

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 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 ?

le zip, un dossier et un jpg

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

On a des fichiers et des dossiers. Les dossiers commencent par d.

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.

mkdir Employe0001

mkdir Employe\$suffix

mkdir Employe0001/Autre

mkdir Employe\$suffix/Autre

mkdir Employe0001/Administratif

mkdir Employe\$suffix/Administratif

mkdir Employe0001/Technique

mkdir Employe\$suffix/Technique

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 ?

Il y a une boucle comme en Python ! Printf et echo semblent afficher des choses même si on ne comprend pas bien comment fonctionne printf.

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/**

Lisons la première ligne de gauche à droite :

"d" : c'est un répertoire

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 "2" donne le nombre de liens : pour un dossier, c'est en gros le nombre de sous-fichiers et dossiers contenus dans le dossier.

le premier "vincentbouillot" représente le nom du propriétaire du fichier

le second "staff" représente le nom du groupe lié au fichier

Ensuite vient la taille en octet

Finalement, la date de création.

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 ?

```
-rw-r--r-- 1 vincentbouillot staff 0 10 mai 20:56 toto.jpg
```

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

```
chmod a+w toto.jpg
```

```
ls -l toto.jpg
```

```
-rw-rw-rw- 1 vincentbouillot staff 0 10 mai 20:56 toto.jpg
```

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

```
chmod o-w toto.jpg
```

```
ls -l toto.jpg
```

```
-rw-rw-r-- 1 vincentbouillot staff 0 10 mai 20:56 toto.jpg
```

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.

```
chmod a=w,o=r,u=x toto.jpg
```

```
ls -l toto.jpg
```

```
---x-w-r-- 1 vincentbouillot staff 0 10 mai 20:56 toto.jpg*
```

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 le groupe ait la permission "lecture et exécution" et 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 ? 111

rwX rwX rw- ? 776

Quels droits correspondent à 654, à 755 ? rw- r-x r-- et rwX r-x r-x

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

Commandes à connaître impérativement :

Commande	Description
pwd	indique le chemin absolu vers le répertoire actif
ls	liste le contenu d'un répertoire
cp	copie un fichier ou un répertoire (option -r) vers un autre répertoire
mv	déplace un fichier ou un répertoire (option -r) vers un autre répertoire
rm	supprimer un fichier ou un répertoire (option -r) définitivement
mkdir	crée un répertoire
cd	se déplace à l'adresse relative ou absolue indiquée
cat	affiche le contenu d'un fichier
touch	crée un fichier ou modifie la date de dernière modification d'un fichier
chmod	change les permissions d'un fichier ou dossier