

Université de Bretagne Sud  
UFR SSI, Dept MIS  
SYSTEME D'EXPLOITATION  
CONCURRENCE  
Les Tubes Unix

Luc Courtrai

## 1 Compression et Majuscule

### 1.1 Tube (exemple)

Reprendre dans les sources attachés au tp, l'exemple du cours tubePereFils.c et testez le après compilation. Le processus fils devrait afficher ce que le père saisit au clavier.

### 1.2 Compression : tube ordinaire

Un processus lit sur l'entrée standard des caractères et les transmet à un second processus. Celui ci supprime les espaces de la ligne puis affiche la ligne sur la sortie standard.

La communication entre les deux processus d'effectue par un tube Unix "ordinaire". Les deux processus sont écrits dans le même source

### 1.3 Tube nommé (exemple)

Reprendre les fichiers mkfifo, produTube et consoTube. Après compilation, dans un terminal lancez mkfifo et produTube et dans un autre terminal lancez consoTube. Les lignes saisies dans le premier terminal devraient s'afficher dans le deuxième.

Relancez les deux programmes mais en remplaçant le produTube par

```
cat > tube
```

Relancez à nouveau les deux programmes mais en remplaçant le consoTube par

```
cat tube
```

### 1.4 Masjuscule : 1 tube nommé

Un processus lit sur l'entrée standard des caractères et les transmet à un second processus. Celui ci passe tous les caractères en majuscule puis affiche la ligne sur la sortie standard.

La communication entre les deux processus d'effectue par un tube nommé. Ecrire le code du processus Majuscule. Le processus producteur de caractères sera la commande Unix :

```
cat > tubeMajuscule
```

### 1.5 Compression Masjuscule : 3 tubes nommés

Un processus P1 lit sur l'entrée standard des caractères et les transmet à un second processus P2.

```
cat > tubeCompress
```

Celui ci, P2, supprime les espaces de la ligne et retourne à un troisième processus P3

```
p2    tubeCompress tubeMasjuscule
```

Le troisième processus lit sur le tube et passe tous les caractères en majuscule et transmet les caractères au quatrième processus.

```
p3    tubeMajuscule tubeResult
```

p4 affiche les lignes sur la sortie standard.

```
cat    tubeResult
```

La communication entre les processus d'effectue par des tubes nommés.

## 1.6 Boucle compression majucsule

Même énoncé que le précédent mais le premier processus P1 lit sur l'entrée standard les caractères, les envoie au fur et à mesure à P2 via le tube tubeCompress et attend le retour sur le tube tubeResult puis les affiche sur la sortie standard.

```
p1    tubeCompress tubeResult
p2    tubeCompress tubeMasjuscule
p3    tubeMajuscule tubeResult
```

## 2 Serveur de Date OPTIONNEL

L'objectif de cet exercice est de mettre en place le serveur de date centralisé. Le client, la fonction getDate(char \*date), demande au serveur la date système. Le serveur crée un processus fils qui retourne au client la date système.

Toutes les communications s'effectuent par tubes nommés. Le client effectue sa demande par un tube commun (partagé par tous les clients et le serveur). Le client crée un tube privé destiné à la réponse du serveur, le nom de ce tube doit être passé en argument de la demande.

Voici le partie du code du serveur qui récupère l'heure système :

```
#include <time.h>

time_t tps;
char date[128];
tps = time(0); // recupère la date système
strcpy(date,ctime(&tps));
```