

## Linux – Fiche 20

### Scripts – Exercices supplémentaires

1. Reprenez le script *smartfind.sh* (exercices 11-12) de la fiche 17-18-19. Améliorez-le de sorte qu'il vérifie qu'il y a bien deux arguments fournis sur la ligne de commande. Si ce n'est pas le cas, il rappelle l'usage du script sur *stderr* et se termine en renvoyant un code d'erreur.

Exemple :

```
./smartfind.sh
ERROR: invalid number of search criterion(s)
Usage:
./smartfind.sh directory extension [regex]
Exemple:
./smartfind.sh ~ c "\bprintf\b"
```

2. Ecrivez un script *man.sh* qui affiche les lignes du manuel de *man* contenant le mot « man » en les numérotant. Il affiche ensuite le nombre de lignes obtenues.
3. Ecrire un script *inverse.sh* qui demande à lire un texte sur l'entrée standard, ligne par ligne, jusqu'à la fin de l'entrée standard (Ctrl-D). Le script affiche ensuite le texte en inversant l'ordre des lignes.
4. Écrire un script *pair-impair.sh* qui reçoit deux noms de fichier sur la ligne de commande : f1 et f2. Il lit des lignes à l'entrée standard. Il écrit dans le fichier f1 toutes les lignes de numéro pair et dans le fichier f2 toutes les lignes de numéro impair.

Pour éviter de taper des lignes au clavier à chaque exécution de votre script, créez un fichier contenant quelques lignes de texte et redirigez-le vers l'entrée de votre script.

Exemple : `./pair-impair.sh pair impair < texte.txt`

5. Ecrivez un script *calc.sh* qui permet de calculer la somme, la différence, la division ou la multiplication de deux entiers. La commande respectera la syntaxe suivante :

```
./calc.sh op1 operateur op2
```

Le script affiche le calcul demandé et son résultat. Il doit afficher un message d'erreur et sa syntaxe d'utilisation puis se terminer si le nombre de paramètres n'est pas correct. Le script doit également afficher un message d'erreur si l'opérateur n'est ni +, -, / ou \*.

6. Ecrire un mini interpréteur "basich" qui affiche une invite de commande « Basich> » et réalisera des commandes entrées par l'utilisateur. Les opérations suivantes seront acceptées :
  - *ls* pour lister les fichiers en affichant leurs propriétés
  - *rename* pour renommer un fichier

- *del* pour effacer un fichier
- *nano* pour éditer le contenu d'un fichier en permettant le placement du curseur où vous cliquez dans le terminal (option *-m*)<sup>1</sup>
- *q* pour sortir de l'interpréteur

L'interpréteur affichera le message « `Command not found` » si la commande est inconnue et indiquera à l'utilisateur comment utiliser la commande lorsqu'il manque des paramètres (ex : `Usage: rename file-src file-dst`).

7. Ecrivez le script *reponse.sh* qui affiche chaque argument sur une ligne, suivi de "(O/N) ?" puis lit la réponse de l'utilisateur. Par exemple,

```
$ ./reponse.sh un deux trois quatre
un (O/N) ? O
deux (O/N) ? N
trois (O/N) ? N
quatre (O/N) ? O
```

Le script enregistre les arguments pour lesquels la réponse est "O" dans un fichier temporaire nommé *.reponses.tmp*, puis affiche le fichier temporaire et le supprime. Dans notre exemple on verra donc :

```
un
quatre
```

---

<sup>1</sup> Grâce au programme GPM (*General Purpose Mouse interface* ; installer avec la commande : `sudo apt install gpm`), la souris peut être utilisée dans la console. Dans nano, l'option *-m* active l'utilisation de la souris (cf. `man nano`). Un simple clic permet de placer le curseur et un double clic permet de placer une marque.