**Utilitaire trash-cli**

Une image contenant tasse, café, canette

Description générée automatiquement

**Introduction**

La suppression de fichiers sous Linux est un processus assez simple si on travaille avec une interface graphique.  
Les fichiers supprimés sont déplacés dans la corbeille.  
Il est donc possible d'accéder aux fichiers supprimés et, même de les restaurer si cela s’avère nécessaire.

En ligne de commande, afin de supprimer un objet (fichier ou répertoire), on doit utiliser la commande **rm**.  
Une fois invoqué, la commande **rm** efface immédiatement les fichiers spécifiés, sans possibilité de retour.

Pour récupérer un fichier supprimé de la ligne de commande Linux, il faut utiliser un logiciel dédié tel que PhotoRec[[1]](#footnote-1).  
Toutefois, le processus est complexe et lourd.

L'utilitaire **trash-cli** ajoute la possibilité de déplacer des fichiers vers une corbeille à partir de la ligne de commande.

La corbeille stocke temporairement les fichiers supprimés, ce qui aide à les restaurer lorsque cela est nécessaire.   
Si on ne veut pas ces fichiers, il suffit de les supprimer définitivement (on vide la corbeille).

Le répertoire **$HOME/.local/share/Trash** contient deux sous-répertoire ...

* Le répertoire **files** stocke les fichiers et répertoires réellement supprimés   
  et
* Le répertoire info contient des informations sur les fichiers et dossiers supprimés   
  tels que le chemin du fichier, la date et l'heure supprimées dans un fichier séparé.

L’utilitaire **trash-cli** est une interface de ligne de commande pour l'utilitaire **Trashcan** conforme aux spécifications de la corbeille FreeDesktop.org.   
Il stocke le nom, le chemin d'accès d'origine, la date de suppression et les autorisations de chaque fichier supprimé.

**Installation**

Pour installer trash-cli ..  
**>> sudo apt install trash-cli --yes**

Une fois installé, l'utilitaire **trash-cli** fournit cinq commandes qui peuvent être utilisées afin de supprimer des fichiers ou des répertoires, afficher le contenu de la corbeille, restaurer des fichiers, …

* **trash-put**  
  Suppression de fichiers ou répertoires spécifiés;
* **trash-list**Affichage du contenu de la corbeille ;
* **trash-rm**  
  Suppression des fichiers ou des répertoires individuels de la corbeille ;
* **trash-empty**  
  Suppression de tous les fichiers et répertoires de la corbeille ;
* **trash-restore**Restauration d’un fichier ou d’un répertoire spécifié.

**Suppression de fichiers ou répertoires**

Pour supprimer un fichier, on utilise la commande **trash-put**.

On peut fournir une liste de fichiers séparés par des espaces et même utiliser la correspondance de modèles (*globs*) pour spécifier les fichiers que l’on souhaite supprimer.

Contrairement à la commande rm, qui ne supprime pas les répertoires non vides à moins que l’on utilise le commutateur **--recursive** (option courte **-R**), **trash-put** n'a pas de telles réserves quant à leur suppression.

On crée plusieurs fichiers vides …  
**>> touch fichier{1..9}.txt  
>> la -l**

L'utilitaire trash-cli utilise le répertoire **$HOME/.local/share/Trash/files/**, il est possible d’xécuter immédiatement la commande **trash-list** pour afficher le contenu de ce répertoire.   
Autrement dit, les fichiers supprimés à l'aide de la commande trash-put sont accessibles avec la commande trash-list ...  
**>> trash-list**

La commande **trash-list** affiche la date de création et la date de suppression des fichiers mis à la corbeille.

**Remarque** …  
Le déplacement de fichiers vers la corbeille ne libère pas d'espace sur le système de fichiers tant que la corbeille n'est pas vidée.

Lors de l’exécution de la commande **trash-restore (**sans arguments supplémentaires), l'utilitaire affiche tous les fichiers du répertoire **$HOME/.local/share/Trash/files/** et demande de choisir le fichier que l’on cherche à restaurer.   
Les fichiers se voient tous attribuer un numéro unique, et il faut indiquer celui correspondant au fichier que l’on cherche à restaurer.

**Remarque** …  
L'inconvénient de cette approche est que l’on ne peut pas restaurer simultanément plusieurs fichiers.

**Restauration de fichiers et répertoire**

La commande **trash-restore** assure également le suivi du répertoire de travail actuel.   
Si on invoque la commande depuis le répertoire /home, **trash-restore** liste tous les fichiers du répertoire **$HOME/.local/share/Trash/files/**.

**Remarque** …  
Cependant, si on exécute la commande **trash-restore** à partir d'un autre répertoire, elle ne liste que les fichiers de la corbeille qui étaient à l'origine stockés dans ce répertoire.

**Vidage de la corbeille**

On peut supprimer des fichiers ou des répertoires **$HOME/.local/share/Trash/files/** avec les commandes **trash-rm <Motif>** ou **trash-rm <Fichier et Répertoires>** …   
**>> trash-rm \*.html**

**Remarque** …  
La commande **trash-rm** attend le nom complet du fichier et on ne peut pas utiliser l’auto-complétion (TAB) avec cette commande.

Ainsi, si on cherche supprimer un fichier nommé **liste des usagers.texte** du répertoire **$HOME/.local/share/Trash/files/**, il faut saisir le nom complet du fichier.   
Si on ne fournit qu'une partie du nom à la commande **trash-rm**, il ne supprime pas le fichier correspondant et ne signale aucune erreur.   
On peut toutefois exécuter la commande trash-rm à l’aide de caractères génériques (globs) pour supprimer le fichier et tous les autres fichiers débutant par le motif memo (trash-rm memo\*).

La commande **trash-empty** peut également être utilisée pour supprimer tous les fichiers et répertoires.   
Le commutateur **<Jours/days>** peut être utilisée pour supprimer uniquement les fichiers effacés pendant plus longtemps qu'un nombre de jours spécifié.

Par exemple, si on souhaite éliminer de la corbeille tous les fichiers qui ont été supprimés il y a plus d'un mois, on peut exécuter la commande **trash-empty 30**.   
La commande trash-empty 1 supprime de la même manière les fichiers qui se trouvent dans **$HOME/.local/share/Trash/files/** depuis plus d'une journée.

**Utilitaire rm-protection**

On peut utiliser l'utilitaire rm**-protection** pour obliger les utilisateurs à répondre à des questions de sécurité avant de pouvoir supprimer des fichiers importants.

**Installation**

Pour installer l’utilitaire …  
**>> sudo pip3 install rm-protection**

**Remarque** …  
Si on obtient un message d'erreur indiquant que pip n'est pas une commande connue, il faut d'abord l’installer …  
**>> sudo apt install python3-pip**

L'utilitaire **rm-protection** fournit deux commandes …

* **protect**  
  La commande est utilisée pour spécifier les fichiers que l’on souhaite protéger contre les suppressions accidentelles   
  et
* **rm-p**  
  Cette commande peut être utilisée afin de supprimer les fichiers protégés.

**Utilisation**

**rm-protect** force les utilisateurs à répondre à une question avant de supprimer un fichier.

**Remarque** …  
On peut toujours supprimer un fichier protégé à l'aide de la commande rm ou de la commande trash-put .

Si on décide d'utiliser la commande protect pour protéger ses fichiers, il est préférable d’envisager de créer un alias pour la commande **rm-p** comme **alias** **rm="rm-p"** afin que les utilisateurs qui utilisent la commande **rm** soient redirigés vers **rm-p**.

Lors de la protection d’un fichier, la question et la réponse sont stockées dans un fichier texte brut nommé   
**.filename.rm-protection**.   
Les réponses sont sensibles à la casse, donc saisir "tux.texe" comme réponse à la place de "Tux.Texte" entraîne une erreur.

Si on est rassuré par la possibilité de restaurer des fichiers supprimés, il est possible d’ignorer la commande **rm-p**.   
Tout comme la commande **rm**, elle contourne le répertoire **$HOME/.local/share/Trash/files/**.

**Utilitaire chattr**

chattr est un utilitaire qui permet de pour protéger en écriture des fichiers.

La commande **sudo** **chattr +i** **<Nom de fichier>** protège un fichier.  
La commande **sudo** **chattr -i <Nom de fichier>** supprime la protection …  
**>> sudo chattr +i fichier6.txt  
>> sudo rm fichier6.txt  
rm: impossible de supprimer 'fichier6.txt': Opération non permise  
>> sudo chattr -i fichier6.txt**

Si on a accidentellement perdu des données précieuses à cause d'une suppression précipitée, il existe une solution rapide pour éviter ce type de perte.   
On peut utiliser la commande **chattr** afin de modifier les attributs associés à un fichier.  
La syntaxe par défaut est **chattr [Opéérateur] [Bascule] [filename]** .

Exécutez la commande sudo chattr +i [nom de fichier] pour définir l'attribut immuable. Vous constaterez que vous ne pouvez plus supprimer le fichier et que vous ne pouvez pas non plus le modifier. Exécutez lsattr filename pour afficher les attributs du fichier spécifié. sudo chattr -R +i <directoryname> peut également être utilisé pour modifier de manière récursive l'attribut de tous les fichiers dans le répertoire spécifié.

Si vous souhaitez pouvoir modifier le fichier, mais souhaitez uniquement le protéger contre la suppression, vous pouvez utiliser l'attribut append au lieu de immutable : sudo chattr +a filename . Remplacez le + par un opérateur - pour supprimer l'attribut d'un fichier. Par exemple, l'exécution de la commande sudo chattr -i filename vous donnera à nouveau la possibilité de modifier ou de supprimer le fichier.

-----

$ sudo pip installer corbeille-cli

Collecte des déchets-cli

Téléchargement de la corbeille-cli-0.17.1.14.tar.gz

Installation des packages collectés : poubelle-cli

Exécution de setup.py bdist\_wheel pour la corbeille-cli ... terminé

Trash-cli-0.17.1.14 installé avec succès

Comment utiliser Trash-Cli

Ce n'est pas un gros problème car il offre une syntaxe native, fournit les commandes suivantes.

trash-put:Supprimer des fichiers et des dossiers.

trash-list:Pint Fichiers et dossiers supprimés.

trash-restore:Restaurez un fichier ou un dossier à partir de la corbeille.

trash-rm:Supprimez des fichiers individuels de la corbeille.

trash-empty:Videz la ou les poubelles.

Essayons quelques exemples pour expérimenter cela.

1) Supprimer des fichiers et des dossiers : Dans notre cas, nous allons envoyer un fichier nommé 2g.txtet un dossier nommé magià la corbeille en exécutant la commande suivante.

$ poubelle-mettre 2g.txt mages

Vous pouvez voir la même chose dans le gestionnaire de fichiers.

2) Pint Supprimer les fichiers et dossiers : Pour afficher les fichiers et dossiers supprimés, exécutez la commande suivante. Comme je peux voir des informations détaillées sur les fichiers et dossiers supprimés tels que le nom, la date et l'heure et le chemin du fichier.

$ corbeille-liste

2017-10-01 01:40:50 /home/magi/magi/2g.txt

2017-10-01 01:40:50 /home/mages/mages/mages

3) Restaurer un fichier ou un dossier à partir de la corbeille : À tout moment, vous pouvez restaurer des fichiers et des dossiers en exécutant la commande suivante. Il vous demandera d'entrer le choix que vous souhaitez restaurer. Dans notre cas, nous allons restaurer 2g.txtle fichier, donc mon option est 0.

$ corbeille-restauration

0 2017-10-01 01:40:50 /home/magi/magi/2g.txt

1 2017-10-01 01:40:50 /home/mages/mages/mages

Quel fichier restaurer [0..1] : 0

4) Supprimer des fichiers individuels de la corbeille : Si vous souhaitez supprimer des fichiers spécifiques de la corbeille, exécutez la commande suivante. Dans notre cas, nous allons supprimer magile dossier.

$ poubelle-rm mages

5) Vider la corbeille : Pour tout supprimer de la corbeille, exécutez la commande suivante.

$ poubelle-vide

6) Supprimer le fichier plus ancien que X jours : Vous pouvez également supprimer les fichiers plus anciens que X jours, exécutez la commande suivante pour le faire. Dans notre cas, nous allons supprimer 10les vieux objets de la corbeille.

$ corbeille vide 10

trash-cli fonctionne très bien mais si vous voulez essayer une alternative, essayez gvfs-trash & autotrash

<https://www.2daygeek.com/trash-cli-command-line-trashcan-linux-system/>

alias pour remplacer la commande rm (|rm)

**Autres installations**

Pour installer l’utilitaire trash-cli pour RHEL et dérivés …  
**>> sudo dnf install trash-cli**

Pour installer l’utilitaire trash-cli pour Arch Linux et dérivés …  
**>> sudo pacman -S trash-cli**

Pour installer l’utilitaire trash-cli pour SUSE et dérivés …  
**>> sudo zypper install trash-cli**

Si sa distribution n'offre pas trash-cli, il est facilement possible de l’installer à partir de pip.   
Le système doit avoir le gestionnaire de paquets **pip** afin d'installer les packages python.

Une solution alternative est l'installation depuis le référentiel Github …  
**>> mkdir /tmp/ && cd /tmp/  
>> git clone https://github.com/andreafrancia/trash-cli.git  
>> cd trash-cli  
>> sudo python setup.py install**La première commande crée un répertoire nommé trash-cli dans le répertoire de travail et copie le contenu du référentiel trash-cli.   
On peut maintenant exécuter le script setup.py pour installer l'utilitaire.

On peut éventuellement ajouter le commutateur **--user** <Nom Utilisateur> avec la dernière commande afin que l'utilitaire trash-cli ne soit pas installé à l'échelle du système et qu’il ne ne soit disponible que pour l'utilisateur spécifié …  
**>> sudo python setup.py install --user tux**

On peut également l’installer avec l’utilitaire pip3 …  
**>> sudo pip3 install trash-cli**

**Remarque** …  
Si on obtient un message d'erreur indiquant que pip3 n'est pas une commande connue, il faut alors installer pip3 …  
**>> sudo apt install python3-pip**

sudo apt-get install python3-pip

Commande lsattr et chattr

<https://www.computerhope.com/unix/chattr.htm>

1. Pour plus d’informations ...  
    https://www.cgsecurity.org/wiki/PhotoRec [↑](#footnote-ref-1)