

Sujet du TP n° 2

Placez-vous dans le répertoire **PSE/TP2**.

Entrées-sorties disque - Exercice 1 (transparentes 46 à 50)

Exercice 1

- copiez le fichier **exemples/fichiers.c** et renommez-le **exercice1.c**.
- compilez ce fichier et exécutez-le (il copie le fichier en argument 1 dans le fichier en argument 2).
- modifiez-le afin qu'il affiche le nombre d'octets lus et écrits à chaque lecture/écriture; pour mieux visualiser le mécanisme, vous pouvez diminuer la taille du tampon **BUF_SIZE** afin que la lecture du fichier nécessite plusieurs appels à **read**.

Les tubes nommés (transparentes 51 à 53)

Pratique 1

- créez un tube nommé **fifo** en utilisant la commande **mkfifo**.
- visualisez le type de fichier avec la commande **ls -l**.

Pratique 2

- pour illustrer le principe des tubes, tapez dans une première fenêtre **terminal** une commande qui lit le tube : **cat < fifo**.
- ouvrez une seconde fenêtre **terminal** et tapez une commande qui écrit dans le tube : **echo bonjour a tous > fifo**.

Lisez-moi

- nous allons utiliser le module **ligne**, de la bibliothèque des **modules**, contenant les fonctions **lireLigne()** et **ecrireLigne()** permettant de lire et écrire des lignes de texte dans un fichier, et donc en particulier dans un tube nommé.
 - **int lireLigne (int fd, char *buffer)**
lit une ligne de texte dans le fichier : lit des caractères jusqu'à rencontrer le caractère '\n' et les stocke dans le buffer,
retourne le nombre de caracteres lus, ou 0 (fin de fichier), ou -1 (erreur).
 - **int écrireLigne (int fd, char *buffer)**
écrit une ligne de texte dans le fichier : ajoute le caractère '\n' au buffer et l'écrit dans le fichier,
retourne le nombre de caracteres écrits ou -1 (erreur).

Programme "serveur" (première version)

- écrivez un programme **serveur.c** qui lit une ligne de texte dans le tube nommé **fifo** et l'affiche sous forme d'un message à la console:

```
$ ./serveur
Serveur. ligne de 15 octets recue: bonjour a tous
```

dans un premier temps, pour tester le serveur, vous pouvez utiliser la commande **echo** comme client; pour cela, lancez le serveur, puis dans un autre terminal tapez **echo bonjour a tous > fifo**.

Programme "client" (première version)

- écrivez un programme **client.c** qui lit une ligne au clavier avec la fonction **fgets()** et l'envoie au **serveur** par l'intermédiaire du tube nommé **fifo**.

Serveur et client : transmission de plusieurs lignes

- mettez la réception et l'envoi d'une ligne dans une boucle afin d'émettre autant de lignes que l'opérateur le souhaite.
- l'arrêt du système se fera par la saisie et l'envoi de la ligne **"fin"**.

Fin abrupte du client

- gérez chez le serveur la fin abrupte du client (par CTRL-C par exemple) : dans ce cas la lecture sur le fifo rencontre une fin de fichier et retourne 0, le serveur doit alors s'arrêter proprement.

Fin abrupte du serveur

- que se passe-t-il chez le client quand le serveur s'arrête brutalement ?
- pour corriger ce problème, ajouter l'instruction suivante en début du programme client :
`signal(SIGPIPE, SIG_IGN);`

Journalisation chez le serveur

- le serveur n'affiche plus à la console les lignes reçues mais les écrit à la fin d'un fichier **journal.log** :
`$./serveur`
`Serveur. ligne de 15 octets écrite dans le journal.`
- ajoutez une signification particulière pour le serveur de la ligne **"init"** : initialisation du fichier `journal.log` (remise à zéro).

(c) Philippe Lalevée, 2023-2024.