L'invite de commandes (shell)

Vous avez donc installé Slackware et vous vous retrouvez devant une invite de commande dans un terminal ; et après ? C'est probablement le bon moment pour découvrir quelques outils basiques en ligne de commande. Et comme vous vous demandez pourquoi ce curseur clignote, vous aurez probablement besoin d'un peu d'aide pour vous débrouiller avec ce qui est installé, et c'est là toute la vocation de ce chapitre.

Documentation système

Votre système Linux Slackware comporte beaucoup de documentation interne, pour pratiquement chacune des applications installées. La méthode la plus connue pour consulter la documentation système est ***man***(1). ***man*** (abréviation pour **manuel**) vous affichera la page de manuel de n'importe quelle application, appel système, fichier de configuration ou bibliothèque que vous lui demanderez. Par exemple, man man vous affichera la page de manuel pour ***man*** lui-même.

Malheureusement, vous ne pouvez pas toujours savoir quel programme utiliser pour votre travail en cours. Heureusement, ***man*** dispose d'une fonction de recherche intégrée. En utilisant l'option *-k*, ***man*** recherchera toutes les pages correspondant à votre critère de recherche.

Les pages de manuel sont organisées en groupes ou sections selon leur type de contenu. Par exemple, la section 1 correspond aux programmes pour les utilisateurs. ***man*** recherchera dans chaque section par ordre et affichera le premier résultat obtenu. Parfois vous verrez qu'une page de manuel dispose d'une entrée dans plus d'une section. Dans ce cas, vous aurez besoin d'indiquer dans quelle section rechercher exactement. Dans ce livre, tous les programmes et un certain nombre d'autres choses auront un chiffre entre parenthèses sur la droite. Ce chiffre est la section du manuel où vous pourrez trouver des informations sur cet outil.

```
darkstar:~$ man -k printf
printf               (1)  - format and print data
printf               (3)  - formatted output conversion
darkstar:~$ man 3 printf
```

Sections du manuel

Section	Contenu
1	Programmes utilisateurs
2	Appels systèmes
3	Appels de la bibliothèque C
4	Périphériques
5	Formats de fichiers / Protocoles
6	Jeux
7	Conventions / Macro Packages
8	Administration système
9	Descriptions des API du noyau
n	“Nouveau” - généralement utilisé pour Tcl/Tk

Gestion des fichiers et répertoires

Lister le contenu des fichiers et des répertoires

ls(1) est utilisée pour lister les fichiers et répertoires, leurs permissions, taille, type, numéro d'inode, propriétaire et groupe et de nombreuses autres informations. Par exemple, listons ce qui est présent dans le répertoire / de votre nouveau système Linux Slackware.

```
darkstar:~$ ls /
bin/   dev/  home/  lost+found/  mnt/  proc/  sbin/  sys/  usr/
boot/  etc/  lib/   media/       opt/  root/  srv/   tmp/  var/
```

Remarquez que chaque élément de la liste est un répertoire. Ils sont faciles à distinguer des fichiers ordinaires, par la présence du caractère / à la fin ; les fichiers réguliers n'ont pas de suffixes. Par ailleurs, les fichiers exécutables ont une astérisque comme suffixe. Mais ***ls*** peut faire bien plus. Pour voir les permissions d'un fichier ou d'un répertoire, vous devrez faire une "*liste longue*".

```
darkstar:~$ ls -l /home/alan/Desktop
-rw-r--r-- 1 alan users 15624161 2007-09-21 13:02 9780596510480.pdf
-rw-r--r-- 1 alan users  3829534 2007-09-14 12:56 imgscan.zip
drwxr-xr-x 3 alan root       168 2007-09-17 21:01 ipod_hack/
drwxr-xr-x 2 alan users      200 2007-12-03 22:11 libgpod/
drwxr-xr-x 2 alan users      136 2007-09-30 03:16 playground/
```

Une liste longue vous permet de voir les permissions, l'utilisateur et le groupe propriétaire, la taille, la date de dernière modification et bien sûr le nom du fichier lui-même. Remarquez que les deux premières entrées sont des fichiers et les trois dernières sont des répertoires. Cela se remarque par le tout premier caractère en début de ligne. Les fichiers réguliers ont un "-" ; les répertoires ont un "*d*". Il y a plusieurs autres types de fichiers avec leurs dénominateurs propres. Les liens symboliques utiliseront "*l*" par exemple.

Pour terminer, nous verrons comment lister les fichiers "points", ou fichiers cachés. Au contraire des autres systèmes d'exploitation tel que Microsoft Windows, il n'y a aucune propriété particulière qui différencie les fichiers "*cachés*" des fichiers "*visibles*". Un fichier caché commence simplement par un point. Pour afficher ces fichiers avec tous les autres, vous avez simplement besoin d'utiliser l'option *-a* avec ***ls***.

```
darkstar:~$ ls -a
.xine/    .xinitrc-backup  .xscreensaver  .xsession-errors  SBo/
.xinitrc  .xinitrc-xfce    .xsession      .xwmconfig/       Shared/
```

Vous remarquerez également que vos fichiers et répertoires apparaissent avec des couleurs différentes. Parmi les améliorations de ***ls*** comme pour l'ajout de caractères en fin de nom pour indiquer le type de fichier des fonctionnalités spéciales du programme ***ls*** sont activées en utilisant certaines options. Par commodité, Slackware paramètre ***ls*** pour utiliser plusieurs de ces options supplémentaires par défaut. Elles sont gérées par les variables d'environnement LS_OPTIONS et LS_COLORS. Nous parlerons plus en détails des variables d'environnement dans le chapitre 5.

Se déplacer dans le système de fichiers

cd est la commande utilisée pour changer de répertoire. Contrairement à la plupart des autres commandes, ***cd*** n'est pas un programme à proprement parler, mais est intégrée au shell. Fondamentalement, cela signifie que ***cd*** n'a pas sa propre page de manuel. Vous devrez consulter la documentation du shell pour plus de détails sur la commande ***cd*** que vous utilisez. Mais pour la plus grande partie, toutes se comportent de manière similaire.

```
darkstar:~$ cd /
darkstar:/$ls
bin/   dev/  home/  lost+found/  mnt/  proc/  sbin/  sys/  usr/
boot/  etc/  lib/   media/       opt/  root/  srv/   tmp/  var/
darkstar:/$cd /usr/local
darkstar:/usr/local$
```

Avez-vous remarqué que l'invite de commande à changé lorsque nous avons changé de répertoire ? Le shell par défaut de Slackware propose cela pour connaître facilement et rapidement votre répertoire courant, mais ce n'est pas une fonctionnalité de ***cd***. Si votre shell ne se comporte pas de cette façon, vous pouvez facilement obtenir votre répertoire courant avec la commande ***pwd***(1). La plupart des shells UNIX ont une invite de commande configurable qui peut être peaufinée pour fournir la même fonctionnalité. En fait, ceci est un autre agrément de la configuration de votre shell par défaut effectué par Slackware.

```
darkstar:~$ pwd
/usr/local
```

Création et suppression de fichiers et de répertoires

Bien que la plupart des programment puissent et doivent créer leur propres fichiers et répertoires, vous aurez souvent envie d'en créer vous même. Heureusement, cela est très simple en utilisant ***touch***(1) et ***mkdir***(1).

touch modifie en fait l'horodatage (*timestamp*) d'un fichier, mais si celui-ci n'existe pas, il sera crée.

```
darkstar:~/foo$ ls -l
-rw-r--r-- 1 alan users 0 2012-01-18 15:01 bar1
darkstar:~/foo$ touch bar2
-rw-r--r-- 1 alan users 0 2012-01-18 15:01 bar1
-rw-r--r-- 1 alan users 0 2012-01-18 15:05 bar2
darkstar:~/foo$ touch bar1
-rw-r--r-- 1 alan users 0 2012-01-18 15:05 bar1
-rw-r--r-- 1 alan users 0 2012-01-18 15:05 bar2
```

Remarquez comment `bar2` a été crée dans notre deuxième commande et la troisième commande a simplement mis à jour l'horodatage de `bar1`.

mkdir est utilisé (de manière assez évidente) pour créer des répertoires [1)]. **mkdir foo** créera le répertoire “foo” dans le répertoire courant. De plus, vous pouvez utiliser l'option *-p* pour créer les

répertoires parents manquants.

```
darkstar:~$ mkdir foo
darkstar:~$ mkdir /slack/foo/bar/
mkdir: cannot create directory `/slack/foo/bar/': No such file or directory
darkstar:~$ mkdir -p /slack/foo/bar/
```

Dans le dernier cas, ***mkdir*** essayera de créer “/slack”, puis “/slack/foo” et enfin “/slack/foo/bar”. Si vous oubliez l'option *-p* ***mkdir*** échouera lors de la création de “/slack/foo/bar”, à moins que les deux premiers répertoires existent déjà, comme vous pouvez le voir dans l'exemple.

Supprimer un fichier est aussi simple que d'en créer un. La commande ***rm***(1) supprimera un fichier (en assumant que vous en avez la permission). Il y a très peu d'options courante pour ***rm***. La première est *-f* est elle est utilisée pour forcer la suppression d'un fichier lorsque vous n'avez pas la permission explicite de le supprimer. L'option *-r* supprimera des répertoires et leur contenu de manière récursive.

Il existe un autre outil pour supprimer les répertoires, le simplissime ***rmdir***(1). ***rmdir*** ne supprimera que les répertoires vides et se plaindra bruyamment à propos de ceux contenant des fichiers ou des sous-répertoires.

```
darkstar:~$ ls
foo_1/ foo_2/
darkstar:~$ ls foo_1
bar_1
darkstar:~$ rmdir foo_1
rmdir: foo/: Directory not empty
darkstar:~$ rm foo_1/bar
darkstar:~$ rmdir foo_1
darkstar:~$ ls foo_2
bar_2/
darkstar:~$ rm -fr foo_2
darkstar:~$ ls
```

Archives et compression

Tout le monde peut avoir besoin de regrouper un paquet de petits fichiers ensemble pour les sauvegarder facilement de temps en temps, ou peut-être avez-vous besoin de compresser de très gros fichiers pour les manipuler plus facilement ? Peut-être souhaitez-vous faire le tout en même temps ? Heureusement, il a plusieurs outils justement là pour ça.

zip et unzip

Vous êtes probablement familier des fichiers .zip. Ce sont des fichiers compressés qui contiennent d'autres fichiers et répertoires. Bien que nous n'utilisions pas habituellement ces fichiers dans le monde Linux, ils sont néanmoins largement utilisés par d'autres systèmes d'exploitation, et nous devons les gérer de temps en temps.

Pour créer un fichier zip, vous aurez besoin d'utiliser (naturellement) la commande ***zip***(1). Vous pouvez compresser soit des fichiers ou des réperoires (ou les deux) avec ***zip*** mais vous devrez utiliser l'option *-r* pour les actions récursives concernant les répertoires.

```
darkstar:~$ zip -r /tmp/home.zip /home
darkstar:~$ zip /tmp/large_file.zip /tmp/large_file
```

L'ordre des options est très important. Le premier nom de fichier doit être le fichier zip à créer (si l'extension .zip est manquante, ***zip*** l'ajoutera pour vous) et ce qui suit sont les fichiers et répertoires à ajouter à votre fichier zip.

Naturellement, ***unzip***(1) décompressera les archives zip.

```
darkstar:~$ unzip /tmp/home.zip
```

gzip

Un des outils de compression les plus anciens inclus dans Slackware est ***gzip***(1), un outil de compression qui n'est capable d'opérer que sur un seul fichier à la fois. Alors que ***zip*** est à la fois un outil de compression et d'archivage, ***gzip*** ne peut que compresser. À première vue, cela semble être une faiblesse, mais c'est vraiment une force. La philosophie UNIX de faire de petits outils faisant leurs petits boulots offre une myriade de façons de combiner ces petits outils. Pour compresser un fichier (ou plusieurs fichiers), il suffit de les passer comme arguments à ***gzip***. Chaque fois que ***gzip*** compresse un fichier, il ajoute une extension .gz et supprime le fichier original.

```
darkstar:~$ gzip /tmp/large_file
```

La décompression est tout aussi simple avec ***gunzip*** qui va créer un nouveau fichier non compressé et supprimer l'ancien.

```
darkstar:~$ gunzip /tmp/large_file.gz
darkstar:~$ ls /tmp/large_file*
/tmp/large_file
```

Mais supposons que nous ne voulons pas supprimer l'ancien fichier compressé, nous voulons juste de lire son contenu ou l'envoyer comme entrée vers un autre programme ? Le programme ***zcat*** va lire le fichier gzip, le décompresser dans la mémoire, et envoyer son contenu sur la sortie standard (l'écran du terminal sauf si la sortie standard est redirigée, consulter la section "Redirection d'entrée et de sortie" pour plus de détails sur la redirection de la sortie).

```
darkstar:~$ zcat /tmp/large_file.gz
Wed Aug 26 10:00:38 CDT 2009
Slackware 13.0 x86 is released as stable!  Thanks to everyone who helped
make this release possible -- see the RELEASE_NOTES for the credits.
The ISOs are off to the replicator.  This time it will be a 6 CD-ROM
32-bit set and a dual-sided 32-bit/64-bit x86/x86_64 DVD.  We're taking
pre-orders now at store.slackware.com.  Please consider picking up a copy
to help support the project.  Once again, thanks to the entire Slackware
community for all the help testing and fixing things and offering
```

```
suggestions during this development cycle.
```

bzip2

Une alternative à ***gzip*** est l'utilitaire de compression ***bzip2***(1) qui fonctionne presque de la même façon. L'avantage de ***bzip2*** est qu'il bénéficie d'une plus grande capacité de compression. Malheureusement, la réalisation d'une plus grande capacité de compression est un processus lent et gourmand en temps processeur, de sorte ***bzip2*** est beaucoup plus lent comparativement à d'autres alternatives.

XZ/LZMA

L'utilitaire de compression récemment ajouté à Slackware est ***xz***, qui implémente l'algorithme de compression LZMA. Cela est plus rapide que ***bzip2*** et offre souvent aussi une meilleure compression. En fait, son mélange de vitesse et de puissance de compression l'a conduit à remplacer ***gzip*** en tant que le schéma de compression de choix pour Slackware. Malheureusement, xz n'a pas de page de manuel, au moment de la rédaction de ces lignes. Pour afficher les options disponibles, utilisez l'argument *–help*. La compression de fichiers se fait avec l'oargument *-z*, et la décompression avec- *-d*.

tar

Nous savons maintenant comment compresser des fichiers en utilisant toutes sortes de programmes, mais aucun d'entre eux n'est capable d'archiver des fichiers de la façon dont ***zip*** le fait. Pas pour longtemps. L'archiveur de bandes ou ***tar***(1)est le programme le plus fréquemment utilisé dans les archives Slackware. Comme les autres programmes d'archivage, ***tar*** génère un nouveau fichier qui contient d'autres fichiers et répertoires. Il ne compresse pas le fichier généré (souvent appelé un “tarball”) par défaut, mais la version de ***tar*** incluse dans Slackware prend en charge une variété de formats de compression, y compris ceux mentionnés ci-dessus.

Utiliser ***tar*** peut être aussi simple ou aussi compliqué que vous le souhaitez. En règle générale, la création d'une archive tar se fait avec les options *-cvzf*. Analysons ces options en détails.

Options de tar

Option	Signification
c	Créer une archive tar
x	Extraire le contenu d'une archive tar
t	Afficher le contenu d'une archive tar
v	Afficher plus de détails
z	Utiliser la compression gzip
j	Utiliser la compression bzip2
J	Utiliser la compression LZMA
p	Conserver les permissions

tar est plus exigeant concernant l'ordre des options que la plupart des autres programmes. L'option *-f* doit être présent lors de la lecture ou de l'écriture dans un fichier, par exemple, et doit être suivi par

le nom du fichier. Considérons les exemples suivants.

```
darkstar:~$ tar -xvzf /tmp/tarball.tar.gz
darkstar:~$ tar -xvfz /tmp/tarball.tar.gz
```

Le premier exemple ci-dessus fonctionne comme souhaité, mais le second échoue parce que ***tar*** a essayé d'ouvrir le fichier z au lieu de `/tmp/tarball.tar.gz`.

Maintenant que nous savons comment fonctionnent les options, considérons quelques exemples de la façon de création d'archives tar et d'extraction. Comme présenté auparavant, l'option *-c* est utilisée pour créer des archives tar et *-x* extrait leur contenu. Toutefois, si nous voulons créer ou extraire un fichier compressé archive cependant, nous devons aussi préciser la méthode de compression correcte à utiliser. Naturellement, si nous ne voulons pas du tout compresser l'archive , nous pouvons ne pas utiliser ces options. La commande suivante crée une nouvelle archive en utilisant l'algorithme de compression ***gzip***. Bien que ce n'est pas une exigence stricte, c'est une bonne pratique que d'ajouter l'extension .tar à toutes les archives tar aussi bien que l'extension utilisée par l'algorithme de compression.

```
darkstar:~$ tar -czf /tmp/tarball.tar.gz /tmp/tarball/
```

Lecture de documents

Traditionnellement, les systèmes d'exploitation UNIX et leurs dérivés comportent des fichiers texte que les utilisateurs vont vouloir lire de lire à un moment donné. Naturellement, il y a plusieurs manières de lire ces fichiers, et nous allons vous montrer les plus courantes.

Autrefois, si vous vouliez juste voir le contenu d'un fichier (n'importe quel fichier, que ce soit un fichier texte ou un programme binaire) vous pouviez utiliser ***cat***(1) pour les visualiser. ***cat*** est un programme très simple, qui prend un ou plusieurs fichiers, les concatène (d'où le nom) et les envoie sur la sortie standard, qui est habituellement l'écran de votre terminal. Ceci était pratique lorsque le fichier était petit et ne dépassait pas la taille de l'écran, mais ne convient pas pour les fichiers plus importants sans disposer de moyen intégré pour se déplacer à l'intérieur du document et lire un paragraphe à la fois. Aujourd'hui, ***cat*** est encore assez largement utilisé, mais surtout dans les scripts ou pour combiner deux ou plusieurs fichiers en un seul.

```
darkstar:~$ cat /etc/slackware-version
Slackware 14.0
```

Étant donné les limites de ***cat***, des personnes très intelligentes se sont mises à travailler sur une application leur permettant de lire des documents par page par page. Naturellement, de telles applications ont commencé à être connu sous le nom de "*pagers*". L'un des premiers d'entre eux était ***more***(1), nommé ainsi parce qu'il laissait voir "plus" du fichier quand vous le vouliez.

more

more affiche les premières lignes d'un fichier texte jusqu'à ce que votre écran soit plein, et ensuite se met en pause. Une fois que vous avez lu cet écran, vous pouvez passer à la ligne suivante en appuyant sur ENTER ou un écran entier en appuyant sur SPACE, ou à un nombre spécifique de lignes

en tapant un nombre suivi de la barre d'espacement SPACE. ***more*** est également capable de rechercher des mots-clés dans un fichier texte. Une fois que vous avez avez affiché un fichier dans ***more***, appuyez sur la touche /, et entrez un mot-clé. En appuyant sur ENTER, le texte défile jusqu'à la prochaine occurence du mot-clé.

Il s'agit clairement d'une amélioration majeure par rapport à ***cat***, mais il reste toujours quelques défauts gênants, ***more*** n'est pas en mesure de faire défiler vers le haut,, pour vous permettre de lire quelque chose que vous auriez pu manquer, lorsque'un fichier lui est redirigé en entrée. Lla fonction de recherche ne met pas en évidence les résultats. Il n'y a pas défilement horizontal, et ainsi de suite. Il est clair qu'une meilleure solution est possible.

En fait, les versions modernes de ***more***, telle que celle présente dans Slackware, possedent une fonction **back** disponible via la touche b. Cependant, cette fonction n'est disponible que lorsqu'un fichier est directement ouvert par ***more*** ; elle n'est pas disponible lorsqu'un fichier est redirigé vers l'entrée de ***more***.

less

Afin de pallier aux lacunes de ***more***, un nouveau pager a été développé et ironiquement surnommé ***less***(1). ***less*** est un très puissant pager qui possède toutes les fonctionnalités de ***more*** avec en plus de nombreuses fonctionnalités supplémentaires. Pour commencer, ***less*** vous permet d'utiliser les touches fléchées pour contrôler les déplacements au sein du document.

En raison de sa popularité, de nombreuses distributions Linux ont commencé à exclure ***more*** en faveur de ***less***. Slackware incorpore les deux. En outre, Slackware comprend également un pré-processeur fort utile pour ***less***, appelé `lesspipe.sh`. Cela permet à un utilisateur d'exécuter ***less*** sur un certain nombre de fichiers non textes. `lesspipe.sh` va générer une sortie texte via l'exécution d'une commande sur ces fichiers, et l'afficher dans ***less***.

less fournit presque autant de fonctionnalités qu'un éditeur de texte sans pour autant être un éditeur de texte. Le déplacement ligne par ligne peut se faire comme avec ***vi*** via les touches j et k, ou avec les touches fléchées, ou ENTER. Dans le cas où un fichier est trop grand pour tenir sur un seul écran, vous pouvez même faire défiler horizontalement avec les touches fléchées gauche et droite. La touche g vous conduit au début du fichier, alors que G vous dirige à la fin.

La recherche est effectuée tout comme dans ***more***, en tapant sur la touche / suivi de l'expression à rechercher. Mais remarquez comment les résultats de la recherche sont mis en évidence pour vous, et en tapant N vous allez à la prochaine occurrence du résultat tandis que la N vous emmène à l'occurrence précédente.

Tout comme ***more***, les fichiers peuvent être ouverts directement dans ***less*** ou redirigés vers son entrée :

```
darkstar:~$  less
/usr/doc/less:/README
darkstar:~$  cat
/usr/doc/less:/README
/usr/doc/util-linux:/README | less
```

Il y a plus de fonctionnalités offertes par ***less***. À l'intérieur de l'application, taper h pour une liste complète de commandes.

Création de liens

Les liens sont une moyen de référencer un même fichier avec plus d'un nom. En utilisant l'application ***ln***(1), un utilisateur peut faire référence à un fichier avec plus d'un nom. Les deux fichiers ne sont pas des copies carbone l'un de l'autre, mais sont plutôt exactement le même fichier, juste avec un nom différent. Pour supprimer le fichier entièrement, tous ses noms doivent être supprimés. (C'est en fait le résultat de la façon dont les outils tels que ***rm*** et autres fonctionnent. Plutôt que de supprimer le contenu du fichier, ils suppriment tout simplement la référence au fichier, libérant l'espace pour être réutilisé. ***ln*** créera une seconde référence ou “lien” à ce fichier.)

```
darkstar:~$ ln /etc/slackware-version foo
darkstar:~$ cat foo
Slackware 14.0
darkstar:~$ ls -l /etc/slackware-version foo
-rw-r--r-- 1 root root 17 2007-06-10 02:23 /etc/slackware-version
-rw-r--r-- 1 root root 17 2007-06-10 02:23 foo
```

Il existe un autre type de lien, appelé le lien symbolique. Les liens symboliques, plutôt que d'être une autre référence au même fichier, sont en réalité un type spécial de fichier en eux-mêmes. Les liens symboliques pointent vers un autre fichier ou un répertoire. Le principal avantage des liens symboliques, c'est qu'ils peuvent référencer des répertoires de même que des fichiers. Ils peuvent aussi s'étendre sur plusieurs systèmes de fichiers. Ils sont créés avec l'argument *-s*.

```
darkstar:~$ ln -s /etc/slackware-version foo
darkstar:~$ cat foo
Slackware 140
darkstar:~$ ls -l /etc/slackware-version foo
-rw-r--r-- 1 root root 17 2007-06-10 02:23 /etc/slackware-version
lrwxrwxrwx 1 root root 22 2008-01-25 04:16 foo -> /etc/slackware-version
```

Lorsque vous utilisez des liens symboliques, n'oubliez pas que si le fichier original est supprimé, votre lien symbolique est inutile ; il pointe tout simplement vers un fichier qui n'existe plus.

Navigation

Chapitre précédent : Démarrage

Chapitre suivant : BASH ou le Bourne Again Shell

Sources

- Source originale : http://www.slackbook.org/beta

- Publication initiale d'Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traduction initiale de escaflown
- Traduction de Ellendhel

slackbook, shell, archive, filesystem

1)

mkdir correspond à ***m**a**k**e **dir**ectory*, c'est à dire “créer un répertoire”. Beaucoup de commandes débutant par *mk* servent à créer quelque chose. NdT.