Trucs et astuces pour linux

Cette page est au départ, un mémo que j'avais fait pour mon usage personnel avec des informations récoltées sur des listes Linux (debian-user-french et linuxbe). J'y ai ajouté des commentaires et des scripts.

Remarque : "~$ " est le prompt du terminal qu'il ne faut pas saisir.

Avertissement: Certains scripts agissent sur un ensemble de fichiers et sont donc à utiliser avec prudence.

*** Convertir les espaces dans le nom des fichiers en caractère souligné "_"**

```
~$ for i in *\ *; do mv "$i" `echo $i | tr " " "_"`; done
```

pour faire l'inverse

```
~$ for i in *_*; do mv "$i" "`echo $i | tr "_" " "`"; done
```

Si voulez tester cette commande sans que les modifications soient faites:

```
~$ for i in *\ *; do printf "$i\n" | tr " " "_"; done
```

affichera les nouveaux noms des fichiers.

Pour "cibler" les fichiers on peut modifier la commande; par exemple pour n'agir que sur les mp3:

```
~$ for i in *\ *.mp3; do mv "$i" `echo $i | tr " " "_"`; done
```

*** Mettre les noms de fichiers en minuscule**

```
~$ for i in *; do mv "$i" `echo $i | tr [:upper:] [:lower:]`; done
```

*** Changer les extensions des tous les fichiers d'un dossier**

```
~$ for i in *.EXT1; do mv $i `basename $i EXT1`EXT2; done
```

EXT1 est l'extension à modifier ; EXT2 est la nouvelle extension
Par exemple pour remplacer l'extension php3 de vos scritps par php:

```
~$ for i in *.php3; do mv $i `basename $i php3`php; done
```

*** Remplacer un mot dans plusieurs fichiers**

Par exemple pour remplacer le mot windows par linux dans vos fichier texte:

```
perl -i.bak -pe 's/windows/linux/g' *.txt
```

Une fois vérifié que tout va bien vous pouvez effacer les fichiers de sauvegarde (.bak)

*** Transformer une page man en fichier texte:**

```
~$ man emacs | col -b > man_emacs.txt
```

*** Ecrire dans un fichier texte sans passer par un éditeur**

Crée ou remplace le fichier mon_fichier si il existe:

```
~$ cat >mon_fichier.txt <<"EOF"
>Le fichier contiendra les lignes
>que vous tapez jusqu'à ce qu'à
>EOF
~$
```

Ajoute le texte à la fin du fichier ou le crée s'il n'existe pas:

```
~$ cat >>mon_fichier.txt <<"EOF"
>Ajout à la fin du fichier ou création
>de celui-ci
>EOF
~$
```

*** Transformer un fichier texte dos/windows en fichier linux**

```
~$ tr -d "\015" < fich_dos >fich_unix
```

fich_dos étant le fichier dos et fich_unix le fichier obtenu après conversion

*** Découper un fichier pour le mettre sur disquettes**

```
~$ split -b 1457664 mon_fichier part.
```

mon_fichier est le nom du fichier à découper
part. est le préfixe de sortie. Les fichier s'appelleront part.aa, part.ab,...
-b 1456664 demande à split de couper le fichier en morceaux de 1456664 octets ce qui correspond à la capacité d'une disquette.

Le recollage se fait de la manière suivante (une fois les fichiers regroupés dans un même dossier):

```
~$ cat part.* >mon_fichier
```

*** Signature avec une citation aléatoire**

Créez un fichier sign_al.sh contenant:

```
#!/bin/sh
SIGFIFO=$HOME/.signature
rm -f $SIGFIFO
mkfifo $SIGFIFO
while true;
do (
printf "Votre nom\n\nCitation :\n" >>$SIGFIFO | fortune -s >> $SIGFIFO
);
done
```

Rendez ce fichier exécutable avec `chmod +x sign_al.sh`. Puis exécutez le fichier sign_al en arrière plan (`./sign_al.sh &`). A chaque accès au fichier ~/.signature son contenu est changé. Vous pourrez le constater en utilisant cat sur celui-ci. Il ne vous reste plus qu'à dire à votre logiciel de mail d'utiliser .signature pour signer vos messages.

*** Rediriger les messages d'erreurs (*stderr*) vers un fichier**

```
~$ commande 2>err.log
```

2 correspond à stderr c'est à dire la sortie standard pour les erreurs
Exemple d'utilisation:

```
~$ gcc -o mon_prog main.c 2>err.log
~$ cat err.log
main.c: In function `main':
main.c:47: warning: unused variable `editor_display'
```

Autre exemple: Version "améliorée" de la commande pour mettre les noms de fichiers en minuscule

```
~$ for i in *; do mv "$i" `echo $i | tr [:upper:] [:lower:]` 2>/dev/null; done
```

Evite d'avoir des message du style:" `mv: `abcdef' et `abcdef' identifient le même fichier` " pour les fichiers dont le nom ne contient pas de lettres majuscules.

*** Rediriger les erreurs vers la sortie standard (*stdout*)**

```
commande 2>&1
```

Ceci peut être utile pour certains programmes (comme cdrecord ou mkisofs) qui envoient l'aide vers stderr ce qui empêche d'utiliser less pour la faire défiler.

Exemple:

```
~$ cdrecord --help | less
(END)
```

ne fonctionne pas par contre :

```
~$ cdrecord --help 2>&1 | less
cdrecord: Usage: cdrecord [options] track1...trackn
Options:
        -version        print version information and exit
        -v              increment general verbose level by one
        -V              increment SCSI command transport verbose level by one
```

```
        -debug          print additional debug messages
        dev=target      SCSI target to use as CD/DVD-Recorder
        timeout=#       set the default SCSI command timeout to #.
        driver=name     user supplied driver name, use with extreme care
        -checkdrive     check if a driver for the drive is present
        -prcap          print drive capabilities for MMC compliant drives
        -inq            do an inquiry for the drive end exit
        -scanbus        scan the SCSI bus end exit
        -reset          reset the SCSI bus with the cdrecorder (if possible)
        -ignsize        ignore the known size of a medium (may cause problems)
:
```

donne le résultat attendu !

***Reinitialiser la console**

```
~$ reset
Erase is delete.
Kill is control-U (^U).
Interrupt is control-C (^C).
~$
```

Il y a une autre commande (dont j' ai oublié le nom) qui permet de faire ça mais `reset` est plus simple à utiliser losqu'on ne peut plus lire ce qu'on tape (par exemple après un `cat` sur binaire :-))