

PROGRAMMATION CONCURRENTE

CPE Lyon – 3ETI

TRAVAUX PRATIQUES SEANCE 3 - TUBES ANONYMES `os.pipe()`, `multiprocessing.Pipe()`

EXERCICE 1 – Introduction aux pipes anonymes

Réalisez un script montrant l'utilisation d'un tube anonyme (**pipe**) entre deux processus (voir cours).

EXERCICE 2 – Les redirections d'entrées/sorties

Testez une à une les commandes *shell* suivantes : **wc**, **sort**, **tail** (consultez le *man*).

Testez les redirections d'entrée (<), de sortie (>) et le pipe (|) à l'aide des commandes suivantes :

- ✓ **sort < fichier > fichier_trie**
- ✓ **sort fichier | grep toto | wc -l**

Ecrire 2 scripts Python réalisant respectivement les 2 commandes composées suivantes :

- ✓ **cat fichier | wc**
- ✓ **sort < fichier | grep chaine | tail -n 5 > sortie**

EXERCICE 3 – Utilisation des pipes anonymes

Développer un script Python composé de 3 processus réalisant les traitements suivants (voir schéma ci-dessous) :

1. Un premier processus, appelé **Générateur**, génère **N** nombres aléatoires positifs ou nuls. Si le nombre généré est **pair** (resp. **impair**) alors il est déposé dans le tube **NombresPairs** (resp. **NombresImpairs**) - à la fin de la génération, ce processus dépose la valeur **-1** dans les 2 tubes (pour indiquer la fin de la série). Ensuite récupère les deux nombres stockés respectivement dans le tube **SommePairs** et **SommeImpairs**, réalise leur somme et affiche le résultat.
2. Un deuxième processus, appelé **FiltrePair** est chargé de récupérer les nombres stockés dans le tube **NombresPairs**, de réaliser la somme de ces nombres et de déposer le résultat dans le tube **SommePairs**.
3. Un troisième processus, appelé **FiltreImpair** est chargé de récupérer les nombres stockés dans le tube **NombresImpairs**, de réaliser la somme de ces nombres et de déposer le résultat dans le tube **SommeImpairs**.

