

Outils système (L2 info.)

TP 2

Tout ce TP doit être fait avec Emacs.

Exercice 1 _____ Help !

Avec l'aide de `man`, retrouvez à quoi correspond l'option `-i` des commandes `ls` et `rm`.

Expliquez dans un fichier les commandes que vous avez tapées, et formulez à votre façon le résultat trouvé.

Exercice 2 _____ Les droits

Comme dans l'exercice précédent, indiquez dans un fichier les commandes vous permettant d'accomplir les étapes suivantes, et leur résultat.

1. Créez un fichier vide nommé `vide`.
2. Ajoutez lui les droits d'exécutions pour vous, et exécutez-le. Que se passe-t-il ?
3. Retirez les droits d'écriture du dossier courant et des fichiers qu'il contient pour votre groupe et les autres.
4. Retirez vos droits de lecture à `vide` et essayez de le lire avec `cat`.
5. Supprimez `vide` et restaurez vos droits sur le dossier courant. Vérifiez que la commande a fonctionné.

Exercice 3 _____ Un peu de rangement

Vous être un programmeur java talentueux et savant depuis votre bonne note en BPO, mais quel bazar vous avez laissé derrière vous ! Vous n'avez pas écouté les instructions et mis tous les fichiers java dans un même dossier, nommé `java`.

Dans ce dossier `java` se trouve les fichiers `.java`, parfois les fichiers `.class` correspondant, des données (fichiers `.png` et `.jpg` dont le nom commence par `DATA`) ainsi que les ressources des TPs (fichier `.zip` et `.png`), des sujets de TP (fichier `.pdf`) et deux ou trois trucs qui n'ont rien à faire là.

Enfin vos fichiers java ne sont pas organisé en répertoire selon leur package. Vous les avez nommés `package.Nom2Classe.java`.

Si Martine Gautier voit ça, vous allez souffrir... Il faut vite ranger tout cela !

Le but Une fois rangé, votre dossier doit contenir :

- un dossier par package avec dedans
 - un dossier **out** contenant les fichiers **.class**,
 - Un dossier **src** contenant les fichiers **.java**,
 - Les dossiers des sous-packages ;
- un dossier **ressource** contenant tous les ressources des TPs ;
- un dossier **sujet** contenant les sujets ;
- un dossier **data** contenant les données ;
- un dossier **bash** avec vos programmes **bash** et un **README** les décrivant ;
- un dossier **divers** contenant le reste.

Vous devez :

- créer un (ou plusieurs) fichier(s) **bash** faisant le travail en toute généralité ; ce(s) fichier(s) devra(ont) fonctionner pour le dossier **java** mais aussi pour tout autre dossier ayant la même structure ;
- faire attention lors du déplacement des **.png**, certains vont dans le dossier **data** d'autres dans le dossier **ressources** ;
- ne pas essayer de compiler les fichiers java (les noms des packages ne correspondent plus) !

Vous pouvez :

- créer des fichiers **.bash** et/ou des fonctions intermédiaires ;
- utiliser **IFS**, *internal field separator* (il suffit de créer une variable **IFS** avec le séparateur voulu dedans, puis de faire une boucle **for** sur la chaîne de caractère à découper ;

```
var="truc1 ; truc2 ; truc3 "  
IFS=";"  
for e in $var ; do echo "$e" ; done
```

Attention : pensez à sauvegarder la valeur de l'IFS par défaut avant de la changer pour pouvoir la restaurer, sinon vos **for** vont avoir ensuite des comportements étranges...

Ordre conseillé de programmation :

- créez un programme **bash** pour créer le dossier correspondant au package donné en paramètre du programme, avec les dossiers **src** et **out** vides, s'ils n'existent pas ;
- trie d'abord les choses simples (pdf, zip, ...), puis les moins simples (png), avant de trier les **.java** à un niveau de profondeur de package fixé, puis triez les **.java** quelque soit leur niveau de profondeur (pensez à faire un tableau).