

II. Commandes de base en Linux

1) Terminal

Quelle que soit la distribution Linux utilisée, on trouve une application appelée "Terminal" ou "Console". Jusqu'en 1996 (!) et avant l'émergence d'un système d'affichage graphique comparable à MacOS/Windows, le terminal était la seule et unique manière de naviguer entre les applications. Ce n'est plus vrai aujourd'hui. Toutefois, **maitriser le terminal est essentiel**.

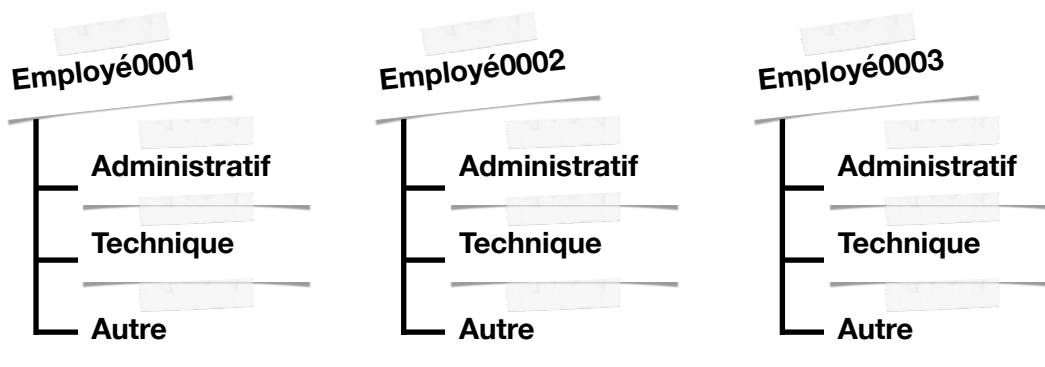
Le terminal permet :

- ❖ d'accéder à des fichiers système "invisibles" ;
- ❖ de déboguer des situations complexes (redémarrage...) ;
- ❖ de se connecter à des serveurs distants ;
- ❖ d'automatiser des tâches qui prendraient énormément de temps à la main grâce à des instructions appelées "lignes de commande".

Rem : Les lignes de commande sont interprétées par un *Shell*. Le plus courant est le Bash (Bourne Again Shell).

Exemple :

Dans une entreprise de 400 salariés, on veut créer un dossier *EmployéXXXX* et chaque dossier doit contenir les sous-dossiers *Administratif*, *Technique* et *Autre*. Et on veut faire cela pour chaque employé avec XXXX qui est l'identifiant du salarié.

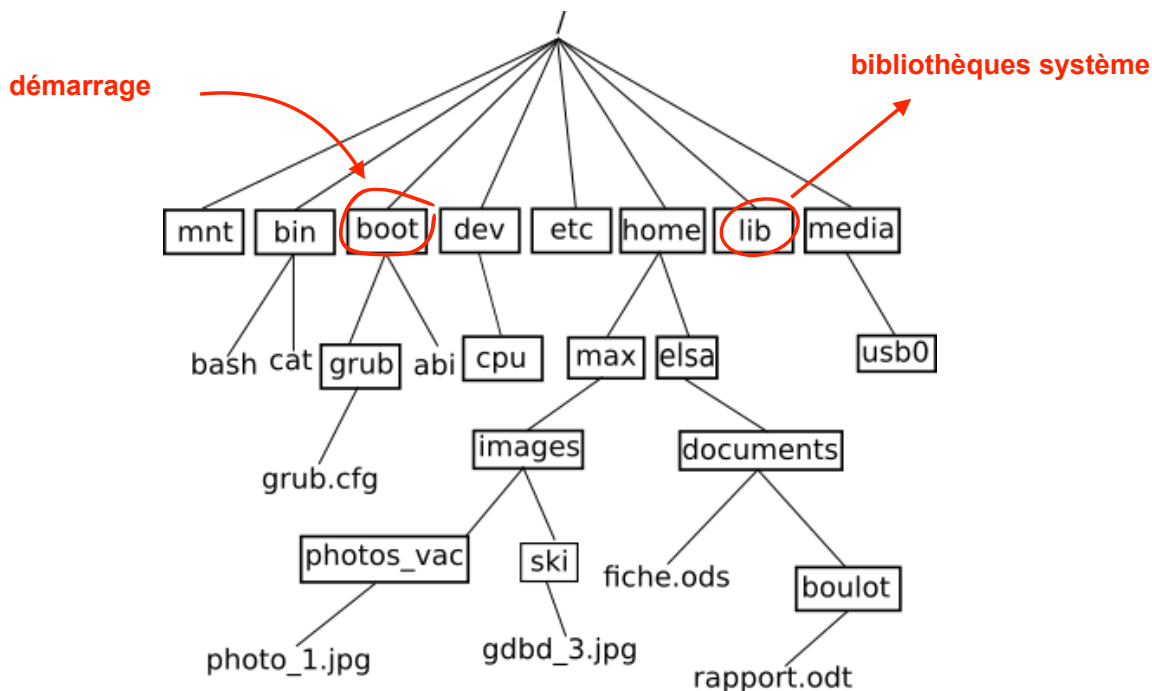


La secrétaire, sur son Windows, ferait sans doute tout cela à la main. et prendrait environ une semaine pour compléter cette tâche... Avec le Terminal, on peut faire cela en 2 minutes (le temps de trouver et d'écrire les lignes de commande).

2) Down the Rabbit Hole : l'arborescence

a. Introduction

Grâce à la ligne de commande, nous allons travailler sur les **fichiers** et les **répertoires**. Dans les systèmes de type "UNIX" (par exemple GNU/Linux ou macOS), nous avons un système de fichier en arborescence comme indiqué ci-dessous :



On parle d'arborescence, car ce système de fichier ressemble à un arbre à l'envers.

Définition :
.....

Dans le schéma ci-dessus on trouve des répertoires (noms entourés d'un rectangle comme "home") et des fichiers (uniquement des noms "grub.cfg").

Rem : les extensions des noms de fichiers (le "cfg" de "grub.cfg") ne sont pas obligatoires dans les systèmes "UNIX". Par exemple, "bash" est bien un nom de fichier et il n'a pas d'extension.

b. Chemin absolu et chemin relatif

Pour indiquer la position d'un fichier (ou d'un répertoire) dans l'arborescence, il existe 2 méthodes : indiquer un chemin absolu ou indiquer un chemin relatif.

Le chemin absolu indique le "chemin" depuis la racine. Par exemple, le chemin absolu du fichier fiche.ods sera : /home/elsa/documents/fiche.ods

Rem : nous démarrons bien de la racine "/" .

Exercice 1 :

Quel est le chemin absolu du dossier usb0 ?

Celui de bash ?

Il est possible d'indiquer le chemin non pas depuis la racine, mais depuis un répertoire quelconque, nous parlerons alors de chemin relatif. Le chemin relatif permettant d'accéder au fichier "photo_1.jpg" depuis le répertoire "max" est : "images/photo_vac/photo_1.jpg"

Rem : Le "/" au début du chemin est absent dans un chemin relatif. Cela nous permettra de distinguer un chemin relatif d'un chemin absolu.

Exercice 2 :

On est dans boot. Quel est le chemin relatif du fichier grub.cfg ?

À présent, nous désirons indiquer le chemin relatif pour accéder au fichier "gdbd_3.jpg" depuis le répertoire "photos_vac". Quel serait le problème ?

.....
.....

Pour ce faire il faut utiliser 2 points : "../ski/gdbd_3.jpg"

Rem : Il est tout à fait possible de remonter de plusieurs "crans" : "../.." depuis le répertoire "photos_vac" permet de "remonter" dans le répertoire "max".

Exercice 3 :

En vous basant sur l'arborescence ci-dessus, déterminez le chemin **absolu** permettant d'accéder au fichier :

- ❖ "bash"
- ❖ "rapport.odt"

Toujours en vous basant sur l'arborescence ci-dessus, déterminez le chemin **relatif** permettant d'accéder au fichier :

- ❖ "rapport.odt" depuis le répertoire "elsa"
- ❖ "fiche.ods" depuis le répertoire "boulot"
- ❖ "gdbd_3.jpg" depuis "boulot"

Rem : les systèmes "UNIX" sont "sensibles à la casse" (majuscules ≠ minuscules) : le répertoire "aBBa" et le répertoire "aBbA" sont deux répertoires différents.

3) Découverte des commandes de base

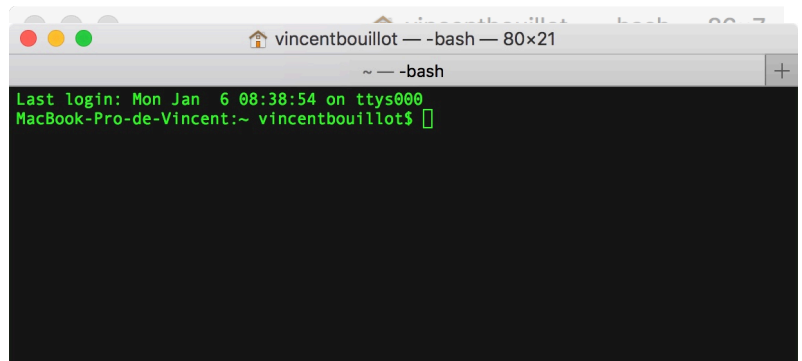
Il existe de très nombreuses commandes interprétables par Bash et celles-ci ont un nombre encore plus important d'options possibles ! Pour consulter le manuel d'une commande, on tape :

Rem : Pour quitter la page de documentation, on appuie sur q.

Téléchargez le fichier C10_OS.zip à partir de bouillotvincent.github.io et dézipper le sur votre Bureau.

Allez sur CoCalc.com. Choisissez "New" puis "_Linux Terminal". Vous devriez avoir quelque chose qui ressemble vaguement à cela :

Nous avons ci-contre le terminal de l'utilisateur "vincentbouillot" qui utilise un ordinateur qui se nomme "MacBook-Pro-de-Vincent".

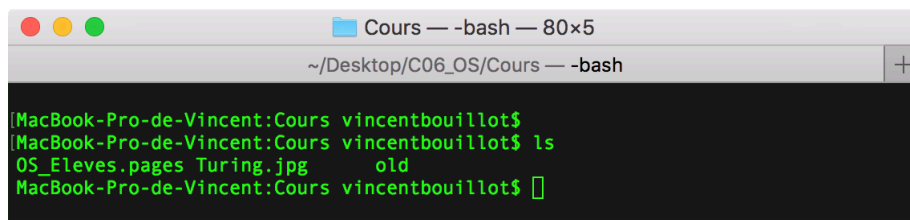


```
vincentbouillot — -bash — 80x21
~ — -bash
Last login: Mon Jan 6 08:38:54 on ttys000
MacBook-Pro-de-Vincent:~ vincentbouillot$
```

La commande ls

À l'aide de "man ls", dire ce que permet de faire la commande ls :

.....



```
Cours — -bash — 80x5
~/Desktop/C06_OS/Cours — -bash
[MacBook-Pro-de-Vincent:Cours vincentbouillot$
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ ls
OS_Eleves.pages Turing.jpg      old
MacBook-Pro-de-Vincent:Cours vincentbouillot$
```

Dans l'exemple ci-dessus, dans le répertoire "Cours" de l'utilisateur "vincentbouillot", nous affichons le contenu du répertoire courant. Dans ce répertoire, il y a un fichier .pages, un fichier .jpg et un répertoire old.

Sur CoCalc, utilisez la commande ls dans votre "répertoire personnel". Listez les fichiers et répertoires qui vous sont donnés par ls :

.....

.....

Options intéressantes avec ls :

.....

La commande `cd`

La commande "`cd`" permet de changer de répertoire courant et donc de naviguer dans l'arborescence. Il suffit d'indiquer le chemin (relatif ou absolu) qui permet d'atteindre le nouveau répertoire : `cd chemin/vers/le/répertoire/voulu`

Exercice 4 :

En utilisant des `ls` et des `cd`, déplacez-vous jusqu'à atteindre le sous-dossier "Cours" du dossier "C10_OS". Quelle est la succession de commandes que vous avez tapé ?

.....

.....

Comment feriez-vous pour aller dans Activités ? puis osPython ?

.....

Revenez à votre répertoire personnel. Quelle commande peut être utilisée ?

.....

La commande `pwd`

À l'aide de "`man pwd`", dire ce que permet de faire la commande `pwd` :

.....

Retournez dans Cours. Quel est le chemin absolu du répertoire "Cours" dans lequel vous vous situez ?

.....

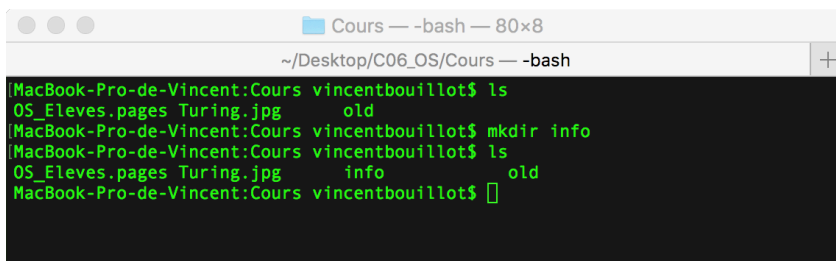
En déduire le chemin relatif du répertoire "Cours" depuis votre répertoire "home".

.....

Rem : Si vous êtes perdu dans votre arborescence, "`pwd`" permet de savoir où vous êtes !

La commande `mkdir`

La commande "`mkdir`" permet de créer un répertoire dans le répertoire courant. La commande est de la forme :
"`mkdir nom_du_répertoire`"



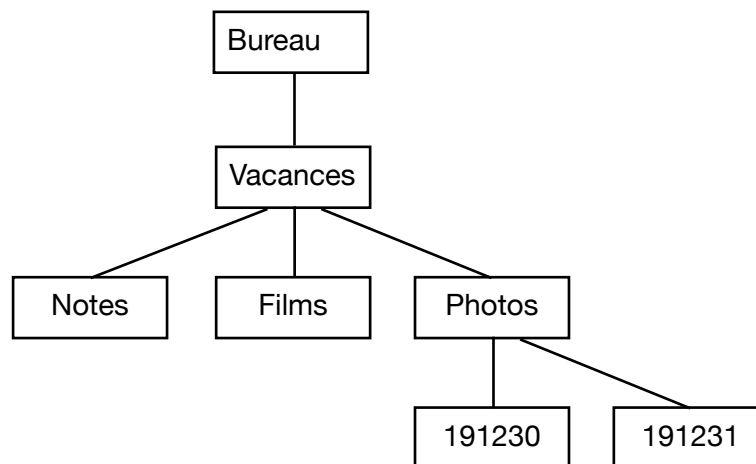
```
MacBook-Pro-de-Vincent:~$ cd ~/Desktop/C06_OS/Cours
MacBook-Pro-de-Vincent:~/Desktop/C06_OS/Cours$ ls
OS_Eleves.pages  Turing.jpg      old
MacBook-Pro-de-Vincent:~/Desktop/C06_OS/Cours$ mkdir info
MacBook-Pro-de-Vincent:~/Desktop/C06_OS/Cours$ ls
OS_Eleves.pages  Turing.jpg      info      old
MacBook-Pro-de-Vincent:~/Desktop/C06_OS/Cours$
```

Déplacez-vous dans le dossier "Cours" et créez un répertoire appelé "info". Un ls doit vous renvoyer la fenêtre de la page 8.

Rem : il est préférable de ne pas utiliser de caractères accentués dans les noms de répertoire ou de fichier. De même, remplacez tous les espaces par des underscore "_".

Exercice 5 :

Revenez sur votre répertoire personnel à l'aide de `cd ..` (ou `cd ~` ou `cd`) et grâce aux lignes de commande, créez une petite arborescence correspondant à celle ci-dessous :



La commande rm

La commande "rm" permet de supprimer un fichier ou un répertoire. La commande est de la forme "rm nom_du_répertoire_ou_nom_du_fichier".

Retourner dans le répertoire "C10_OS/Cours" et à l'aide de la commande rm, supprimez le fichier Turing.jpg. À l'aide d'un ls, vérifiez que votre fichier est bien supprimé :

```
Cours — -bash — 80x8
~/Desktop/C06_OS/Cours — -bash
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ ls
OS_Eleves.pages Turing.jpg info old
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ rm Turing.jpg
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ ls
OS_Eleves.pages info old
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ ]
```

À l'aide de la commande rm, supprimez le répertoire info. Que se passe-t-il ?

La plupart des commandes UNIX peuvent être utilisées avec une ou plusieurs options. Par exemple, pour supprimer un répertoire, il est nécessaire d'utiliser la commande "rm" avec l'option "-r" : "rm -r nom_du_répertoire". On peut aussi faire **rmdir**.

Supprimez à présent le répertoire "info".

Exercice 6 :

En utilisant l'instruction "**cd ~**" ou "**cd**", retournez sur votre home en une instruction. Puis, en une instruction également, retournez dans le répertoire "Bureau", "Vacances" puis "Photo". **On peut appuyer sur la touche tabulation pour afficher les répertoires potentiellement accessibles via cd.**

Grâce à l'instruction rm, supprimez le répertoire "191231" : les photos du Réveillon, c'est trop dangereux...

La commande touch

La commande "touch" permet de créer un fichier vide. La commande est de la forme "touch nom_du_fichier_à_créer". Elle permet aussi de changer la **date de dernière modification** d'un fichier (ca commence à devenir intéressant...).

Retourner dans le répertoire "C10_OS/Cours" (ca devrait commencer à rentrer...) et créez un fichier vide "toto.jpg" .

Effectuer la commande "ls -l". Émettez une hypothèse sur ce à quoi correspondent toutes ces colonnes ?

.....

.....

Faire un touch sur OS_Eleves.pages, puis refaire la commande "ls -l". Que constatez-vous ?

.....

.....

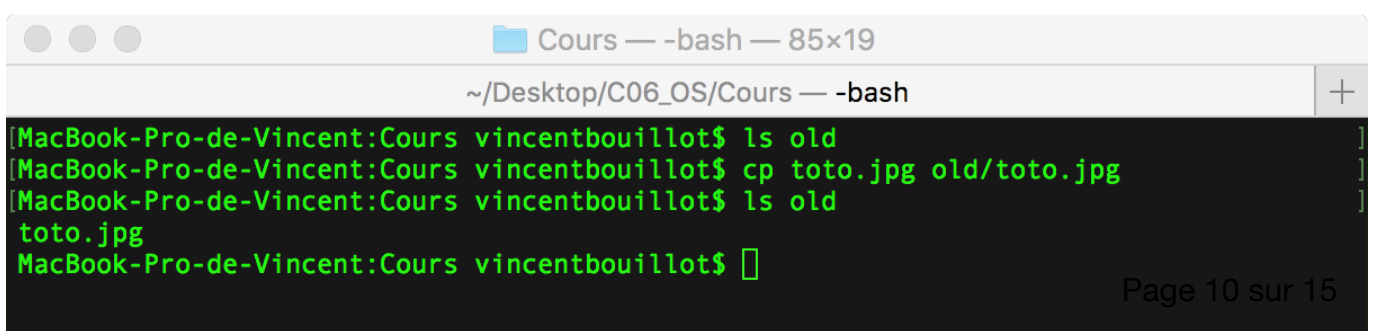
Alors, à quoi correspondent toutes ces colonnes ?

.....

.....

La commande cp

La commande "cp" permet de copier un fichier. La commande est de la forme :
"cp répertoire_source/nom_fichier_à_copier répertoire_destination/nom_fichier"



```
Cours — -bash — 85x19
~/Desktop/C06_OS/Cours — -bash
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ ls old
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ cp toto.jpg old/toto.jpg
[MacBook-Pro-de-Vincent:Cours vincentbouillot$ ls old
toto.jpg
MacBook-Pro-de-Vincent:Cours vincentbouillot$
```

Rem : le nom du fichier "destination" n'est pas obligatoirement le même que le nom du fichier "source" (on peut avoir "cp toto.jpg info/avengers.avi")

Exercice 7 :

Copiez OS_Eleves.pages et renommez-le Transformers.mkv

Exercice 8 : La secrétaire en galère

Reprenons notre exemple initial de la secrétaire qui doit **créer** un dossier EmployeXXXX pour les 400 employés de l'entreprise, chaque dossier **devant contenir** 3 sous-dossiers : Administratif, Technique, Autre. Les salariés sont représentés par des numéros allant de 0001 à 0400 (on pourrait disposer d'une liste avec leurs noms, ce serait exactement pareil). Pour générer cette arborescence, on va utiliser un **script bash**, qui sera **exécutable**.

Déplacez-vous dans Activités : quels fichiers trouvez-vous ?

.....
À l'aide de "ls -l", déterminez quelles sont les deux familles de "fichiers" présents :
.....

Écrivez ci-dessous la liste d'instructions (mkdir, cd, ls, cp etc.) permettant de créer le dossier et les sous-dossiers de **l'employé0001** et testez ces instructions sur l'ordinateur.

.....
.....
.....
.....
.....

Intégrer votre liste d'instructions dans le script afin de générer les 10 premiers dossiers. Vérifiez que votre arborescence fonctionne.

Si ce n'est pas le cas, supprimez tout ce que vous avez créé à l'aide de **rm -r Employe*** (l'étoile permet de supprimer tous les dossiers commençant par Employe) et reprenez l'étape précédente.

Sinon, généralisez à 400 employés.

Que remarquez-vous dans ce script ? Y-a-t'il des choses que vous connaissez déjà ?

.....
.....

4) Gestion des utilisateurs et des groupes

Les systèmes UNIX sont des systèmes multi-utilisateurs : plusieurs utilisateurs peuvent partager un même ordinateur, chacun disposant d'un environnement de travail qui lui est propre.

Chaque utilisateur possède certains droits lui permettant d'effectuer certaines opérations et pas d'autres. Un utilisateur particulier est autorisé à modifier tous les droits : ce "super utilisateur" est appelé "administrateur" ou "root".

Pour un fichier ou un répertoire, le système d'exploitation permet de gérer les droits d'utilisation selon trois catégories :

- ❖ utilisateur (symbolisé par la lettre u = user)
- ❖ groupe d'utilisateur (symbolisé par la lettre g = group)
- ❖ reste du monde (symbolisé par la lettre o = others)

De plus, les fichiers et les répertoires possèdent 3 types de droits :

- ❖ les droits en lecture (symbolisés par la lettre r = Read) : est-il possible de lire le contenu de ce fichier ?
- ❖ les droits en écriture (symbolisés par la lettre w = Write) : est-il possible de modifier le contenu de ce fichier ?
- ❖ les droits en exécution (symbolisés par la lettre x = eXecute) : est-il possible d'exécuter le contenu de ce fichier (quand le fichier du code exécutable) ?

Rappel : Il est possible d'utiliser la commande "ls" avec l'option "-l" afin d'avoir la majorité des informations concernant un fichier.

Exercice 9 :

Effectuer un `ls -l` dans `C10_OS/Cours` . Observez la première colonne d'informations et réécrivez ci-dessous les symboles que vous obtenez :

.....
.....
.....

Dans mon cas :

```
total 3088
-rwxr-xr-x@ 1 vincentbouillot  staff    369K 27 nov 23:00 OS_Eleves.pages*
-rwxr-xr-x@ 1 vincentbouillot  staff    1,1M  6 jan 23:19 OS_Prof.pages*
-rw-r--r--@ 1 vincentbouillot  staff     80K  6 jan 21:22 Turing.jpg
drwxr-xr-x@ 2 vincentbouillot  staff     64B  6 jan 21:24 old/
```

Lisons la première ligne de gauche à droite :

- ❖ le premier symbole "-" signifie que l'on a affaire à un fichier, dans le cas d'un répertoire, nous aurions un "d" (voir la 2e ligne)

- ❖ les 3 symboles suivants "rwx" donnent les droits du propriétaire du fichier : lecture autorisée (r), écriture autorisée (w), exécution autorisée (x)
- ❖ les 3 symboles suivants "r-x" donnent les droits du groupe lié au fichier : lecture autorisée (r), écriture interdite (- au lieu de w), exécution autorisée (x)
- ❖ les 3 symboles suivants "r-x" donnent les droits des autres utilisateurs : lecture autorisée (r), écriture interdite (- au lieu de w), exécution autorisée (x)
- ❖ le caractère suivant "1" donne le nombre de liens (nous n'étudierons pas cette notion ici)
- ❖ le premier "vincentbouillot" représente le nom du propriétaire du fichier
- ❖ le second "staff" représente le nom du groupe lié au fichier

Exercice 10 :

De la même manière, étudiez la quatrième ligne concernant **old/**

.....

.....

.....

.....

.....

.....

.....

.....

.....

Définition : modification des permissions d'un fichier

Le propriétaire d'un fichier peut modifier les permissions d'un fichier ou d'un répertoire à l'aide de la commande "chmod" ayant la syntaxe suivante:

```
chmod [u g o a] [+ - =] [r w x] nom_du_fichier
```

Rem :

- ❖ "u" correspond au propriétaire, "g" correspond au groupe lié au fichier (ou au répertoire), "o" correspond aux autres utilisateurs et "a" correspond à "tout le monde" .
- ❖ "+" ajoute une permission, "-" retire une permission
- ❖ "r" : lecture, "w" : écriture, "x" : exécution.

Exemple :

`chmod g+x monfichier.txt`

donne les droits d'exécution au groupe sur le fichier monfichier.txt.

`chmod u-wr monfichier.txt`

retire les droits d'écriture et de lecture à l'utilisateur sur le fichier monfichier.txt.

`chmod u=wr,g=r,o= monfichier.txt`

donne les droits d'écriture et de lecture à l'utilisateur et les droits de lecture au groupe sur le fichier monfichier.txt.

Exercice 11 :

Dans le répertoire Desktop, créez deux répertoires appelés "Vacances" et "Travail". Dans le répertoire Vacances, créez un fichier toto.jpg à l'aide de touch.

Que s'affiche-t-il avec `ls -l` ?

.....
Ajoutez des droits en écriture à tous les utilisateurs. Que s'affiche-t-il avec `ls -l` ?

.....
Retirez les droits en écriture aux utilisateurs invités. Que s'affiche-t-il avec `ls -l` ?

.....
Donnez les droits en écriture à tout le monde, les droits en lecture aux invités et les droits d'exécution au créateur du fichier en une ligne ! Vérifiez votre réponse avec `ls -l`.

Exercice 12 :

Créez un répertoire "test_NSI" dans votre répertoire personnel. Placez-vous dans le répertoire "test_NSI".

Créez un fichier "toto.txt", vérifiez les permissions associées à ce fichier. Modifiez les permissions associées au fichier "toto.txt" afin que les "autres utilisateurs" aient la permission "écriture".

Exercice 13 : Permission et code octal :

Les droits peuvent également s'écrire et se modifier grâce à une notation en base 8 (octal).

Par exemple, les droits en écriture symbolique :

`rwX --X r-X`

se traduisent par `111 001 101` en binaire, soit `715` en octal.

`chmod 715 monfichier.txt` donne alors directement ces droits au fichier.

En octal, comment se traduit `--x --x --x` ? `rwX rwX rw-` ?

Quels droits correspondent à `654`, à `755` ?

Pour faire le point et à l'aide du cours, compléter le tableau ci-dessous :

Commandes à connaître impérativement :

Commande	Description
pwd	
ls	
cp	
mv	
rm	
mkdir	
cd	
cat	
touch	
chmod	