'entrée en lecture
- Nombre d'écrivains : nombre de descripteurs associés à l'entrée en écriture



Les tubes sans nom (ordinaires)

- Fichier sans nom
- Contenu dans aucun répertoire
- Utilisation par un processus à travers un descripteur
 - Création
 - Héritage

Les tubes sans nom (ordinaires)

- Création

```
#include <unistd.h>
```

```
int pipe(int p[2])
```

`p[0]` : descripteur en lecture

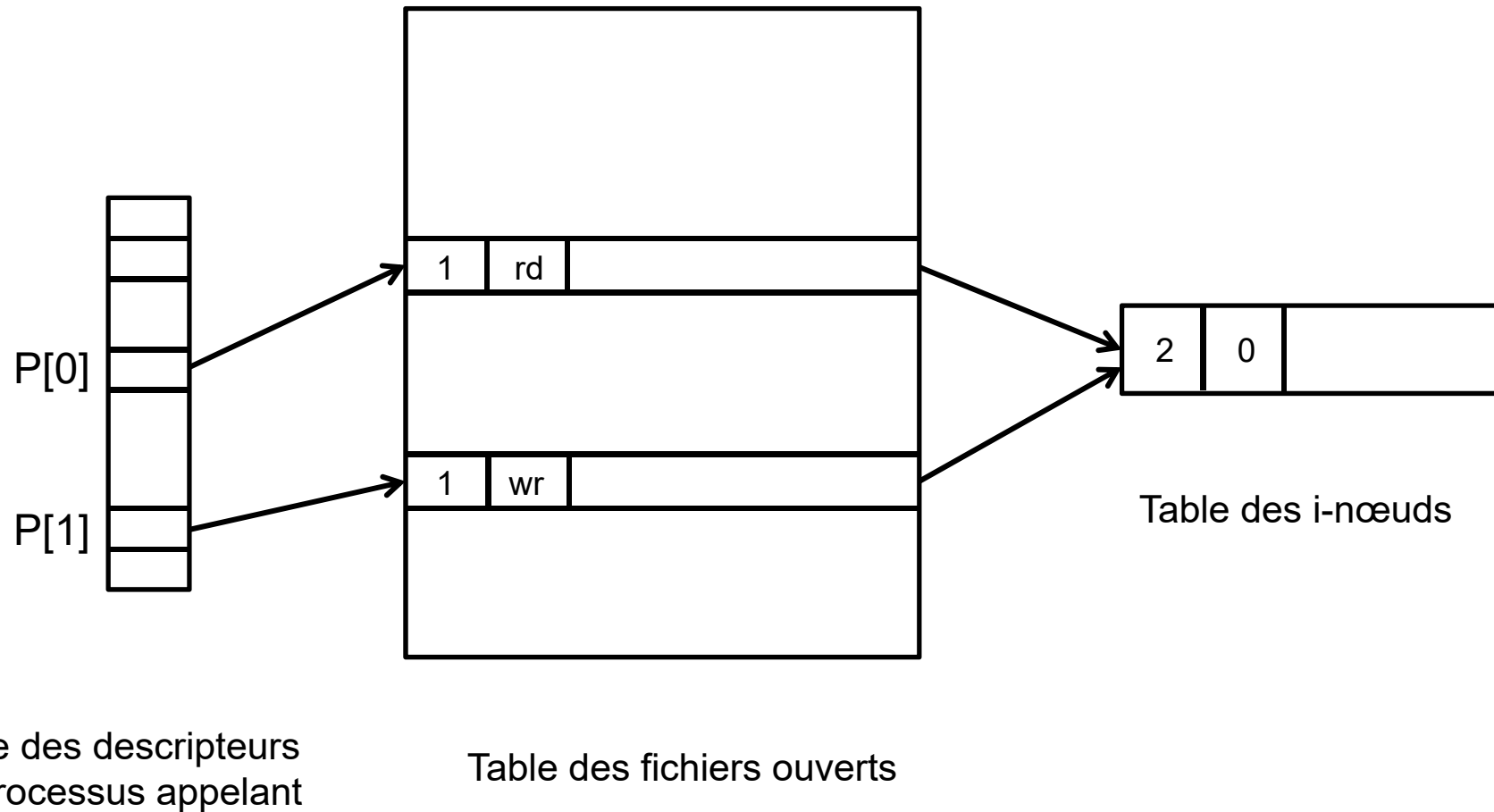
`p[1]` : descripteur en écriture

Retour : 0 si succès, -1 si échec



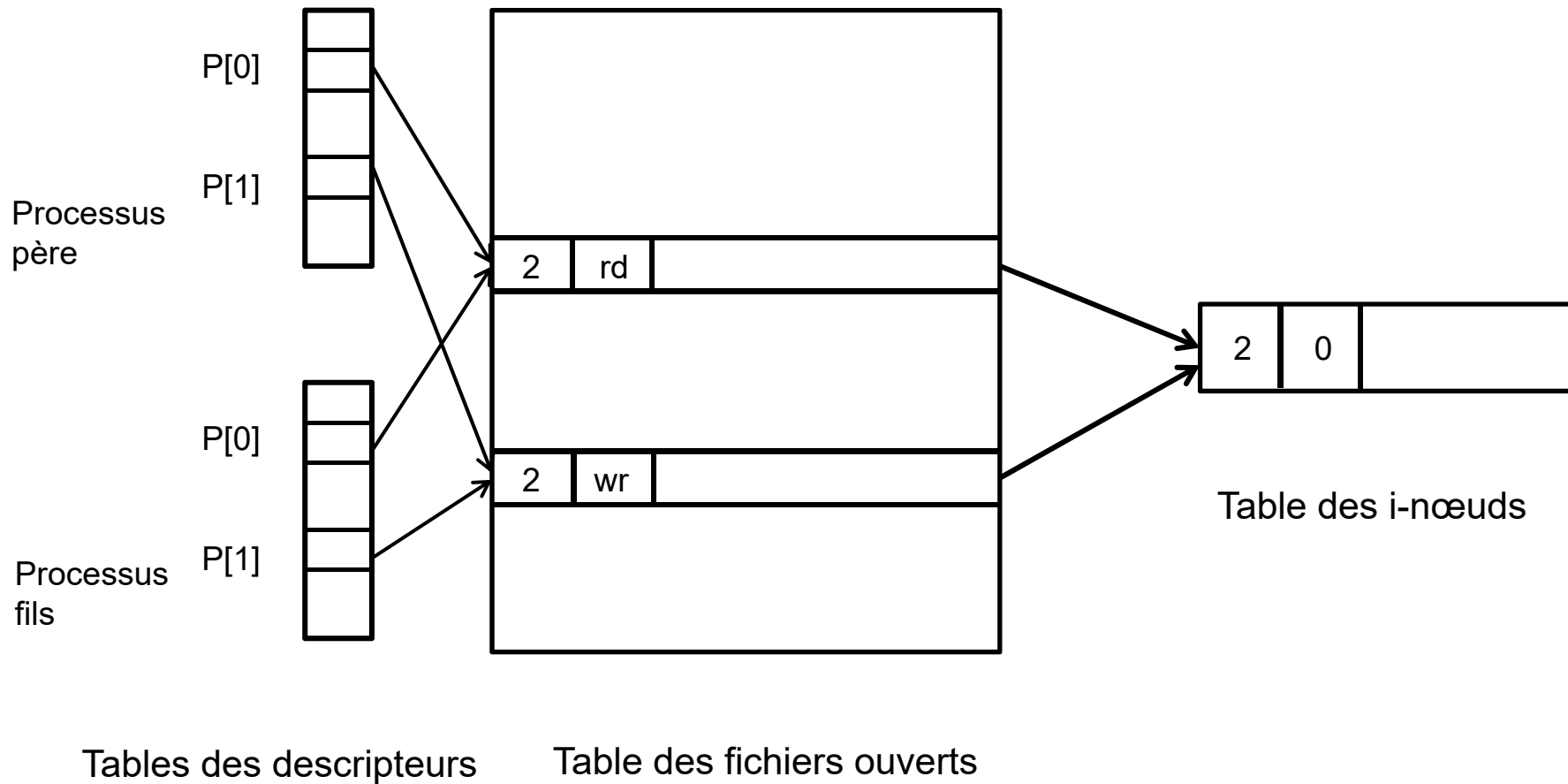
Les tubes sans nom (ordinaires)

- Héritage



Les tubes sans nom (ordinaires)

- Héritage



Les tubes sans nom (ordinaires)

- Lecture

```
nb_lu = read(p[0], buf, TAILLE_BUF)
```

- Utilise un descripteur en lecture
- Lecture d'au plus `TAILLE_BUF` caractères qui seront placés à l'adresse `buf`

Les tubes sans nom (ordinaires)

- Si le tube n'est pas vide et contient n caractères
 - La fonction `read()` extrait `nb_lu=min(n, TAILLE_BUF)` caractères et les place en mémoire à l'adresse `buf`

Les tubes sans nom (ordinaires)

- Si le tube est vide
 - Si le nombre d'écrivains est nul
 - Fin de fichier atteinte
 - Aucun caractère lu
 - `Nb_lu` = 0

Les tubes sans nom (ordinaires)

- Si le tube est vide
 - Si le nombre d'écrivains n'est pas nul
 - Si lecture bloquante (cas par défaut)
 - Processus mis en sommeil
 - Si lecture non bloquante
 - Retour immédiat de -1 ou 0

Les tubes sans nom (ordinaires)

- Ecriture

```
nb_ecrit = write(p[1], buf, n)
```

- Utilise un descripteur en écriture
- Ecriture dans le tube `p[1]` des `n` premiers caractères accessibles à partir de l'adresse `buf`

Les tubes sans nom (ordinaires)

- Si le nombre de lecteurs dans le tube est nul
 - Signal SIGPIPE délivré au processus appelant
 - Termine le processus par défaut
 - Message d'erreur « Broken pipe »

Les tubes sans nom (ordinaires)

- Si le nombre de lecteurs n'est pas nul

- Cas1 : écriture bloquante

Le retour de l'appel n'a lieu que quand n caractères ont été écrits

- si $n \leq \text{PIPE_BUF}$, écriture atomique dès qu'il y a n emplacements libres dans le tube

- si $n > \text{PIPE_BUF}$, écriture découpée en plusieurs écritures de PIPE_BUF caractères

Les tubes sans nom (ordinaires)

- Si le nombre de lecteurs n'est pas nul
 - Cas2 : écriture non bloquante
 - si $n > \text{PIPE_BUF}$, écriture non réalisée, retour de -1
 - si $n \leq \text{PIPE_BUF}$ et il y a au moins n emplacements libres, écriture atomique réalisée
 - si $n \leq \text{PIPE_BUF}$ et il y a moins de n emplacements libres, écriture non réalisée et retour immédiat de -1 ou 0

Les tubes sans nom (ordinaires)

- Autres opérations
 - Fermeture : `close()`
 - Duplication : `dup()`
 - Extraction des caractéristiques d'un tube : `fstat()`
 - Extraction et modification des caractéristiques d'un tube : `fcntl()`

Les tubes sans nom (ordinaires)

- Exemple

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main(void)
{
    int p[2];
    char chaine[8]="";

    if (pipe(p)==-1) {
        perror("Creation de tube");
        exit(1);
    }
    write(p[1], "Hello World!", 12);
    read(p[0], chaine, 5);
    chaine[5]='\0';
    printf("Premiere lecture : %s\n", chaine);
    read(p[0], chaine, 7);
    chaine[7]='\0';
    printf("Deuxieme lecture : %s\n", chaine);
    exit (0);
}
```

Les tubes nommés

- Ils permettent la communication entre processus sans lien de parenté car ils possèdent une référence dans le système de gestion de fichiers.
- Création

```
#include<sys/types.h>
#include<sys/stat.h>
int mkfifo(char *ref, mode_t mode);
```
- La commande " ls -l " affiche ces tubes avec la lettre **p** comme type de fichier.

Les tubes nommés

- Ouverture

- L'ouverture est bloquante par défaut

Ouverture en lecture

```
des_lec = open("fifo", O_RDONLY)
```

→ bloquante si pas d'écrivain

Ouverture en écriture

```
des_ecr = open("fifo", O_WRONLY)
```

→ bloquante si pas de lecteur

Les tubes nommés

- Ouverture non bloquante en lecture

```
des_lec = open("fifo", O_RDONLY | O_NONBLOCK)
```

→ non bloquante si pas d'écrivain

- Ouverture non bloquante en écriture

```
des_ecr = open("fifo", O_WRONLY | O_NONBLOCK)
```

→ non bloquante si pas de lecteur, renvoie -1

Les tubes nommés

- Une fois un tube nommé ouvert en lecture ou en écriture, la lecture et l'écriture se passent comme pour les tubes sans nom.
- Suppression d'un tube nommé :
 - rm
 - unlink