

TD 1 - LIF12 système d'exploitation

Passage d'informations sur les pipes

11 février 2017

I Envoie et lecture de données

Vous devez écrire 2 fonctions :

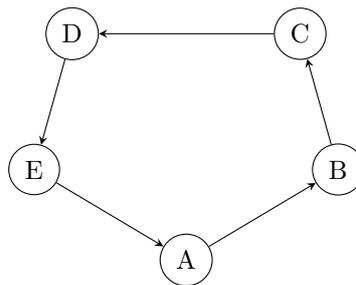
- `void` `envoie.donnees(int fd, std::vector<char> data)` : Cette fonction doit envoyer le contenu du vecteur `data` à travers le *file descriptor* `fd`. Cette fonction doit s'assurer que toutes les données sont envoyées.
- Comment allez-vous assurer un passage correcte des valeurs entre les processus.
- `std::vector<char>` `lire.donnees(int fd, int taille)` : Cette fonction lit les données sur le file descriptor `fd` et les renvoie sous la forme d'un vecteur. Cette fonction doit s'assurer de bien lire toutes les données. Elle doit donc se bloquer jusqu'à leur arrivée. Si le flux est fermé trop tôt, elle doit remplir les octets suivant de 0.

En cas d'erreur sur le flux, les 2 fonctions, doivent lancer une exception ou sortir du programme.

II Élections

Dans certaines circonstances, différents programmes doivent s'organiser pour résoudre un problème. En général, on procède à une élection, c'est à dire que chacun tire une valeur au hasard et celui qui a la plus forte devient le *leader*.

Vous devez effectuer les élections dans le cas d'un réseau de processus relié en anneau. Chacun peut envoyer une valeur à son voisin de droite et lire celle du voisin de gauche.



Q.II.1) - Vous devez proposer un algorithme qui permet aux processus de désigner un leader.

Q.II.2) - Écrire la fonction :

```
int election(int num, int nb_proc, int entree, int sortie);
```

Où `num` est le numéro du processus, `nb_proc` le nombre totale de processus, `entree` et `sortie` les *file descriptor* permettant de lire ce qu'envoie le voisin de gauche et d'écrire au voisin de droite. Pour cela chaque processus doit lire la valeur du précédent et l'envoyer au suivant 2 fois. Sauf le premier qui ne lit pas au début et le dernier qui n'envoie rien à la fin.