

Ariyathashan Nilanth  
27/03/15

1TSIOC-SISR1

## Gestion d'une poubelle

Création de la commande **poubelle** dans le répertoire **bin**

```
root@debian:~/bin# cd /bin
root@debian:/bin# nano poubelle
root@debian:/bin#
```

une fois terminé nous allons mettre un script qui affiche un texte à fin de voir le bon fonctionnement de la commande

```
GNU nano 2.2.4 Fichier
#!/bin/bash
echo 'bonjour'
```

ensuite on exécute la commande

```
root@debian:/bin# poubelle
-bash: /bin/poubelle: Permission non accordée
root@debian:/bin#
```

mais cela ne fonctionne pas car nous n'avons pas attribué de permission pour l'exécution donc il est nécessaire d'attribuer une permission.

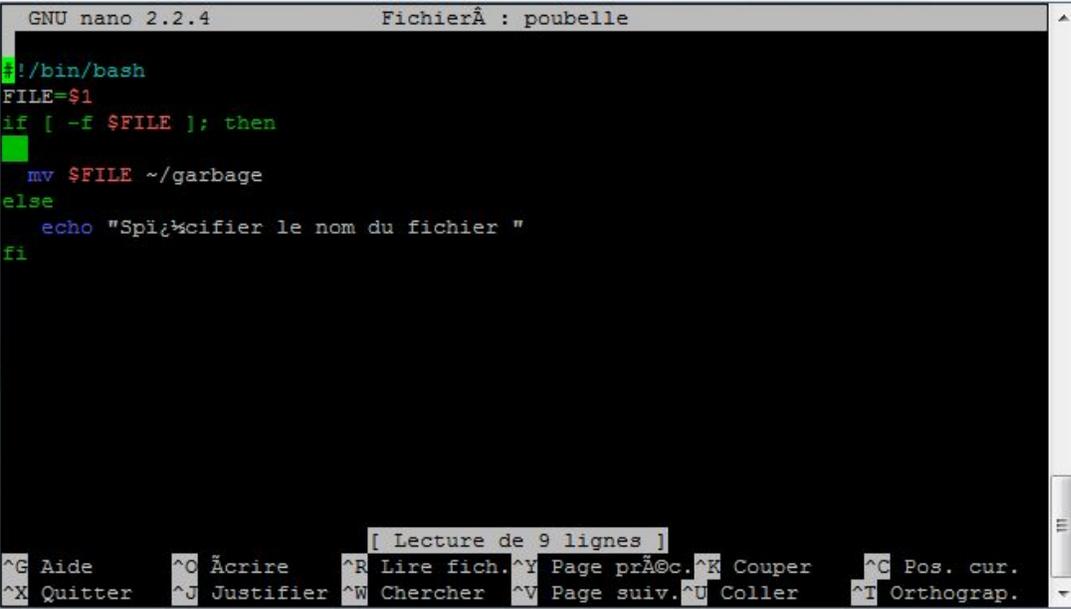
```
root@debian:/bin# poubelle
-bash: /bin/poubelle: Permission non accord  e
root@debian:/bin# chmod o+x poubelle
root@debian:/bin# █
```

Apr  s l'attribution du permission nous allons remettre la commande

```
root@debian:/bin# poubelle
bonjour
█
```

On constate la commande fonctionne , le Texte '**Bonjour**' s'affiche.

Maintenant nous allons pouvoir mettre en place le script qui permet de g  rer la suppression des fichier en mettant les fichiers dans le dossier garbage.



```
GNU nano 2.2.4      Fichier   : poubelle
#!/bin/bash
FILE=$1
if [ -f $FILE ]; then
mv $FILE ~/garbage
else
echo "Sp  cifier le nom du fichier "
fi
[ Lecture de 9 lignes ]
^G Aide      ^O   crire   ^R Lire fich.^Y Page pr  c. ^K Couper  ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller   ^T Orthograp.
```

Nous allons ensuite tester le bon fonctionnement, premi  rement nous allons mettre un nom du fichier qui n'existe pas avec la commande.

```
root@debian:/bin# poubelle ddf
Sp  cifier le nom du fichier
root@debian:/bin# █
```

On peut voir que que la commande n'a pas trouvé ce fichier inconnu , donc cela semble bien logique.

ensuite nous allons tenter la même chose avec un fichier qui existe  
on peut voir que la commande a fonctionné en plaçant le fichier qu'on a voulu supprimer dans le répertoire **garbage**.

```
root@debian:~# poubelle nilanth.html
root@debian:~# ls garbage
nilanth.html ok.html
root@debian:~# █
```

### Fonctionnalité numéro 2 : Lister le contenu de la poubelle

On pourrait évidemment faire `ls ~/garbage` pour afficher le contenu de la poubelle ; cependant dans la plupart des cas on va masquer l'endroit où se trouve la poubelle. Il faut donc permettre à l'utilisateur de lister le contenu de la poubelle sans savoir où elle se trouve.

**Travail à faire : Modifier le script poubelle (ne pas écrire un nouveau script) pour qu'il affiche le contenu de la poubelle en réponse à la commande (avec -l comme « liste »)**

- `poubelle -l`

### Fonctionnalité numéro 3 : Restaurer un fichier depuis la poubelle

Il faut permettre à l'utilisateur de récupérer un fichier effacé par mégarde. Pour cela la commande sera (avec -r comme « restaurer »)

- `poubelle nomfichier -r`

**Le script doit :**

**vérifier que le nomfichier est bien dans la poubelle, sinon indiquer une erreur « fichier absent de la poubelle »**

**recopier le fichier vers le répertoire « home de l'utilisateur » ~**

**Fonctionnalité numéro 4 : Vider la poubelle**

**Ajoutez l'option -v qui offre les possibilités suivantes**

**poubelle nomfichier -v**

**enlève définitivement « nomfichier » de la poubelle (s'il y est ; sinon affiche « fichier absent de la poubelle » )**

**poubelle -v**

**vide toute la poubelle avec message préalable de confirmation « êtes-vous sûr ? »**

---