

# Linux 2 : Appels Systèmes - BINV2181 (tp01- IO)

Tous les appels système doivent être testés. En cas d'erreur, un message adéquat doit être affiché et le programme arrêté. Pour faciliter ces tests, des fonctions « check » sont définies dans le module utilitaire `utils_v2` (fourni dans l'archive « LAS\_01\_exemple.zip » sur mooVin).

## 1. Comparaison entre `fwrite` et `write`

Question : Écrivez un programme C qui génère une suite d'entiers allant de 0 à 10 000 000 et qui les écrit dans un fichier binaire. Écrivez deux versions de ce programme : la première version en utilisant les fonctions `fopen()` et `fwrite()`, et la seconde version en utilisant les fonctions `open()` et `write()`. Toutes deux écrivent un entier à la fois dans le fichier. Mesurez le temps d'exécution de chaque version avec la commande `time` et comparez-les. Quelle version est plus rapide ? Pourquoi ?

## 2. Exercice sur les appels systèmes : `open` – `read` – `write` – `close`

Écrivez un programme `lireEtEcrire.c` qui

- reçoit deux arguments sur la ligne de commande qui sont deux noms de fichiers ;
  - lit à l'entrée standard des lignes d'au plus 80 caractères (`\n` non inclus) ;
  - écrit les lignes qui commencent par une majuscule dans le premier fichier, celles qui commencent par une minuscule dans le second et ignore les autres.
- Si le programme ne reçoit pas les deux arguments attendus, il se termine en affichant l'usage à la sortie d'erreur.
- Si un des fichiers dont le nom est passé sur la ligne de commande n'existe pas dans le répertoire courant, il est créé, sinon il est vidé de son contenu.
- Si une ligne lue comprend plus de 80 caractères, elle est ignorée et le buffer du clavier vidé.
- Si une ligne lue ne commence pas par un caractère alphabétique la ligne est ignorée.
- Toutes les entrées-sorties doivent être programmées grâce à des appels systèmes, à l'exception des éventuelles invites destinées à l'utilisateur, qui utilisent `printf()`, et des messages envoyés sur `stderr` en cas de plantage, qui utilisent `perror()`.

## Exemples d'exécution :

### > lireEtEcrire Maj min

Rien ne sert de courir ; il faut partir à point.  
Le Lièvre et la Tortue en sont un témoignage.  
Gageons, dit celle-ci, que vous n'atteindrez point  
si tôt que moi ce but. Si tôt ? Êtes-vous sage ?  
repartit l'Animal léger.  
Ma Commère, il vous faut purger  
avec quatre grains d'ellébore.  
Sage ou non, je parie encore.  
Ainsi fut fait : et de tous deux  
on mit près du but les enjeux.  
Savoir quoi, ce n'est pas l'affaire ;  
ni de quel juge l'on convint.  
Notre Lièvre n'avait que quatre pas à faire ;  
J'entends de ceux qu'il fait lorsque prêt d'être atteint  
il s'éloigne des Chiens, les renvoie aux calendes,  
et leur fait arpenter les landes.  
Ayant, dis-je, du temps de reste pour brouter,  
pour dormir, et pour écouter  
d'où vient le vent, il laisse la Tortue  
aller son train de Sénateur.  
Elle part, elle s'évertue ;  
elle se hâte avec lenteur.  
Lui cependant méprise une telle victoire ;  
tient la gageure à peu de gloire ;  
croit qu'il y va de son honneur  
de partir tard. Il broute, il se repose,  
il s'amuse à toute autre chose  
qu'à la gageure. À la fin, quand il vit  
que l'autre touchait presque au bout de la carrière,  
il partit comme un trait ; mais les élans qu'il fit  
furent vains : la Tortue arriva la première.  
Eh bien, lui cria-t-elle, avais-je pas raison ?  
De quoi vous sert votre vitesse ?  
Moi l'emporter ! et que serait-ce  
si vous portiez une maison ?

### > cat Maj

Rien ne sert de courir ; il faut partir à point.  
Le Lièvre et la Tortue en sont un témoignage.  
Gageons, dit celle-ci, que vous n'atteindrez point  
Ma Commère, il vous faut purger  
Sage ou non, je parie encore.  
Ainsi fut fait : et de tous deux  
Savoir quoi, ce n'est pas l'affaire ;  
Notre Lièvre n'avait que quatre pas à faire ;  
J'entends de ceux qu'il fait lorsque prêt d'être atteint  
Ayant, dis-je, du temps de reste pour brouter,  
Elle part, elle s'évertue ;  
Lui cependant méprise une telle victoire ;  
Eh bien, lui cria-t-elle, avais-je pas raison ?  
De quoi vous sert votre vitesse ?  
Moi l'emporter ! et que serait-ce

#### > cat min

si tôt que moi ce but. Si tôt ? Êtes-vous sage ?  
repartit l'Animal léger.  
avec quatre grains d'ellébore.  
on mit près du but les enjeux.  
ni de quel juge l'on convint.  
il s'éloigne des Chiens, les renvoie aux calendes,  
et leur fait arpenter les landes.  
pour dormir, et pour écouter  
d'où vient le vent, il laisse la Tortue  
aller son train de Sénateur.  
elle se hâte avec lenteur.  
tient la gageure à peu de gloire ;  
croit qu'il y va de son honneur  
de partir tard. Il broute, il se repose,  
il s'amuse à toute autre chose  
qu'à la gageure. À la fin, quand il vit  
que l'autre touchait presque au bout de la carrière,  
il partit comme un trait ; mais les élans qu'il fit  
furent vains : la Tortue arriva la première.  
si vous portiez une maison ?

#### > lireEtEcrire

Usage: lireEtEcrire fichier1 fichier2

### 3. Exercice sur les appels systèmes : read – write

Ecrivez un programme `lire.c` qui

- lit à l'entrée standard grâce à l'appel système `read` des chaînes de maximum 10 caractères (`\n` inclus) en affichant une invite à l'utilisateur avant chaque lecture ;
- affiche, après chaque lecture, la ligne lue à la sortie standard grâce à l'appel système `write` ;
- se termine lorsque l'utilisateur entre Ctrl-D.

- Les invites destinées à l'utilisateur peuvent être affichées via la fonction `printf`.
- Testez votre programme en introduisant successivement des chaînes de moins de 10 caractères.
- Poursuivez le test en introduisant successivement des chaînes de 9, 10, 11 et 15 caractères. Si votre programme a un comportement étrange, expliquez pourquoi.
- Si nécessaire, modifiez ensuite votre code pour que le programme avertisse l'utilisateur si la ligne fait plus de 10 caractères (`\n` inclus) et réclame d'introduire une nouvelle ligne dans ce cas.

### 4. Exercice avec redirection de l'entrée standard

Reprenez l'exercice 2 et testez-le en redirigeant l'entrée standard :

```
lireEtEcrire Maj min < fichierIn
```

Le résultat est-il celui qui est attendu ? Pourquoi ?

Comment faut-il modifier le programme pour qu'il produise le même effet, qu'il y ait redirection ou pas ?