I2180 PROGRAMMATION SYSTÈME

PIPE

PIPES : GENERALITES

- En ligne de commande : ls -l | wc -l
- Un *pipe* (tube) est un moyen de communication entre processus.
- On utilise généralement de manière FIFO et unidirectionnel :
  - un seul processus lit
  - un seul processus écrit

PIPE

Création d'un pipe par programmation :

```
#include <unistd.h>
int pipe(int filedes[2])
```

Où: filedes : tableau vide qui sera initialisé par pipe

filedes[0] : permettra de lire dans le pipe

filedes[1] : permettra d'écrire dans le pipe

REMARQUE

- Tubes anonymes
  - Communication entre un processus père et un fils

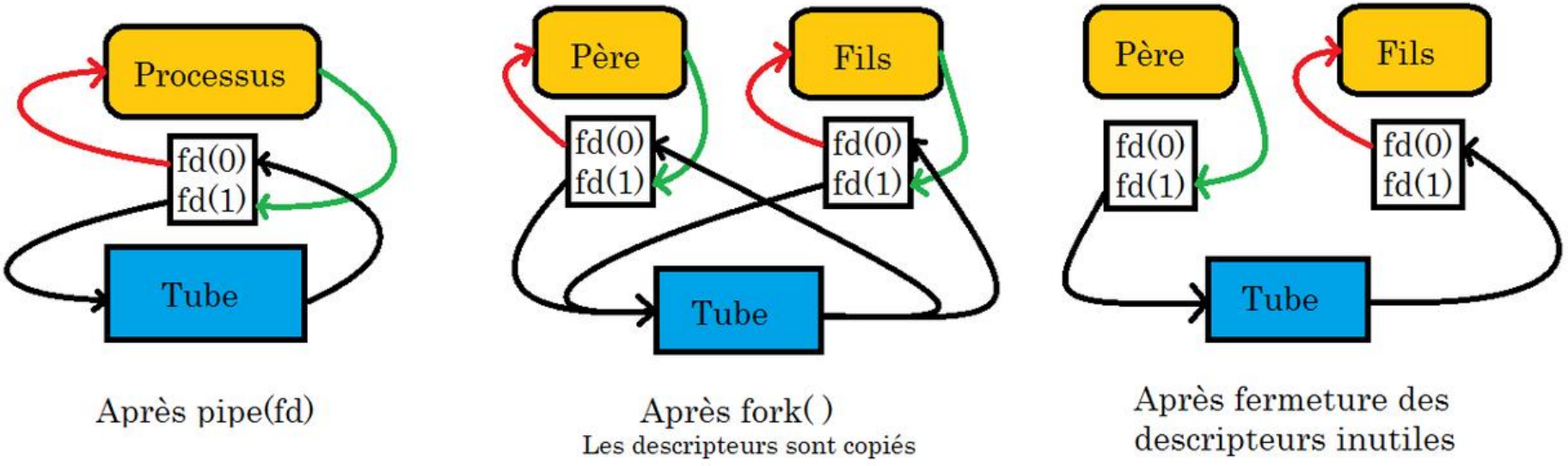
  → Objet de ce cours
- Tubes nommés
  - Communication entre processus quelconques
  - `int mkfifo(char* name, mode_t mode)`

  → Pas l'objet de ce cours

CONFIGURATION D'UN PIPE

- Création du pipe AVANT la création du processus fils
- Les 2 processus ont ainsi accès aux descripteurs de fichiers du pipe
- Dans chaque processus, fermer le descripteur de fichier non utilisé

CONFIGURATION D'UN PIPE



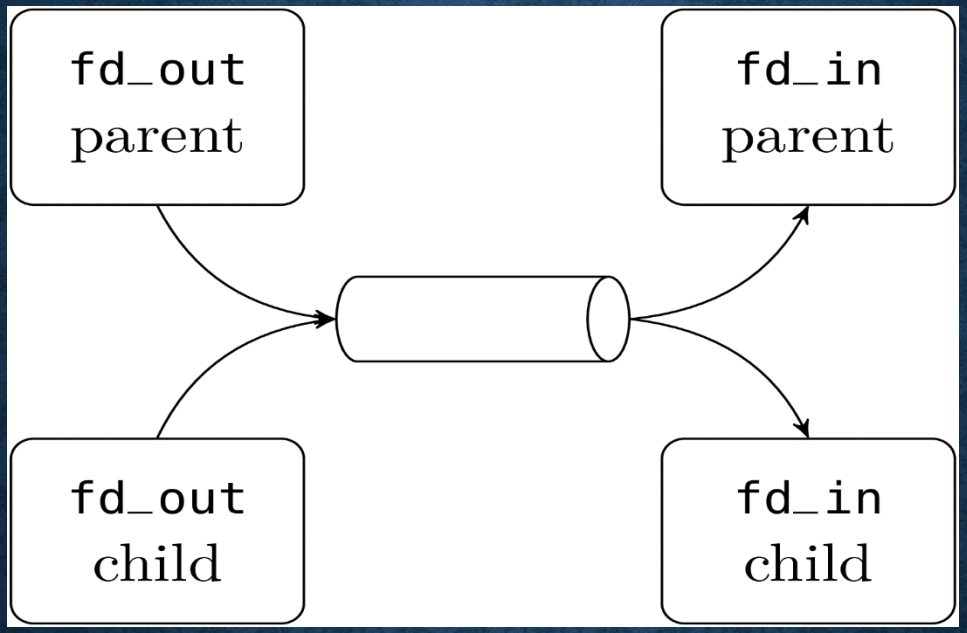
Après pipe(fd)

Après fork( )
Les descripteurs sont copiés

Après fermeture des descripteurs inutiles

https://fr.wikipedia.org/wiki/Tube_(informatique)

CONFIGURATION D'UN PIPE



- La fermeture d'un descripteur de fichier (*close*) permet d'indiquer au système d'exploitation qu'un processus n'utilise plus ce descripteur.
- Une lecture sur un descripteur de fichier totalement fermé (plus aucun processus n'a de référence vers ce descripteur) renvoie 0 (EOF)

EXEMPLE

- Envoi d'un entier depuis le processus père vers son fils

EXEMPLE

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

int main(int argc, char **argv){
   int pid, nbChar, ret, intVal;
   int pipefd[2]; /* File descriptors du pipe */
```

EXEMPLE

```
ret = pipe(pipefd); /* create pipe */
pid = fork();  /* create child process */
/* parent */
if (pid) {
   ret = close(pipefd[0]); /* pipe config. */
   intVal = 7;
   nbChar = write(pipefd[1],
                  &intVal, sizeof(int));
   close(pipefd[1]);
} else {
```

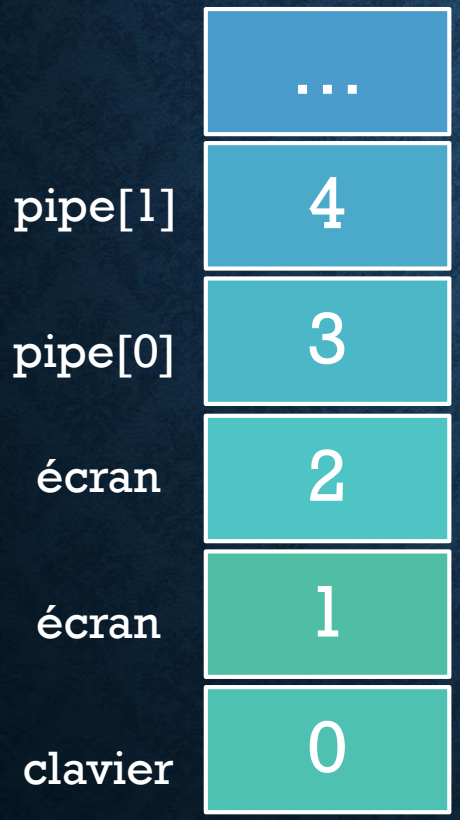
EXEMPLE

```
    /* child */
    ret = close(pipefd[1]); /* pipe config. */
    nbChar = read(pipefd[0],
                  &intVal, sizeof(int));
    printf("entier reçu par mon père: %i\n",
           intVal);
    close(pipefd[0]);
  }
}
```

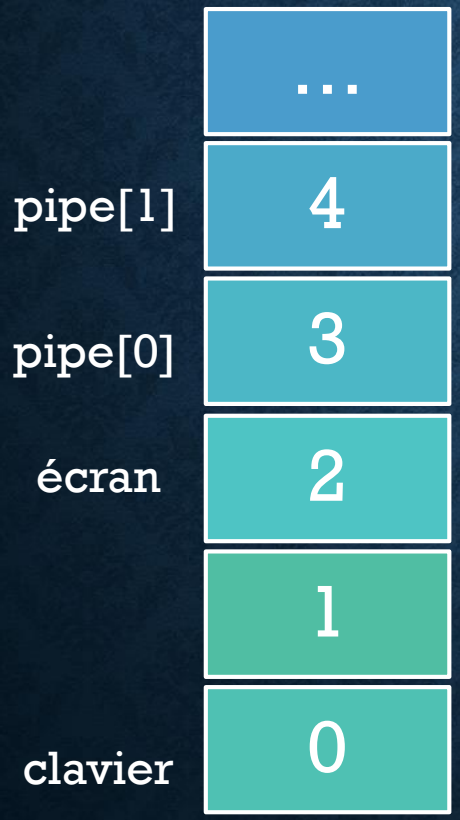
REDIRECTION

- ls -l > sortie.txt
- Exemple de redirections :
  - Redirection de l'entrée standard dans un fichier
  - Redirection de la sortie standard dans un pipe
  - …

REDIRECTION

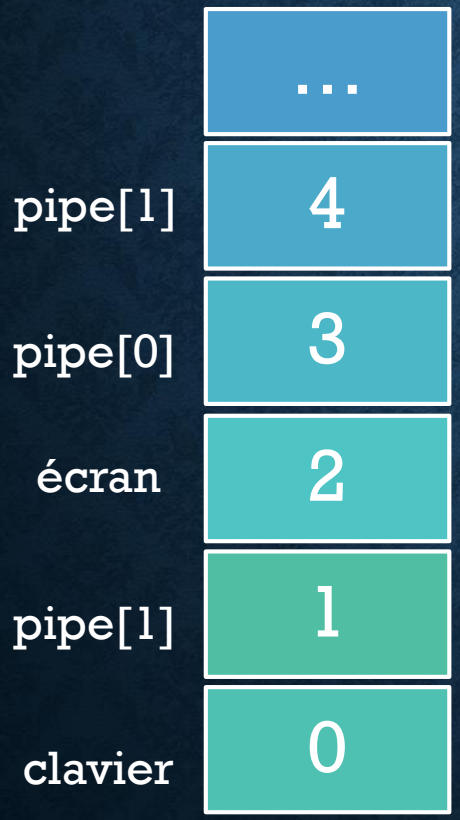


REDIRECTION



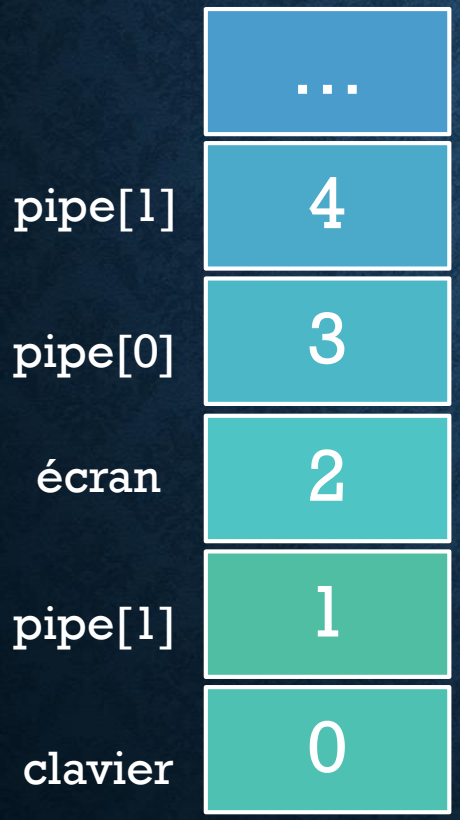

```
close(1);
```

REDIRECTION




```
close(1);
dup(pipe[1]);
```

REDIRECTION




```
close(1);
dup(pipe[1]);
```

OU

```
dup2(pipe[1],1)
```

DUP / DUP2

```
int dup(int fd);
```

- **dup**() makes the lowest available fd be the copy of *fd*

```
int dup2(int oldfd, int newfd);
```

- **dup2**() makes *newfd* be the copy of *oldfd*, closing *newfd* first if necessary