# Linux: appels système(I2180)

*Tous les appels système doivent être testés, en cas d'erreur, un message adéquat doit être affiché et le programme arrêté (les macros ne peuvent pas être utilisées).*

1) Exercice préliminaire 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);
- affiche, après chaque lecture, la ligne lue à la sortie standard grâce à l'appel système `write`.

- 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.

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

Ecrivez 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 ;
- 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 messages envoyés sur *stderr* en cas de plantage, qui utilisent `perror(…)`.

Exemples d'exécution :

```
> lireEtEcrire Majuscule minuscule
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 Majuscule
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 minuscule
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) **[BONUS]** Exercice avec redirection de l'entrée standard

Reprenez l'exercice 1.B 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 ?

4) **[BONUS]** Exercice sur l'appel système : `stat`

Vous allez ajouter des fonctionnalités à votre programme `lireEtEcrire.c` obtenu à l'exercice 1.B en utilisant l'appel système `stat`.

Ces fonctionnalités sont :

1. Ne pas autoriser de rediriger l'entrée standard ;
2. En fin de programme, affichez le nom, la taille et les permissions des deux fichiers qui sont passés en arguments sur la ligne de commande.

Consultez le man de `stat`. Laquelle des différentes versions de cet appel système allez-vous utiliser ?

Indication pour la fonctionnalité 1 : c'est le « type » du fichier qui permet de savoir s'il s'agit d'un fichier classique (« block device ») ou bien d'informations en provenance du clavier (« character device »). Il vous faut donc trouver cette information et la tester pour stopper votre programme en cas de redirection d'un fichier vers le clavier.

Indication pour la fonctionnalité 2 : ce sont les 9 derniers bits du « mode » du fichier qui permettent de coder les 512 possibilités de permissions allouables à un fichier. Il vous faut donc trouver cette information et l'analyser pour en afficher le détail.

Remarque supplémentaire : si vous imprimez les informations concernant le « type » et le « mode » d'un fichier, par exemple pour tester votre programme, n'oubliez pas d'utiliser un format octal dans un souci de lisibilité.

Exemples d'exécution : voir page suivante.

Exemples d'exécution :

```
> lireEtEcrireStat un deux < fichierIn
Programme non concu pour lire un fichier, pas de redirection possible

> lireEtEcrireStat
Usage: lireEtEcrire fichier1 fichier2

> lireEtEcrireStat un deux
Bonjour
1 jour
la lune
      le soir
le soleil
Bonsoir
12334567890123345678901233456789012334567890123345678901233456789012334567890123345678901233456789012334567890123345
6789012334567890
derniere
Non derniere
CTRL D

un a une taille de 29 bytes et possede les permissions 100644
        user : rw-      group : r--     other : r--
deux a une taille de 27 bytes et possede les permissions 100644
        user : rw-      group : r--     other : r—

> cat un
Bonjour
Bonsoir
Non derniere

> cat deux
la lune
le soleil
derniere
```