

Initiation Shell

Exercice 1

Nous allons créer une arborescence dans laquelle vous pourrez organiser et stocker votre travail. Nous allons travailler en ligne de commandes dans un terminal. La commande qui permet de créer un répertoire est `mkdir`. Son appel va créer un répertoire vide au niveau de l'arborescence auquel nous nous trouvons au moment de l'appel. Le nom du répertoire à créer est passé en paramètre à la commande `mkdir`.

1. Ouvrez un terminal.
2. Créer un répertoire `Methodologie`.
3. Créer un autre répertoire au même niveau que vous appellerez `ProgFonctionnelle`.

Exercice 2

Pour savoir où nous nous trouvons dans l'arborescence de fichiers, nous disposons de la commande `pwd`. Pour se déplacer dans l'arborescence, nous utilisons la commande `cd`. Pour savoir comment utiliser les différentes commandes, utilisez `man` suivi du nom de la commande. Chaque commande a donc son manuel. Vous choisirez l'éditeur de texte de votre choix (`gedit`, `emacs`, `vi`, `vim` ...).

1. Positionnez vous dans le répertoire `Methodo`.
2. Créez un fichier `premier` contenant le texte « premier fichier ».
3. Faites une copie de `premier` appelée `double`.
4. Comparez leurs tailles.
5. Renommez `double` en `introduction`.
6. Quelle différence y a-t-il entre `mv double introduction` et `cp double introduction`?

Exercice 3

1. Positionnez vous dans le répertoire `Methodo`.
2. Créez un répertoire `td1`.
3. Déplacez `introduction` dans `td1`.
4. Faites une copie de `premier` appelée `copie`, et placez-la également dans `td1`.
5. Affichez une liste de ce que contient `td1`.
6. Essayez de détruire `td1` en tapant `rmdir td1`. Que se passe-t-il? Que faut-il faire pour détruire un répertoire?
7. Détruisez tout ce que contient `td1`.
8. Détruisez `td1`.

Exercice 4

1. Comment visualiser deux fichiers l'un après l'autre à l'aide de la commande `cat`?
2. Comment faire une copie d'un fichier sans utiliser `cp` ni ouvrir d'éditeur de texte?
3. Utiliser `cat` pour écrire un peu de texte et mettre le résultat dans un fichier `test.txt`.
4. Quelle différence y a-t-il entre `cat tata toto > tutu` et `cat tata toto >> tutu`? Que se passe-t-il, selon que le fichier `tutu` existe ou n'existe pas?
5. Comment obtenir un fichier `toto` qui corresponde à un fichier `tata` dont les lignes seraient désormais numérotées?

Exercice 5

1. Comment chercher tous les fichiers commençant par un a majuscule ou minuscule, suivi éventuellement d'un ou plusieurs caractères, et terminant par un chiffre entre 3 et 6?

2. Comment fait-on pour indiquer que le fichier recherché a été modifié il y a plus de 30 jours? Il y a 30 jours? Il y a moins de 30 jours?
3. Comment faire chercher un fichier qui a été modifié plus récemment que le fichier `introduction`?
4. Comment fait-on pour afficher tous les répertoires de notre répertoire personnel?
5. Comment indiquer que le fichier recherché a une taille supérieure à une taille donnée?
6. Utilisez `find` pour effacer tous vos fichiers de sauvegarde (terminés par un tilde) qui ont plus d'un mois.

Exercice 6

Écrivez un script qui prend un paramètre et affiche la valeur de ce paramètre à l'écran. Par exemple, la commande `./monscript toto` doit afficher le paramètre : `toto`. Exécutez-le et vérifiez que le résultat est conforme.

Exercice 7

Écrire un script shell qui affiche le maximum de 2 valeurs passées en argument du script.

Exercice 8

Écrire un script shell qui affiche le maximum de 3 valeurs passées en argument du script.

Exercice 9

Écrire un script shell qui calcule x^3 , x étant passé en argument du script.

Écrire un script shell qui calcule x^5 , x étant passé en argument du script.

Écrire un script shell qui calcule x^n , x et n étant passés en argument du script.