

Systemes d'exploitation: exercices

Objectifs :

- Identifier les fonctions d'un système d'exploitation.
- **Utiliser les commandes de base en ligne de commande.**
- **Gérer les droits et permissions d'accès aux fichiers.**
- Les différences entre systèmes d'exploitation libres et propriétaires sont évoquées.

Exercice 1 :

Pour l'exercice on considère l'arborescence suivante :

Les noms de répertoires sont encadrés.

On suppose vous êtes sous la session de l'utilisateur **carole** donc le répertoire par défaut est : **/home/carole**

Partie A

Décrire l'effet de chacune des commandes suivantes, en supposant qu'elles sont exécutées les unes à la suite des autres.

1. `cd ~`

- Quel est le répertoire **courant** (ou se retrouve-t-on) ? Comment le vérifier ?

2. `mkdir NSI`

3. `mkdir NSI/TP_SHELL`

- Quelle est la nouvelle arborescence à partir du répertoire **carole** ?

4. `cd NSI/TP_SHELL`

5. `cd ..`

- Quel est alors le répertoire **courant** ?

6. `ls`

- Quel est le résultat suite à cette commande ? Quel est l'affichage ?

7. `chmod u+rwx,g-rwx,o-rwx TP_SHELL`

- Quel est le résultat de cette commande ? Quel est l'affichage ? Comment visualiser le résultat ? (aide : si besoin voir la question **b.** de cet exercice)

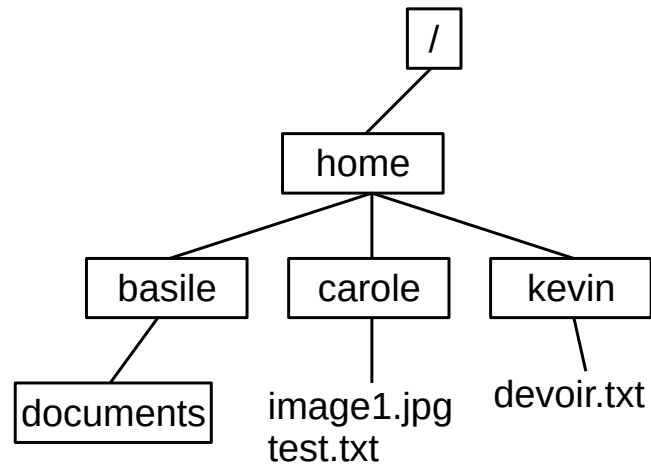
8. `touch devoir.txt`

9. `cp devoir.txt /home/carole/devoir.old`

- Quel est le résultat de cette commande ? Quel est l'affichage ?
- Quels fichiers sont dans le répertoire **carole** ? Même question dans le répertoire **NSI** ? Même question dans le répertoire **TP_SHELL** ?

10. `mv /home/carole/image1.jpg /home/basile`

- Schématiser l'arborescence finale des répertoires et fichiers à partir du répertoire **carole**.
- Dans l'arborescence totale combien de fichiers **devoir.txt** ?
- Dans quel cas la commande 10. se réalisera correctement ?



Partie B

L'instruction **chmod** permet de modifier les droits et permissions des fichiers et répertoires.

(les arguments d'une commande peuvent se suivre séparé par une virgule sans espace. Par exemple, la commande

```
chmod u-rwx,u+r chemin_répertoire_ou_fichier
```

supprime les droits r, w et x à l'utilisateur par défaut avant de (re)donner le droit r à l'utilisateur par défaut sur chemin_répertoire_ou_fichier)

Ainsi le répertoire NSI créé précédemment a par défaut les droits : drwxr-xr-x.

- Quelle est la commande pour lister (visualiser) les droits du répertoire NSI ?
- Que signifie le premier caractère d dans les droits de NSI ?
- Quels sont pour le répertoire NSI les droits de l'utilisateur par défaut ? Du groupe ? Des autres ?
- Quelle est la commande pour que les droits de NSI deviennent drw-r-----
- Quelle est la commande pour que les droits de NSI deviennent drwxrwxrwx

Partie C

Donner les commandes permettant de mettre les permissions demandées, quelles que soient les permissions initiales sur le fichier ou répertoire. On suppose que les répertoires NSI et NSI/TP_SHELL existent.

- Commande pour que le répertoire personnel possède tous les droits pour l'utilisateur et uniquement le droit d'exécution pour le groupe et les autres.
- Commande pour que les répertoires NSI et NSI/TP_SHELL possèdent tous les droits pour l'utilisateur et les droits de lecture/exécution pour le groupe et les autres (une commande par répertoire)
- Commande pour que le fichier lisible.txt du répertoire NSI/TP_SHELL (on suppose qu'il existe) possède les droits de lecture/écriture pour l'utilisateur et uniquement les droits de lectures pour le groupe et les autres.
- Commande pour que le fichier test.txt du répertoire par défaut possède les droits de lecture/écriture pour l'utilisateur et aucun droits pour le groupe et les autres.

Partie D

Quel est l'effet des commandes suivantes, quel seraient alors les droits affichés (10 premiers caractères) pour les fichiers et répertoires.

- `chmod o-rx,o+w ~/devoir.txt`
- `chmod u+wrx,g-wrx,o-wx,o+r ~/NSI`
- `chmod 777 test.txt`
- `chmod 025 /home/carole`

Exercice 2 :

On suppose que l'on se trouve dans un répertoire TEST, que ce dernier est vide et que l'on exécute les sept commandes suivantes.

- | | |
|--|--|
| <ol style="list-style-type: none"> <code>mkdir a b c d</code> <code>touch a/t.txt d/foo.txt</code> <code>cd c</code> <code>mkdir ../d/e f g</code> | <ol style="list-style-type: none"> <code>cd ..</code> <code>cp */*.txt c/g</code> <code>rm -rf d</code> |
|--|--|

- Dessiner l'arborescence** des fichiers et répertoires à partir du répertoire TEST après la ligne 4.
- Dessiner l'arborescence** des fichiers et répertoires à partir du répertoire TEST après la ligne 7.
(Aide: pour la ligne 6. cf complément au cours motif *glob* avant exercice 4)

Exercice 3 :

On suppose que votre répertoire par **défaut** est le répertoire : /home/utilisateur1 et qu'il est **vide**.

Partie A

Déterminer **les commandes** permettant de **réaliser** les actions suivantes (qui vont s'enchaîner):

- Quelle/s est/sont la/les commande/s pour obtenir le manuel ou de l'aide sur la commande mv ?
- Commande pour déterminer le répertoire **courant** (ici le répertoire par **défaut**) dans la hiérarchie des répertoires ?
- Commande pour lister le répertoire **courant**. (Y a t-il des fichiers, des répertoires dans ce répertoire ?)
- Commande pour lister le répertoires /bin ?
- Commande pour créer sous votre répertoire courant deux sous-répertoires : "source" et "Data" ?
- Commande pour se positionner sous "Source" ?
- Commande pour lister le répertoire courant ?
- Commande pour revenir sur le répertoire de départ ?
- Commande pour détruire "Source" ?
- Commande pour créer un fichier nommé "Mon_fichier.old" ?
- Commande pour créer un fichier nommé "Mon_fichier2" ?
- Commande pour créer un répertoire "old" ?
- Commande pour déplacer les fichiers avec l'extension old vers le répertoire "old" ? (Aide: voir encadré « compléments de cours » avant l'exercice 4)
- Commande pour copier les fichiers sans extension dans le répertoire "Data" ? (Aide: voir encadré « compléments de cours » avant l'exercice 4)
- Schématiser l'arborescence des répertoires et fichiers à partir du répertoire courant de départ (les répertoires seront encadrés)
- Commande pour déterminer la taille totale des fichiers contenus dans le répertoire courant ?
- Commande pour effacer tous les fichiers créés ?

Complément au cours :

Motifs glob

Dans une commande de la forme : utilisateur\$ commande arg1 arg2 ... argn

Les arguments arg1 arg2 ... argn peuvent contenir des caractères normaux et des caractères spéciaux : « * », « ? » et des ensembles de caractères entre crochet []. Les motif *glob* permettent de représenter avec une expression très compacte un ensemble de noms de fichiers potentiellement grand.

Par exemple le caractère « * » signifie « n'importe quelle séquence de caractères (potentiellement vide) ».

Ainsi dans le répertoire utilisateur, le motif *glob* Photos/img_*.jpg permet de lister d'un seul coup tous les noms de fichiers qui commencent par img_ et qui se terminent par .jpg, avec n'importe quelle suite de caractères entre les deux.

Le caractère « ? » est quand à lui remplacé par un seul caractère quelconque.

Le motif [123aE] correspond ici aux caractères 1 ou 2 ou 3 ou a ou E . (nb : dans une série logique on peut utiliser le caractère - , et le caractère ^ exclu des caractères :

Le motif [^5-9] exclu les entiers entre 5 et 9 inclus.

Exemples :

??? une chaîne de exactement 3 caractères quelconque (sauf vide).

[A-Z]* une chaîne d'au moins 1 caractère qui commence par une lettre majuscule.

?[^\0-9]? une chaîne de 3 caractères exactement, et dont le deuxième n'est pas un chiffre.

Exercice 4 :

Pour chacun des motifs *glob* ci-dessous, donner une suite de caractères (de longueur au moins un) reconnue par le motif.

1. *.jpg

2. [0-9]*[a-z]

3. *.???

4. *[^A-Z]*

5. ?????

6. ??*

7. *[^.]???

8. img??.???