

# Contrôles des E/S - Couplage mémoire

## Thèmes

- Utilisation des tubes
- Pipelines

**Ressources** Pour ce TP, comme pour les suivants, vous pourrez vous appuyer sur

- Le polycopié intitulé « Systèmes d'exploitation : Unix », qui fournit une référence généralement suffisante sur la sémantique et la syntaxe d'appel des différentes primitives de l'API Unix. Chaque section du sujet de TP indique la (ou les) section(s) du polycopié correspondant au contenu présenté.
- Les pages du manuel en ligne (commande `man`), et plus particulièrement les sections 2 et 3.

## 1 Minishell : Tubes et finalisation

Terminez les étapes ⑨ et ⑩ du TP5, si ce n'est pas déjà fait.

**Rendu final du Minishell** Vous devez rendre pour le **vendredi 24 mai 23h59** l'archive du Minishell. Cette archive devra comporter, en plus du code que vous avez réalisé, un document indiquant pour chaque étape les tests et leurs résultats dans un document PDF. Comme pour les TP précédents, l'archive sera réalisée à l'aide de la commande :

```
make archive
```

Le résultat est un fichier `minishell-nomutilisateur.tgz` à fournir sous Moodle à l'adresse <https://moodle-n7.inp-toulouse.fr/mod/assign/view.php?id=86287>.

## 2 Contrôle des E/S

Le but de l'exercice est de reprendre et mettre en œuvre les mécanismes de contrôle des entrées/sorties vues en TD. Nous allons donc considérer le code du programme `filtrage-squelette.c` fourni sous Moodle. Pour rappel, le programme exécute 2 processus :

- un processus fils qui lit un fichier et écrit le contenu dans un tube ;
- un processus père qui lit un caractère au clavier et effectue un traitement sur les données lues dans le tube. Le caractère est une commande : `'M'` met le texte en majuscule, `'m'` affiche le texte en minuscule, `'R'` affiche le contenu du texte à l'écran.

Dans la version du programme `filtrage-squelette.c`, le père alterne en séquence la saisie d'une commande et le traitement du texte reçu depuis le tube. Le but est alors d'utiliser d'adapter le programme fourni, afin de découpler la saisie des commandes et l'affichage du texte traité.

**Le code du programme** Téléchargez le code du programme `filtrage-squelette.c` sous Moodle. Compilez le et testez le.

**Création du programme `filtrage_fcntl.c` (Rendre les entrées/sorties non bloquantes)** Modifiez le code du programme `filtrage-squelette.c` de manière à rendre la lecture du caractères au clavier et la lecture dans le tube non bloquantes, en utilisant la primitive `fcntl` comme vu en TD.

**Création du programme `filtrage_select.c` (Attendre un ensemble de descripteurs de fichiers)**

Modifiez le code du programme `filtrage-squelette.c` de manière à attendre à la fois que les descripteurs de fichiers 0 et `p[0]` soient prêts en utilisant la primitive `select`. Nous considérerons que le délai d'attente n'est pas défini.

### 3 Couplage mémoire

Écrire un programme sous Linux, qui crée un fichier `tempo` contenant trois pages de caractères 'a', ouvre ce fichier en lecture/écriture, et crée un processus fils.

- Le processus père :
  - couple alors un segment de taille 3 pages à ce fichier en mode **partagé** et lecture/écriture,
  - attend 2 secondes,
  - remplit les pages 1 et 2 de caractères 'b',
  - attend 6 secondes,
  - affiche le premier caractère des pages 1 et 2,
  - remplit la page 2 de caractères 'c',
  - et termine.
- Le processus fils :
  - couple un segment de taille 3 pages au fichier `tempo` en mode **privé** et lecture/écriture,
  - affiche le premier caractère de la page 1,
  - attend 4 secondes,
  - affiche le premier caractère de chacune des pages 1, 2 et 3,
  - remplit la page 2 de caractères 'd',
  - attend 8 secondes,
  - affiche à nouveau le premier caractère de chacune des pages 1, 2 et 3,
  - et termine.

**Affichage du programme** Quel devrait (probablement) être l'affichage du programme? Vérifiez en l'exécutant.

**Contenu du fichier `tempo`** Quel est le contenu du fichier `tempo`?