

Systemes d'exploitation : exercices

Objectifs :

Identifier les fonctions d'un système d'exploitation.

Utiliser les commandes de base en ligne de commande.

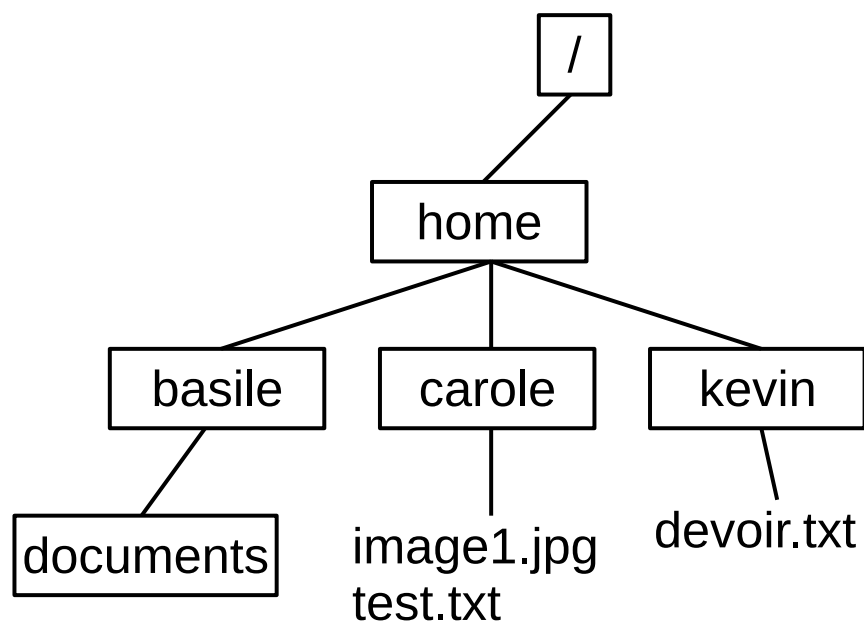
Gérer les droits et permissions d'accès aux fichiers.

Les différences entre systèmes d'exploitation libres et propriétaires sont évoquées.

Exercices

Ex 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 bien :

`/home/carole.`)

Partie A

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 ?

Fait du répertoire utilisateur le répertoire courant ici `/home/carole`.

On peut utiliser la commande **pwd** qui permet de savoir où se trouve le répertoire courant ici la commande `pwd` renvoie : **`/home/carole`**

2. `mkdir NSI`

3. `mkdir NSI/TP_SHELL`

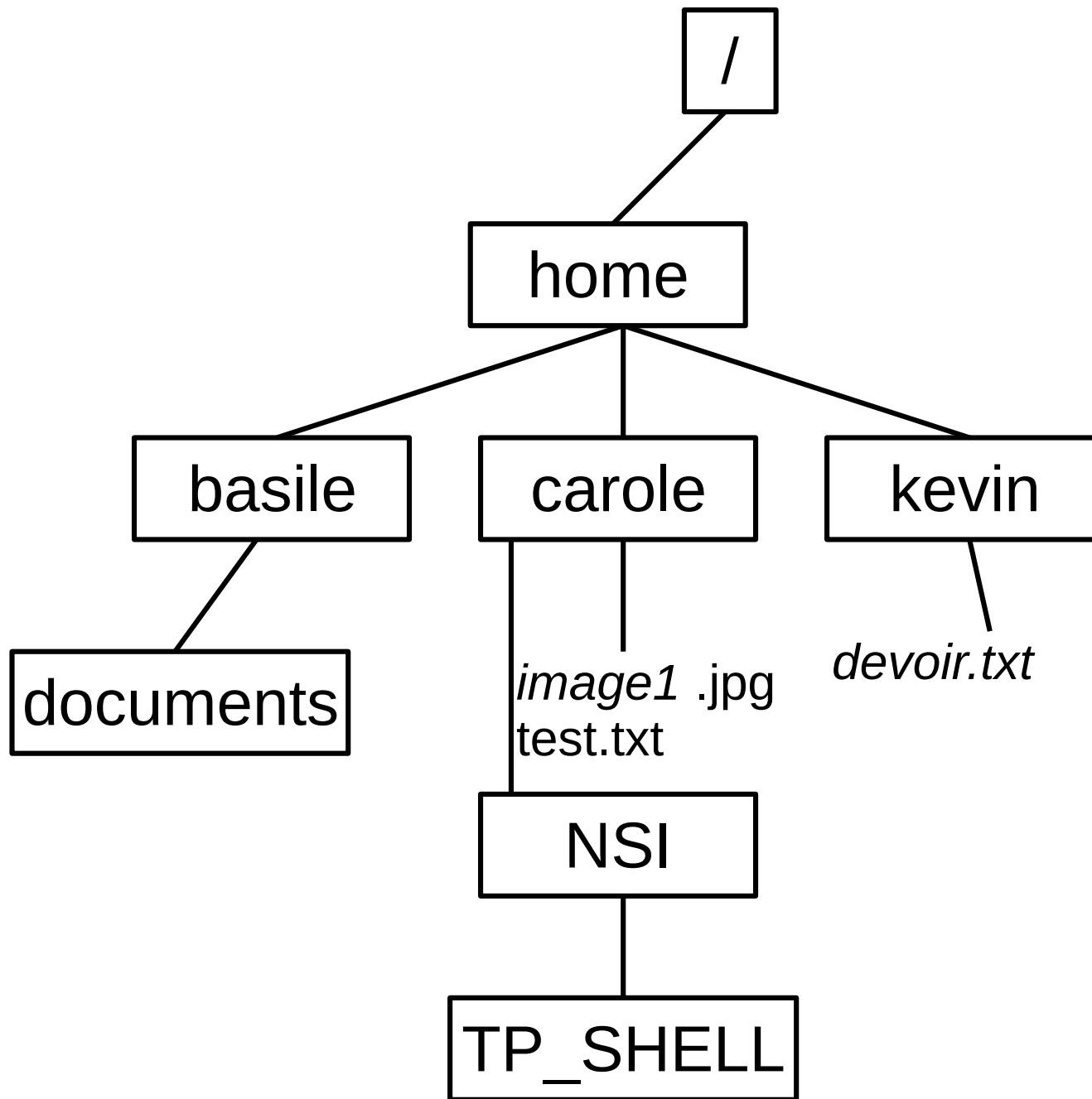
Quelle est la nouvelle arborescence à partir du répertoire carole ?

2. `mkdir NSI` ⇒ création du répertoire NSI dans le répertoire courant, qui est ici le répertoire utilisateur. (une erreur se produit si il existe déjà)

3. `mkdir NSI/TP_SHELL` ⇒ création du répertoire TP_SHELL dans le répertoire NSI à partir du répertoire courant. (une erreur se produit si il existe déjà ou si le répertoire NSI n'existe pas)

Quelle est la nouvelle arborescence à partir du répertoire carole ?

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 ?

4. `cd NSI/TP_SHELL` ⇒ `NSI/TP_SHELL` devient le répertoire courant.

5. `cd ..` ⇒ Le répertoire parent devient le répertoire courant donc ici le répertoire `NSI`

Quel est alors le répertoire courant ?

Le répertoire courant est bien `NSI`, donc chemin absolu : `/home/carole/NSI`

6. `ls`

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

⇒ affiche les répertoires et fichiers du répertoire courant.

L'affichage est :

`TP_SHELL`

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)

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)

Les droits du répertoire `TP_SHELL` sont modifiés : l'utilisateur a tous les droits `r`, `w` et `x`. Il devient donc accessible en lecture (`r`), écriture (`w`) et 'traversable' (`x`)(on peut rentrer dedans) par contre les autres n'ont plus aucun droits.

Attention cette commande s'exécute mais n'affiche pas le résultat (sauf si erreur..) rien ne s'affiche, le prompteur passe simplement à la ligne suivante.

Donc pour visualiser si la commande s'exécute 'comme on le désire' il faut a minima utiliser la commande `ls -l` sur le répertoire (ou fichier) dont on vient de modifier les droits.

Ici la commande `ls -l TP_SHELL`

le résultat :

```
drwx-----  reste de la ligne  TP_SHELL
```

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 dans le répertoire carole ? Même question dans le répertoire NSI ? Même question dans le répertoire TP_SHELL ?

8. `touch devoir.txt` ⇒ création du fichier vide `devoir.txt` dans le répertoire courant (ici c'est toujours le répertoire NSI)

9. `cp devoir.txt /home/carole/devoir.old` ⇒ copie du fichier `devoir.txt` du répertoire courant dans le répertoire `/home/carole` (chemin absolu)

Quel est le résultat de cette commande ? Quel est l'affichage ?

Comme pour la commande `chmod` aucun affichage particulier (sauf si erreur). Cette commande `cp` copie le fichier.

Quels fichiers dans le répertoire `carole` ? Même question dans le répertoire NSI ? Même question dans le répertoire `TP_SHELL` ?

Dans le répertoire `carole` on retrouve les fichiers : `images1.jpg` et `test.txt` et le fichier copie : `devoir.old`

Dans le répertoire NSI on trouve le fichier `devoir.txt`

Dans le répertoire `TP_SHELL` on ne trouve aucun fichier...

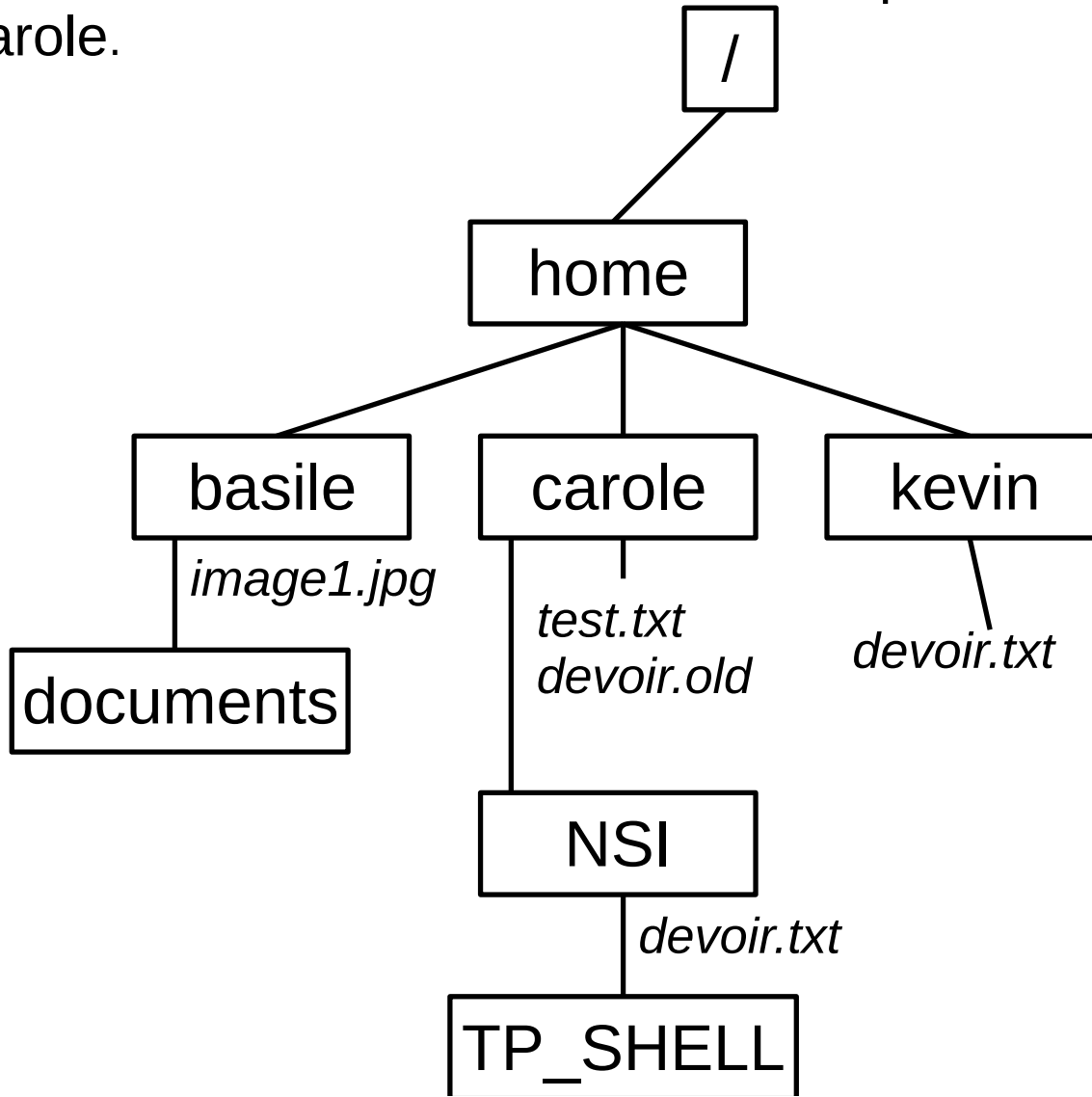
```
10. mv /home/carole/image1.jpg /home/basile
```

Schématiser l'arborescence finale des répertoires et fichiers à partir du répertoire carole.

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

La commande `mv` déplace le fichier `image1.jpg` du répertoire `carole` dans le répertoire `/home/basile` (chemin absolu)

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 ?

Dans l'arborescence totale combien de fichiers doivent avoir .txt ?

On trouve donc bien 2 fichiers doivent avoir .txt, mais qui n'ont aucun 'liens'.

Le fichier vide créé dans le répertoire NSI et le fichier déjà présent dans le répertoire kevin

Dans quel cas la commande 10. se réalisera correctement ?

Pour que la commande 10 : déplacement du fichier image1.jpg dans le répertoire basile, qui doit être le répertoire d'un autre utilisateur (fils de home) il est nécessaire que l'utilisateur, ici carole, ait les droits d'écriture dans le répertoire basile. Sinon message d'erreur indiquant que la commande ne peut pas être exécutée.

```
mv: can't stat '/home/carole/image1.jpg': Permission denied
```

```
mv: can't stat '/home/carole/image1.jpg':  
Permission denied
```

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 :
exemple :

```
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`)

[Rappels :](#)

schéma 1

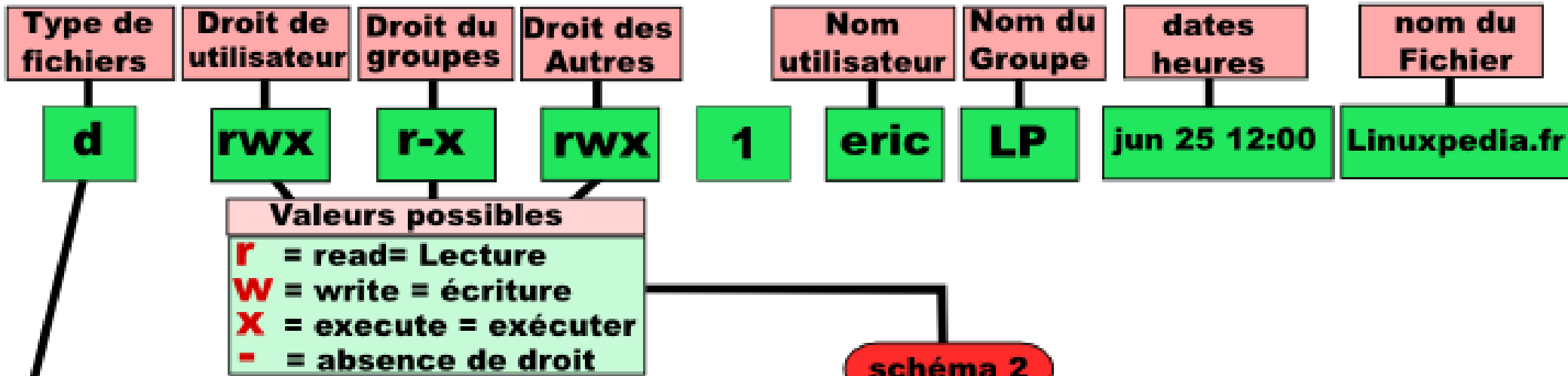


schéma 2

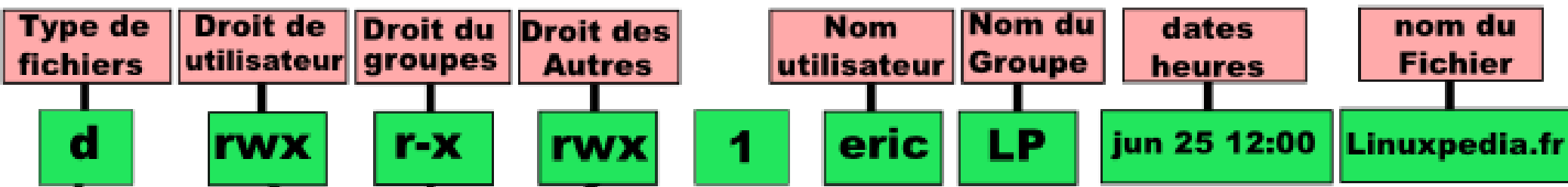
Conversion en valeur numérique

Droit de l'utilisateur			Droit du groupe			Droit des autres		
lecture	écriture	exécution	lecture	écriture	exécution	lecture	écriture	exécution
r	w	x	r	w	x	r	w	x
400	200	100	40	20	10	4	2	1

Valeurs possibles

- Fichier ordinaire
- d** Répertoire
- | lien symbolique
- B** périph caractère
- c** périphérique bloc
- P** tube nommé
- S** Socket locale

schéma 1



Valeurs possibles
r = read= Lecture
w = write = écriture
x = execute = exécuter
- = absence de droit

Valeurs possibles
 - Fichier ordinaire
d Répertoire
 | lien symbolique
B périph caractère
c périphérique bloc
P tube nommé
S Socket locale

schéma 2

Conversion en valeur numérique

Droit de l'utilisateur			Droit du groupe			Droit des autres		
lecture	écriture	exécution	lecture	écriture	exécution	lecture	écriture	exécution
r	w	x	r	w	x	r	w	x
400	200	100	40	20	10	4	2	1

sources: <https://www.linuxpedia.fr/lib/exe/fetch.php/droit.png>

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

a) Quelle est la commande pour lister (visualiser) les droits du répertoire NSI ?

Depuis le répertoire carole (par défaut) :

b) Que signifie le premier caractère `d` dans les droits de NSI ?

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

a) Quelle est la commande pour lister (visualiser) les droits du répertoire NSI ?

Depuis le répertoire carole (par défaut) :

`ls -l` : permet de lister les fichiers et répertoires et indique les droits

```
drwxr-xr-x 1 runner runner 0 May 2 19:19 NSI
```

b) Que signifie le premier caractère d dans les droits de NSI ?

d pour directory = répertoire

c) Quels sont pour le répertoire NSI les droits de l'utilisateur par défaut ? Du groupe ? Des autres ?

d) Quelle est la commande pour que les droits de NSI deviennent drw-r-----

c) Quels sont pour le répertoire NSI les droits de l'utilisateur par défaut ? Du groupe ? Des autres ?

Pour l'utilisateur : rwx droits de lecture d'écriture et d'exécution

Pour le groupe : r-x droits de lecture et d'exécution pas de droit d'écriture

Pour les autres: r-x droits de lecture et d'exécution pas de droit d'écriture

d) Quelle est la commande pour que les droits de NSI deviennent drw-r-----

```
chmod u-x,g-x,o-x ~/NSI
```

ou `chmod u-x,g-x,o-x NSI` (si dans le répertoire carole)

```
ou chmod 640 ~/NSI
```

```
ou chmod 640 NSI
```

d) Quelle est la commande pour que les droits de NSI deviennent drw-r-----

`chmod u-x,g-x,o-x ~/NSI`

ou `chmod u-x,g-x,o-x NSI` (si dans le répertoire carole)

ou `chmod 640 ~/NSI`

Ou `chmod 640 NSI`

e) Quelle est la commande pour que les droits de NSI deviennent drwxrwxrwx

d) Quelle est la commande pour que les droits de NSI deviennent drw-r-----

`chmod u-x,g-x,o-x ~/NSI`

ou `chmod u-x,g-x,o-x NSI` (si dans le répertoire carole)

ou `chmod 640 ~/NSI`

ou `chmod 640 NSI`

e) Quelle est la commande pour que les droits de NSI deviennent drwxrwxrwx

`chmod u+rwx,g+rwx,o+rwx ~/NSI`

ou `chmod u+rwx,g+rwx,o+rwx NSI`

`chmod 777 ~/NSI`

ou `chmod 777 NSI`

Partie C

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

a) 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.

b) 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)

Partie C

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

a) 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.

```
chmod u+rwx,g-rw,g+x,o-rw,o+x /home/carole  
      (chemin absolu)  
chmod 711 /home/carole
```

b) 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)

```
chmod u+rwx,g-w,g+rx,o-w,o+rx /home/carole/NSI  
      /home/carole/NSI/TP_SHELL
```

```
chmod 755 /home/carole/NSI /home/carole/NSI/TP_SHELL
```

c) 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.

d) 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.

c) 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.

```
chmod u+rw,u-x,g+r,g-wx,o+r,o-wx /home/carole/NSI/TP_SHELL/lisible.txt
```

```
chmod 644 /home/carole/NSI/TP_SHELL/lisible.txt
```

d) 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.

```
chmod u+rw,u-x,g-rwx,o-rwx test.txt
```

```
chmod 600 test.txt
```

Partie C

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

a) `chmod o-rx,o+w ~/devoir.txt`

b) `chmod u+rwx,g-rwx,o-wx,o+r ~/NSI`

c) `chmod 777 test.txt`

d) `chmod 025 /home/carole`

Partie C

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

a) `chmod o-rx,o+w ~/devoir.txt` `-rwxr-x-w-`

b) `chmod u+rwx,g-rwx,o-wx,o+r ~/NSI` `drwx---r--`

c) `chmod 777 test.txt` `-rwxrwxrwx`

d) `chmod 025 /home/carole` `d-----w-r-x`

Ex 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._

1. `mkdir a b c d`

2. `touch a/t.txt`
`d/foo.txt`

3. `cd c`

4. `mkdir ../d/e f g`

5. `cd ..`

6. `cp */*.txt c/g`

7. `rm -rf d`

Ex 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._

1. `mkdir a b c d`

2. `touch a/t.txt
d/foo.txt`

3. `cd c`

4. `mkdir ../d/e f g`

5. `cd ..`

6. `cp */*.txt c/g`

7. `rm -rf d`

```
1. mkdir a b c d
```

```
2. touch a/t.txt  
d/foo.txt
```

```
3. cd c
```

```
4. mkdir ../d/e f g
```

a) Dessiner l'arborescence des fichiers et répertoires à partir du répertoire TEST après la ligne 4.

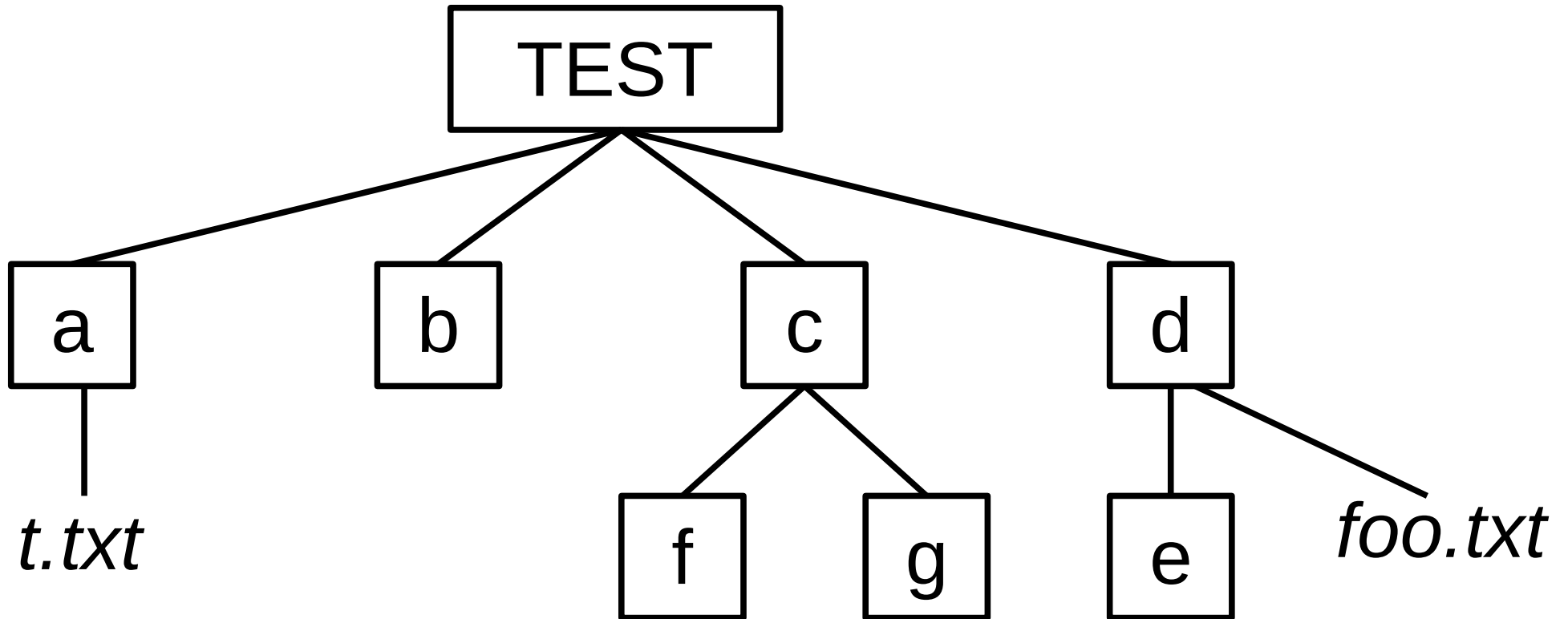

```
1. mkdir a b c d
```

```
2. touch a/t.txt  
d/foo.txt
```

```
3. cd c
```

```
4. mkdir ../d/e f g
```

a) Dessiner l'arborescence des fichiers et répertoires à partir du répertoire TEST après la ligne 4.



```
5. cd ..
```

```
6. cp */*.txt c/g
```

```
7. rm -rf d
```

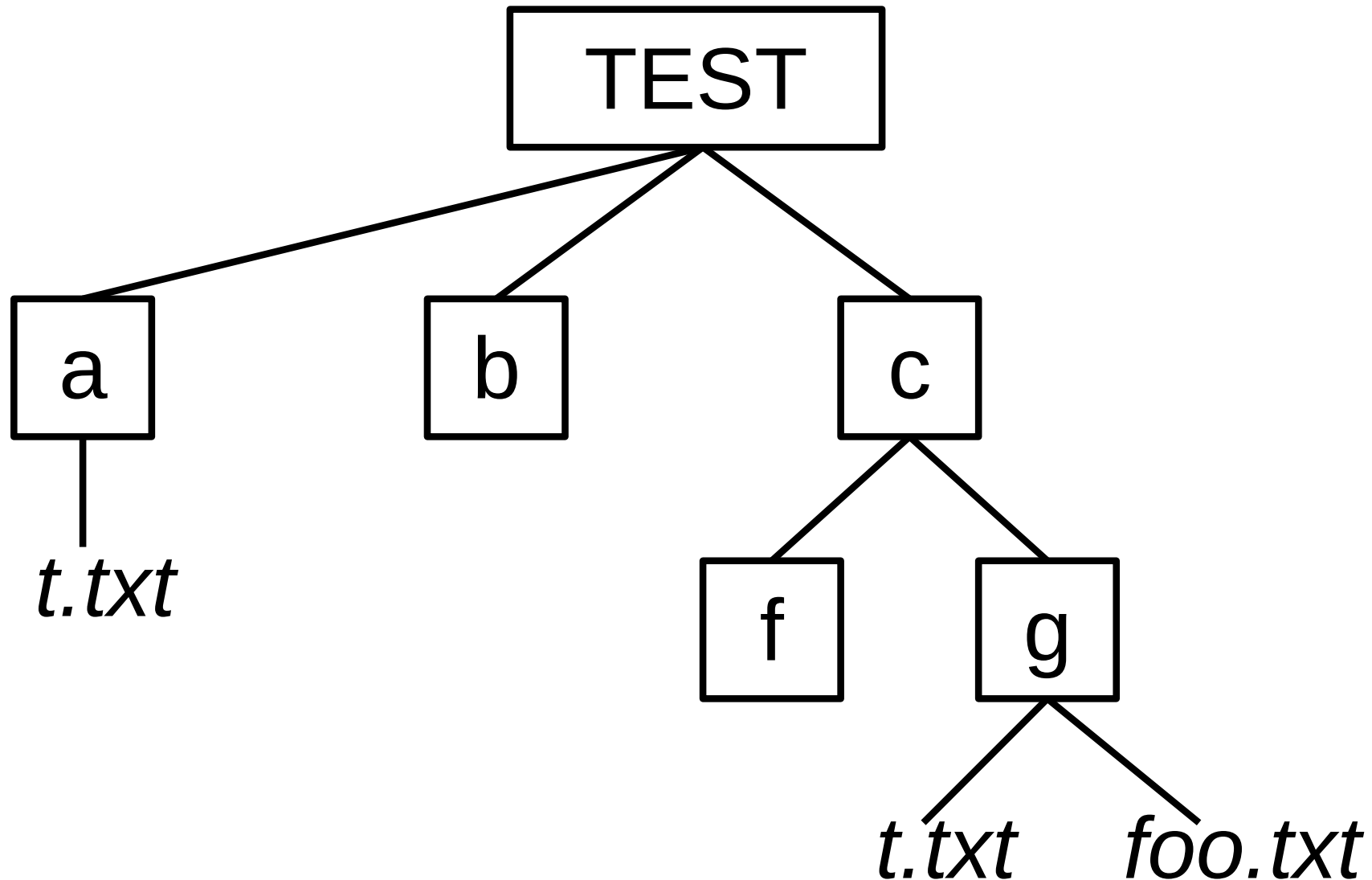
b) Dessiner l'arborescence des fichiers et répertoires à partir du répertoire TEST après la ligne 7.

```
5. cd ..
```

```
6. cp */*.txt c/g
```

```
7. rm -rf d
```

b) Dessiner l'arborescence des fichiers et répertoires à partir du répertoire TEST après la ligne 7.



Ex 3 :

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

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

a) Quelle/s est/sont la/les commande/s pour obtenir le manuel ou de l'aide sur la commande mv ?

■

b) Commande pour déterminer le répertoire courant (ici le répertoire par défaut) dans la hiérarchie des répertoires ?

Ex 3 :

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

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

a) Quelle/s est/sont la/les commande/s pour obtenir le manuel ou de l'aide sur la commande mv ?

. **man mv ou mv -h ou mv --help**

b) Commande pour déterminer le répertoire courant (ici le répertoire par défaut) dans la hiérarchie des répertoires ?

pwd

(permet de localiser le répertoire courant dans la hiérarchie)

b) Commande pour déterminer le répertoire courant (ici le répertoire par défaut) dans la hiérarchie des répertoires ?

pwd

(permet de localiser le répertoire courant dans la hiérarchie)

c) Commande pour lister le répertoire courant. (Y a-t-il des fichiers, des répertoires dans ce répertoire ?)

d) Commande pour lister le répertoires /bin.

e) Commande pour créer sous votre répertoire courant deux sous-répertoires : "Source" et "Data".

b) Commande pour déterminer le répertoire courant (ici le répertoire par défaut) dans la hiérarchie des répertoires ?

pwd

(permet de localiser le répertoire courant dans la hiérarchie)

c) Commande pour lister le répertoire courant. (Y a-t-il des fichiers, des répertoires dans ce répertoire ?)

ls (ou ls -l ou ls -a pour voir tous les fichiers et répertoire) aucun fichier ou répertoire ici (sauf les fichiers systèmes)

d) Commande pour lister le répertoires /bin.

ls /bin

e) Commande pour créer sous votre répertoire courant deux sous-répertoires : "Source" et "Data".

mkdir Source Data

e) Commande pour créer sous votre répertoire courant deux sous-répertoires : "Source" et "Data".

```
mkdir Source Data
```

f) Commande pour se positionner sous "Source".

g) Commande pour lister le répertoire courant.

h) Commande pour revenir sous le répertoire de départ.

i) Commande pour détruire "Source".

e) Commande pour créer sous votre répertoire courant deux sous-répertoires : "Source" et "Data".

```
mkdir Source Data
```

f) Commande pour se positionner sous "Source".

```
cd Source
```

g) Commande pour lister le répertoire courant.

```
ls (ou ls -l ou ls -a)
```

h) Commande pour revenir sous le répertoire de départ.

```
cd ..
```

i) Commande pour détruire "Source".

```
rmdir Source (suppression répertoire vide)
```

ou

```
rm -rf (suppression de répertoire et de tout son contenu)
```

h) Commande pour revenir sous le répertoire de départ. `cd ..`

i) Commande pour détruire "Source". `rmdir Source`

j) Commande pour créer un fichier nommé "Mon_fichier.old".

k) Commande pour créer un fichier nommé "Mon_fichier2"

l) Commande pour créer un répertoire "old".

m) Commande pour déplacer les fichiers avec l'extension .old vers le répertoire "old".

i) Commande pour détruire "Source". `rmdir Source`

j) Commande pour créer un fichier nommé "Mon_fichier.old".

`touch Mon_fichier.old`

k) Commande pour créer un fichier nommé "Mon_fichier2"

`touch Mon_fichier2`

l) Commande pour créer un répertoire "old".

`mkdir old`

m) Commande pour déplacer les fichiers avec l'extension .old vers le répertoire "old".

`mv *.old old`

ou `mv *.old ~/old` (chemin relatif)

ou `mv *.old /home/utilisateur1/old` (chemin absolu)

l) Commande pour créer un répertoire "old".

```
mkdir old
```

m) Commande pour déplacer les fichiers avec l'extension .old vers le répertoire "old".

```
mv *.old old
```

n) Commande pour copiez les fichiers sans extension dans le répertoire "Data".

o) Schématiser l'arborescence des répertoires et fichiers à partir du répertoire courant de départ.

l) Commande pour créer un répertoire "old".

```
mkdir old
```

m) Commande pour déplacer les fichiers avec l'extension .old vers le répertoire "old".

```
mv *.old old
```

n) Commande pour copiez les fichiers sans extension dans le répertoire "Data".

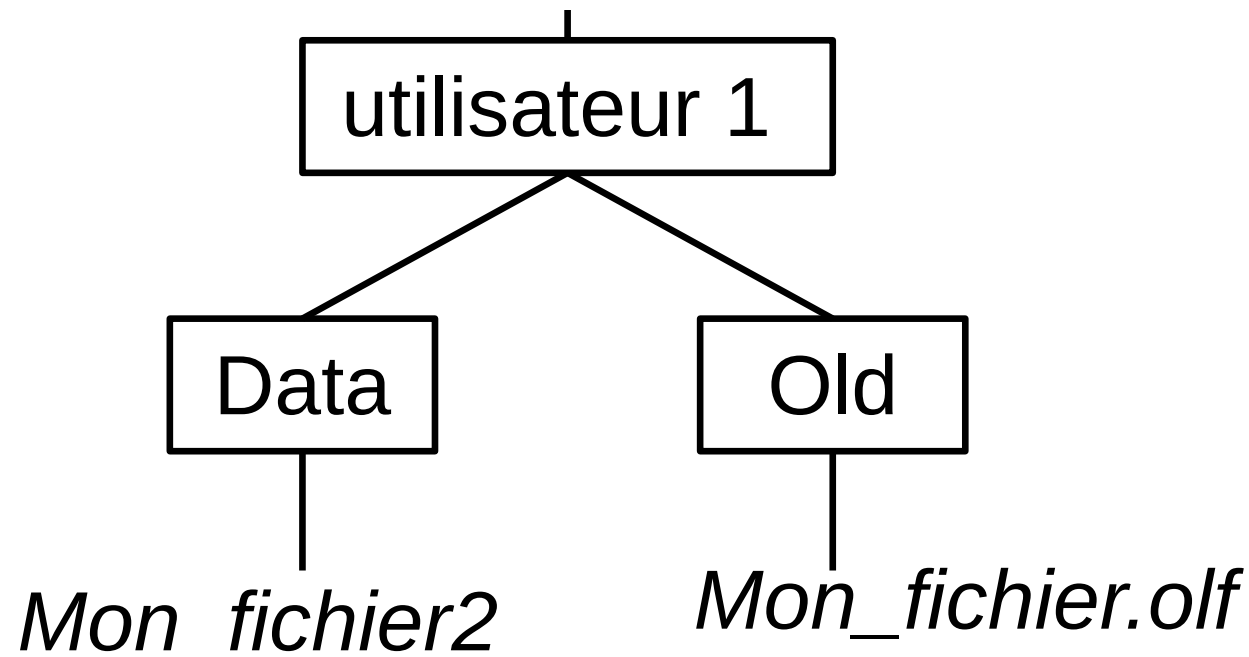
```
mv * Data car il ne reste qu'un fichier sans extension  
(ou mv *[^.]* Data qui ne sélectionne que  
les fichiers sans . donc sans extension)
```

o) Schématiser l'arborescence des répertoires et fichiers à partir du répertoire courant de départ.

n) Commande pour copiez les fichiers sans extension dans le répertoire "Data".

`mv * Data` car il ne reste qu'un fichier sans extension
(ou `mv *[^.]* Data` qui ne sélectionne que les fichiers sans . donc sans extension)

o) Schématiser l'arborescence des répertoires et fichiers à partir du répertoire courant de départ.



p) Commande pour déterminer la taille totale des fichiers contenus dans le répertoire courant ?

q) Commande pour effacer tous les fichiers créés.

p) Commande pour déterminer la taille totale des fichiers contenus dans le répertoire courant ?

`ls -l` (ou `ls -la`) mais il faut additionner les valeurs

ou la commande : `du -b disk usage`, « utilisation du disque » qui permet de connaître l'espace utilisé .

q) Commande pour effacer tous les fichiers créés.

`rm -r *`

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 motifs 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.

Ex 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] *

1. *.jpg

Une chaîne quelconque se terminant par « .jpg »
ex :

hello.jpg ; 15.jpg ; 1rf5A.jpg

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

Une chaîne commençant par un chiffre et se terminant par une minuscule ex :

2c ; 5ARD.PRa_e5.5.g ; 6_R8a

3. *.???

Une chaîne se terminant par « . » suivi d'exactly 3 caractères ex :

.5Az ; 84AEX.F5R ; hi....

4. *[^A-Z]*

Une chaîne contenant au moins un caractère qui n'est pas une majuscule ex :

8 ; a ; ABCD_FE ;

5. ??????

6. ?*?

7. *[^.]???

8. img??.???

5. ??????

Une chaîne quelconque contenant exactement 5 caractères ex :

5A.jpg ; 12345 ; hello ; 1A6.t

6. ?*?

Une chaîne quelconque contenant au moins 2 caractères ex :

5A.jpg ; 12345 ; he ; 1A

7. *[^.]????

Une chaîne contenant au moins 4 caractères et dont le quatrième à partir de la fin n'est pas un point . ex :

5Apg ; _345 ; hello_ ; Ae.r.4p

8. img??.????

Une chaîne qui commence par img suivi de deux caractères quelconques, d'un point « . » et ensuite de 3 caractères quelconques. ex :

img25.jpg ; imgAF.bmp ; img_.._A7