

TP4 : Fifos et Setuid

Systeme M1

semaine du 18 octobre

1 Tubes nommés

Question 1. Écrire un programme `fifow` qui permettra d'écrire un sur un tube nommé avec une fréquence paramétrable. Par exemple :

```
moi@dumbbox $ mkfifo test
moi@dumbbox $ ./fifow test 2 a
```

écrira un caractère 'a' sur le tube `test` deux fois par seconde.

Question 2. Écrire un programme `fifor` qui lira des caractères sur un tube nommé avec une fréquence paramétrable. On fera en sorte qu'il affiche ce qu'il a lu sur sa sortie standard.

```
moi@dumbbox $ ./fifor test 2
aaaaa
```

Question 3. Décrivez ce qu'il se passe lorsque :

- l'écrivain écrit plus vite que ne lit le lecteur.
- l'écrivain écrit plus lentement que ne lit le lecteur.
- il y a plusieurs lecteurs sur un même tube.
- il y a plusieurs écrivains.
- on n'utilise pas ces programmes sur un fifo, mais sur un fichier régulier.

2 Petite messagerie à base de setuid

On souhaite écrire un programme qui permettra aux autres utilisateurs de vous laisser un petit message en l'exécutant :

```
buddy1@dumbbox $ ~moi/message "je ne pourrai pas venir ce soir"
```

Vous souhaitez que le répertoire `/home/moi/.messages/` où seront stockés les messages que vous avez reçus ne soit accessible qu'à vous seul (droits `rwX-----`), quitte à activer le `setuid-bit`.

Question 4. Écrivez un tel programme. Il vérifiera que le message n'est pas trop long (limité à 256 caractères par exemple). Il stockera ensuite ce message dans un fichier qui est nommé d'après l'`uid` de l'envoyeur : `/home/moi/.messages/uid`. Pour simplifier, seul le dernier message de chaque utilisateur sera conservé.

Fonctions utiles : `getuid`, `geteuid`, `getpwuid`.

Question 5. Relisez votre programme et vérifiez qu'offrir vos droits d'accès au code que vous avez écrit n'est pas dangereux. Activez alors le setuid-bit. Pour tester, demandez à votre voisin de vous laisser un message. (De son côté, il devra vous faire confiance, car il exécute un programme a priori inconnu.)

Question 6. Faites en sorte que vous puissiez consulter vos messages en lançant vous-même le programme :

```
moi@dumbbox $ ./message
buddy1 (il y a 6 minutes):
    je ne pourrai pas venir ce soir
buddy2 (hier):
    desole, je n'ai pas eu le temps de corriger le bug :-/
```

Fonctions utiles : `opendir`, `getpwuid`, `stat`, `time`.