

Le système d'exploitation sur votre Raspberry Pi est une version de Linux. Selon toute probabilité, il ressemble un peu à Windows ou - plus vraisemblablement - au système d'exploitation d'un Mac. Il dispose d'une Interface Utilisateur Graphique (GUI) qui vous permet d'interagir avec vos dossiers et fichiers par le double-clic, le clic droit ou gauche et le glisser-déposer. Si vous voulez ouvrir un programme, vous le cherchez dans un menu appelé généralement "Applications".

Tout cela est très bien, mais il y a une autre façon d'interagir avec Linux : utiliser l'**Interface de Ligne de Commande (CLI)**. Avec la CLI, il n'y a pas d'images, rien à cliquer non plus. Pour faire faire quelque chose à votre ordinateur, vous devez taper une commande de texte correctement construite. Souvent, vous saurez que votre commande a été exécutée avec succès parce que votre ordinateur ne retourne rien. Si vous avez une réponse, quelque chose s'est mal passé et il vous donne un message d'erreur.

Cela semble représenter beaucoup de travail. Souhaitons nous vraiment nous soucier de cela ? Oui, nous le devons ! Au démarrage, par défaut, certaines versions de Linux se lance directement dans l'interface CLI. Pour ouvrir l'interface graphique, avec toutes ses belles fenêtres et ses menus, vous devez alors taper la commande "**startx**" et appuyez sur Entrée. Mais ce n'est pas la seule raison pour vous familiariser avec l'interface CLI.

Avec la ligne de commande, vous pouvez faire des choses que vous ne pourriez pas faire en utilisant l'interface graphique et vous pouvez facilement faire des choses qui sont difficiles à faire avec l'interface graphique. Par rapport à l'interface graphique, la ligne de commande n'utilise pratiquement pas de traitement ou de puissance graphique, elle est donc idéal si vous avez un travail lourd à faire pour votre ordinateur. Et si vous vous connectez à distance à un Raspberry Pi qui ne dispose pas de son propre moniteur, alors vous devrez utiliser l'interface CLI.

Si vous êtes nouveau sur Linux, vous ne serez surement pas familier avec la ligne de commande en général et, qui plus est, avec les commandes Linux en particulier (elles sont parfois semblables à leurs équivalents Windows mais elles sont plus souvent totalement différentes). Ce chapitre est une introduction rapide à la magie de la ligne de commande. Il contient tout ce dont vous avez besoin pour commencer.

## Leçon 6.1 : Les commandes sont tout simplement des programmes

Lorsque vous tapez du texte dans le terminal, vous exécutez des programmes. La plupart des commandes listées ici exécute des programmes qui vous donnent la possibilité de commander le système pour faire quelque chose. Lorsque vous ajouterez des programmes sur votre Raspberry Pi, vous serez en mesure d'exécuter également d'autres commandes.

Si vous entrez une commande et qu'elle ne fonctionne pas, cela peut signifier que le programme n'est pas encore installé. Par exemple, si vous essayez de créer un nouveau fichier texte appelé "Mes commandes Linux" en utilisant l'éditeur de texte nano, vous devrez taper :

```
sudo nano "Mes commandes Linux"
```

Mais il se pourrait qu'au lieu d'ouvrir le nouveau fichier texte, votre système renvoie le message d'erreur :

```
bash: nano: command not found
```

Cela signifie simplement que vous n'avez pas installé nano. Cependant, la plupart des commandes décrites sur ces pages sont des commandes de type «nettoyage» qui existent par défaut, de sorte que vous ne devriez pas avoir ce problème. Chaque programme a un auteur qui est précisé au bas de la page principale.

Ne vous inquiétez pas trop sur les commandes utilisées plus loin - je vous expliquerais tout lorsque nous avancerons.

## L'interpréteur de commande "Bash"

Pour utiliser la CLI, vous devez connaître les commandes. Le jeu de commandes par défaut est contenu dans l'interpréteur de commande que vous utilisez. Il y a beaucoup d'interpréteur de commande, mais le principal sur Raspberry Pi s'appelle "Bash". Bash est l'interpréteur de ligne de commande par défaut sur la plupart des versions de Linux, ainsi que Mac OS, il est donc utile de l'apprendre.

Ce chapitre couvre la plupart des commandes les plus courantes et utiles. Si vous ne trouvez pas la commande dont vous avez besoin, essayez de regarder en ligne. Il y a beaucoup de bons guides sur Internet pour l'utilisation de la ligne de commande Linux.

Si vous connaissez la commande que vous souhaitez utiliser, mais ne savez pas exactement comment l'utiliser - par exemple, vous ne connaissez pas exactement la bonne «syntaxe» - vous pouvez utiliser la commande Linux "man" (manuel). Sur la ligne de commande, vous pouvez taper :

```
$ man <commande>
```

Notes:



### Conseils...

La «syntaxe» d'une commande signifie la façon dont elle doit être écrite pour la faire fonctionner. Cela signifie que l'ordre des mots, ainsi que les signes important de ponctuation sont également nécessaires.

**man**

**Affiche des informations sur la "cible", en provenance des manuels de référence en ligne.**

Ou, vous pouvez utiliser la commande "info" :

```
$ info <commande>
```

**info**

**Affiche la documentation en ligne sur la cible.**

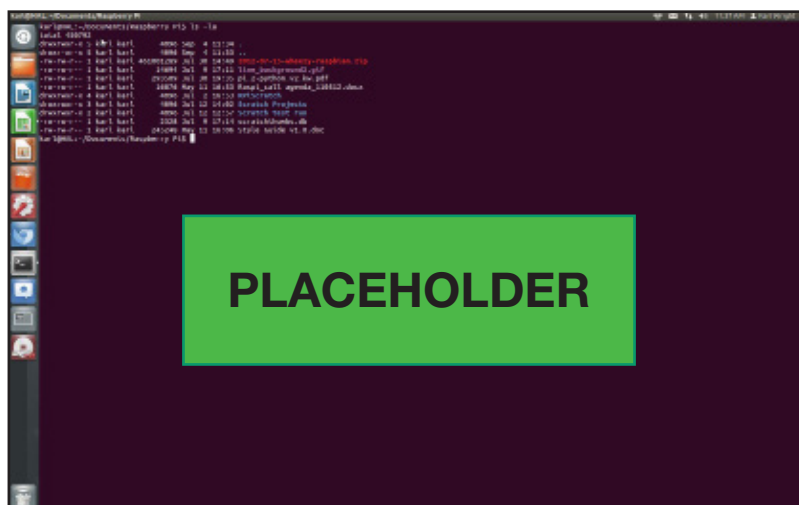
Dans les exemples ci-dessus, remplacez <commande> avec la commande dont vous voulez en savoir plus. Message d'avertissement : l'explication donnée par «man» est parfois un peu formel et très, très détaillé. On pourrait presque dire : «Si vous pensez que vous connaissez une commande alors consultez la fonction "man" de cette commande dans le but de vérifier que vous ne la connaissez pas totalement."

Pour en savoir plus sur Bash, jetez un oeil à sa définition Wikipedia :

[http://en.wikipedia.org/wiki/Bash\\_%28Unix\\_shell%29](http://en.wikipedia.org/wiki/Bash_%28Unix_shell%29)

*Taper "man apt-get" dans la ligne de commande pour retourner toutes les informations sur la commande «apt-get» et savoir comment l'utiliser.*

*"Man" est le raccourci de manuel*



Les commandes prennent cette forme :

**<Commande> | <Options> | <Paramètres> | <Cible>**

Dans les guides, comme celui-ci, les crochets "<" et ">" sont souvent utilisés pour indiquer la zone que prendrait une chaîne de texte qui est tapée dans la ligne de commande.

La ligne horizontale "|" est utilisée pour signifier «ou». Perdu ? Bon, regardons l'exemple ci-dessus. Nous voyons que les commandes peuvent être utilisées elle-même ou elles peuvent être utilisées avec :

- **Option** : lettres simples, précédée d'un tiret, qui définit ce que fait la commande
  - **Paramètres** : Elements dont la commande a besoin de savoir pour travailler
  - **Cible** : l'élément (comme un fichier) auquel la commande sera appliqué
- Voyons un exemple. Nous allons commencer avec la commande "ls", que vous pouvez utiliser pour afficher la liste du contenu d'un dossier.

```
ls -l /home/brian
```

Commande Option Cible

Cette commande indique de faire une liste, au format long, du contenu du répertoire "/home/brian". La commande est "ls", l'option "-l" indique à Linux que vous voulez la liste au format long, et la cible de cette commande est le répertoire «/home/brian». Dans cet exemple, il n'existe pas de paramètres, de sorte qu'une partie de la commande est tout simplement ignorés.

### ls

**Liste le contenu d'un répertoire.**

Pour un exemple un peu plus détaillé, penchons-nous sur la commande "mount", qui indique à Linux d'intégrer un nouveau système de fichiers (comme un CD ou DVD) dans sa structure de système fichiers afin que vous puissiez le parcourir.

```
mount -F smbfs //workgroup;fred:toto@192.168.1.99/homes /mnt/net
```

Commande Options Paramètres Cible

Cette commande indique au système d'exploitation d'utiliser le nom d'utilisateur "fred" et le mot de passe "toto" pour connecter le disque partagé appelé "homes" sur le serveur Windows à l'adresse 192.168.1.99 (paramètres) pour apparaître dans l'arborescence des répertoires au point "/mnt/net" (cible) en utilisant le système de fichier Server Message Block (Option -F).

### mount

**Crée une structure de fichier à un nouvel emplacement.**

## Navigation dans le système de fichiers en utilisant "cd"

Le système de fichiers sous Linux est hiérarchique avec des sous répertoires (souvent appelés «dossiers») dans une «arborescence». La racine de la structure de répertoire est désigné par le symbole "/" et les répertoires suivant "/" se réfèrent à l'utilisation de «chemins», tout comme les URL dans un navigateur Web.

Pour accéder à une place particulière dans la structure de répertoire, vous devez utiliser la commande "cd", qui signifie "change directory", suivi de son emplacement dans l'arborescence. Par exemple, la commande suivante vous amène au dossier «Documents» de Brian dans son répertoire personnel.

```
cd /home/brian/Documents
```

Si vous voulez remonter dans l'arborescence, utilisez la commande suivante :

```
cd ..
```

Par exemple, si vous travaillez dans le dossier "/home/brian/Documents/projet", la commande "cd .." vous renvoie à «/home/brian/Documents». Assurez-vous de laisser un espace entre les lettres «cd» et les deux points, sinon la commande ne fonctionnera pas.

Pour en savoir plus sur la commande "cd" voir sa définition Wikipedia :  
[http://en.wikipedia.org/wiki/Cd\\_%28command%29](http://en.wikipedia.org/wiki/Cd_%28command%29)

### cd

**Change le répertoire de travail courant.**

Notes:



### Conseil...

*Le nom des fichiers et des répertoires (dossiers) sont sensibles à la casse sous Linux, le fichier "my\_stuff" est différent du fichier "My\_stuff".*

La commande "ls" répertorie tous le contenu du répertoire ou vous travaillez. Il y a toute une gamme d'options que vous pouvez utiliser avec "ls" pour faire afficher exactement les fichiers qui vous intéressent et toutes les informations utiles sur ces fichiers. Nous avons répertorié quelques-unes de ces options ici:

- **"ls"**, la commande sans option, répertorie tous les fichiers dans le répertoire courant.
- **"ls -l"** affiche la liste des fichiers et l'information en version longue de chaque fichier ou répertoire. La sortie peut être sous forme de code couleur en fonction des préférences de l'éditeur de commande que vous avez définis.
- **"ls -R"** demande une liste "récursive" - qui est, une liste mentionnant le contenu du sous-répertoires ainsi que de ce répertoire.
- **"ls -A"** force le système à montrer les fichiers «cachés». Les fichiers cachés ont des noms qui commencent par un point, ce qui n'est généralement pas visible lorsque vous faites une liste de répertoire normal.

Voyons deux exemples. Dans le premier, nous utilisons la commande "ls" sans options.

```
$ ls
```

```
An_Gott_und_meine_Mutter.mid Domestic Programming Test.mid  
An_Gott_und_meine_Mutter.mscz Engineering Quantum Physics Tutoring  
appliances FluidR3_GM.ins School Windaz
```

Dans le second cas, nous utilisons l'option "-l". Ceci affiche les fichiers dans le format long, en vous disant, entre autres choses, la taille, le nom du propriétaire et la sécurité pour chaque fichier.

```
$ ls -l
```

```
total 336  
-rw-rw-r--. 1 brian brian 2429 Apr 2 20:27 An_Gott_und_meine_Mutter.mid  
-rw-rw-r--. 1 brian brian 4085 Apr 2 19:52 An_Gott_und_meine_Mutter.mscz  
drwxrwxr-x. 4 brian brian 4096 Apr 2 20:38 appliances  
-rw-rw-r--. 1 brian brian 10919 Apr 2 19:52 brotplot.odt
```

Ne vous inquiétez pas, nous verrons les détails de ces informations plus tard.

Pour en savoir plus sur la commande "ls", jetez un oeil à sa définition Wikipedia :  
<http://en.wikipedia.org/wiki/Ls>

## Changer le propriétaire d'un repertoire ou d'un fichier

Dans la longue liste ci-dessus, les fichiers sont la propriété de Brian et sont également dans le groupe de Brian. Que faire si nous voulons changer la propriété de certains de ces dossiers ? Nous allons utiliser la commande "chown".

Nous allons utiliser le fichier "brotplo.t.odt" en exemple. Disons que nous voulons faire de Fred le propriétaire du fichier. Nous pourrions utiliser la commande:

```
$ chown fred brotplo.t.odt
```

Si nous lançons la commande «ls-l» de nouveau, nous allons maintenant voir :

```
-rw-rw-r--. 1 fred brian 10919 Apr 2 19:52 brotplo.t.odt
```

Le fichier est toujours dans le groupe Brian mais il est maintenant la propriété de Fred. Mais que faire si nous voulons faire de l'utilisateur "Toto" le propriétaire du fichier et également déplacer le fichier dans le groupe "Toto". Nous devons utiliser la commande:

```
$ chown toto:toto brotplo.t.odt
```

Si nous utilisons "ls-l" de nouveau pour voir les infos du fichier, nous allons voir :

```
-rw-rw-r--. 1 toto toto 10919 Apr 2 19:52 brotplo.t.odt
```

Pour en savoir plus sur la commande "chown", voir sa définition Wikipedia :  
<http://en.wikipedia.org/wiki/Chown>

### **chown**

**Modifie la propriété d'un ou plusieurs fichiers.**

Tant que nous y sommes, jetez un oeil à "chgrp" pour voir la différence :  
<http://en.wikipedia.org/wiki/Chgrp>

### **chgrp**

**Modifie le groupe d'un ou plusieurs fichiers.**

## Modifier les accès aux répertoires et fichiers

Lorsque vous utilisez l'option "-l" pour voir les informations complètes du contenu d'un répertoire, les résultats que vous voyez sont tous précédés par une série de lettres et de tirets. Par exemple :

```
-rw-rw-r--. 1 fred etudiants 10919 Apr 2 19:52 brotplot.odt
```

Le premier caractère de la séquence, dans ce cas, un tableau de bord, nous dit ce qu'est l'objet. Les caractères les plus couramment utilisés sont les suivants :

- d** Indique un répertoire
- l** Indique un lien
- Indique un fichier

Dans notre exemple ci-dessus, il commence par un tiret, et nous savons donc que c'est un fichier. Dans ce cas, il s'agit en fait d'un fichier de traitement de texte.

Les neuf caractères qui suivent, décrivent les droits d'accès pour le propriétaire (utilisateur), le groupe et le reste (tout le monde qui peut être en mesure d'accéder au fichier), dans cet ordre, chacun avec trois paramètres. Les caractères utilisés ici sont les suivants :

- r** Accès en lecture
- w** Accès en écriture
- x** Autorisation d'exécuter le programme
- Pas d'accès de ce type.

Donc, si les trois groupes avaient tous les trois ces types d'accès, on verrait "rwxrwxrwx". Cependant, chaque fois que quelqu'un n'a pas un de ces types d'accès, sa lettre est remplacée par un tiret.

Si l'on regarde à nouveau notre exemple ci-dessus, nous pouvons voir que :

- Brian a accès en lecture et écriture sur le fichier.
- Les membres du groupe nommé "etudiants" ont également l'accès en lecture et écriture.
- Tout le monde (le "monde") a seulement accès en lecture au fichier.

L'indicateur «x» n'apparaît pas du tout. Rappelez-vous, x représente la capacité d'exécuter un fichier. Si l'indicateur x n'est pas présent, le fichier ne peut pas être exécuté comme un programme.

Mais que faire si nous voulons changer les permissions du fichier ? Il y a différentes manières d'utiliser la commande "chmod" pour ce faire, mais le plus facile est probablement d'utiliser ces groupes :



- u** pour l'utilisateur / propriétaire
- g** pour le groupe
- o** pour les autres (tout le monde)
- a** pour tous les trois

Maintenant, nous allons essayer la commande "chmod" :

```
chmod o+w brotplot.odt
```

Cela signifie ajouter l'accès en écriture pour le groupe «autres» à «brotplot.odt». L'ajout est indiqué par le signe "plus".

```
chmod a-r brotplot.odt
```

Cela signifie de retirer l'accès en lecture pour tous les trois (utilisateur, groupe et autres) de «brotplot.odt». La suppression est indiqué par le signe "moins".

```
chmod u+x brotplot.odt
```

Cela signifie ajouter la possibilité d'exécution pour l'utilisateur/propriétaire au fichier "brotplot.odt".

Pour en savoir plus sur la commande "chmod", voir sa définition Wikipedia :  
<http://en.wikipedia.org/wiki/Chmod>

## **chmod**

***Change le mode d'accès (permissions) d'un ou plusieurs fichiers.***

La commande **"sudo"** introduit le "super utilisateur" ou "utilisateur root". Le terme "root" est le nom de l'administrateur principal dans un "Unix-like" du système, tel que Linux. Il ya beaucoup de commandes que seul l'utilisateur root peut exécuter.

En fonction de la version de Linux que vous utilisez, vous devrez soit vous connecter en tant qu'utilisateur root ou précéder votre commande de "sudo". La distribution Debian de Linux par défaut, par exemple, n'a pas de mot de passe root. Donc, vous devrez utiliser la commande "sudo".

En utilisant la commande «sudo», vous dites : «Faites la commande suivante en tant qu'utilisateur root." Quand vous faites cela, vous serez invité à fournir votre mot de passe et, si vous avez les autorisations du système pour un utilisateur root (communément appelé "étant dans le groupe admin "), alors la commande sera exécutée. Si vous n'avez pas les permissions root, vous obtiendrez un message d'erreur.

Regardons l'exemple d'ajout d'un nouvel utilisateur sur votre système, quelque chose que seul l'utilisateur root peut faire. Sur Debian, la commande ressemble à ceci:

```
sudo adduser brian
```

Cela va démarrer un script qui vous permet de créer un nouvel utilisateur appelé "brian". Si jamais vous essayez de faire quelque chose dans la ligne de commande et que vous avez le message d'erreur, "Seul l'utilisateur root peut ...", alors essayez la commande sudo qui peut résoudre votre problème.

Pour en savoir plus sur la commande "sudo", voir la définition Wikipedia :  
<http://en.wikipedia.org/wiki/Sudo>

### **sudo**

***Si vous disposez des autorisations appropriées, exécutez la commande suivante en tant que super-utilisateur.***

## Devenir un utilisateur administrateur

Il y a une autre façon de faire les choses comme utilisateur root et c'est à l'aide de la commande "su", qui signifie "substitut utilisateur" (ou super-utilisateur). L'invocation de la commande "su" signifie "devenir l'utilisateur root".

Comme précédemment, il vous sera demandé d'entrer votre mot de passe pour vous assurer que vous disposez des droits du super-utilisateur. Une fois l'authentification réussie avec votre mot de passe, toutes les commandes sont exécutées en tant que superutilisateur.

Dans certains systèmes, cependant, vous ne serez pas en mesure d'utiliser la commande "su" car, par défaut, l'utilisateur root n'est pas activé. Dans ce cas, vous devrez activer l'utilisateur root, utilisez alors cette commande :

```
passwd root
```

Vous serez invité à entrer et confirmer un mot de passe root. En supposant que vous ayez entré le même mot de passe deux fois, vous avez maintenant un utilisateur root et vous pouvez exécuter la commande :

```
su -
```

Lorsque vous entrez les mots de passe dans les systèmes de type Unix, la zone reste vide : pas de points ou d'astérisques substitués aux caractères de votre mot de passe. Ne vous laissez pas abuser, le mot de passe est entré quand même.

Enfin, un mot d'avertissement : si vous prenez l'habitude de faire tout avec l'utilisateur root, vous finirez par faire quelque chose de désastreux. Par exemple, vous pourriez invoquer :

```
rm -r -f /*
```

Ceci va supprimer tous les fichiers de l'ensemble du système.

Ou vous pourriez taper malencontreusement :

```
$ rm -r -f ./*
```

Ce serait tout simplement la suppression des fichiers dans le répertoire courant. Et vous ne recevez pas les avertissements habituels des autres utilisateurs en tant que root. La première chose que vous saurez au sujet de votre erreur, c'est quand tous vos fichiers auront soudainement disparu.

N'utilisez les privilèges root que lorsque vous avez vraiment besoin d'eux.

Pour en savoir plus sur la commande "su". voir sa définition Wikipedia:

**[http://en.wikipedia.org/wiki/Su\\_%28Unix%29](http://en.wikipedia.org/wiki/Su_%28Unix%29)**

## **SU**

***Devenir un super utilisateur.***

## **passwd**

***Créer ou changer un mot de passe associé avec l'utilisateur identifié.***



### **Conseils...**

*Ne jamais utiliser ces commandes lors de l'utilisation en tant que super-utilisateur!*

## Leçon 6.4 : Création et suppression de fichiers et répertoires

Notes:

Enfin, nous allons voir comment créer et détruire des fichiers et répertoires, ainsi que l'utilisation de la commande `mount` de plus près.

### Créer un nouveau fichier vide

Pour créer un nouveau fichier vide, allez à l'endroit où vous voulez que le fichier soit et tapez :

```
$ touch <nom du fichier>
```

La commande "touch" met à jour le "dernier accès" d'un fichier au niveau de l'heure et la date, mais si un tel fichier n'existe pas encore, il va créer un nouveau fichier avec ce nom, et une taille de fichier de 0.

Pour en savoir plus sur la commande "touch", voir sa définition Wikipedia :

[http://en.wikipedia.org/wiki/Touch\\_\(Unix\)](http://en.wikipedia.org/wiki/Touch_(Unix))

#### **touch**

**Met à jour le date d'accès du fichier nommé et l'heure de modification (et la date) à l'heure et la date. Si un fichier n'existe pas, un nouveau fichier sera créé avec une taille de fichier de 0.**

### Créer un nouveau répertoire vide

Pour créer un nouveau répertoire, allez à l'endroit où vous voulez que le répertoire soit et tapez :

```
$ mkdir <nom du répertoire>
```

Pour en savoir plus sur la commande "mkdir", voir sa définition Wikipedia :

<http://en.wikipedia.org/wiki/Mkdir>

#### **mkdir**

**Créez un ou plusieurs répertoires dans l'emplacement actuel.**

### Supprimer un fichier

Pour supprimer un fichier, allez à l'emplacement de ce fichier et tapez :

```
$ rm <nom du fichier>
```

Pour en savoir plus sur la commande "rm", voir sa définition Wikipedia :

[http://en.wikipedia.org/wiki/Rm\\_%28Unix%29](http://en.wikipedia.org/wiki/Rm_%28Unix%29)

#### **rm**

**Enlève (supprime) un ou plusieurs fichiers à l'emplacement actuel.**

## Suppression d'un répertoire

Pour supprimer un répertoire, rendez-vous à son emplacement et taper :

```
$ rm -r -f <nom du répertoire>
```

Celui-ci reprend la commande "rm" que nous avons utilisé avant, mais les commutateurs supplémentaires lui indique de supprimer le répertoire, son contenu ainsi que les sous-répertoires et leur contenu.

L'option "-r" signifie "récursif", qui - dans le cas d'un répertoire - supprime tout le répertoire et tout son contenu, y compris les sous-répertoires. Soyez bien sûr de ce que vous voulez faire avant de l'utiliser - il n'y a pas d'option d'annulation !

L'option "-f" indique que l'action est «forcé» donc, le programme supprimera les fichiers protégés en écriture sans demander confirmation. C'est également dangereux, alors soyez très prudent lorsque vous l'utilisez aussi.

Une méthode plus sûre, mais moins puissante est d'utiliser la commande "rmdir"

```
$ rmdir <nom du répertoire>
```

Cela permettra également de supprimer des répertoires, mais seulement si ils sont vides.

### **rmdir**

**Enlève (supprime) un ou plusieurs répertoires dans l'emplacement actuel, à condition qu'ils soient vides.**

## Se connecter à un appareil ou un système de fichier

La commande "mount" vous permet de connecter un système Unix à des appareils externes. Il n'y a pas de lecteur "C", comme sous Windows. Qu'est-ce qui se passe dans Linux lorsqu'un dispositif est "monté" quelque part dans le système de fichier. Lorsque vous accédez à cet zone, les articles proposés par le périphérique apparaîtront à ce point.

Il s'agit d'une commande complexe. Les commutateurs, les options et les cibles de la commande de montage varie selon le protocole du système à monter. Certaines choses seront "auto-mount". C'est pourquoi, lorsque vous branchez une carte SD dans un système Linux moderne, le système de fichier saura automatiquement le monter.

Le montage manuel nécessite un "point de montage". Ce qui signifie un répertoire qui sera utilisé avec le dispositif monté lors de son montage. Souvent, c'est quelque part dans le répertoire "/mnt/". En règle générale, avant de monter les nouveaux médias, vous devez d'abord vous assurer qu'il y a un point de montage. S'il n'y en a pas, vous devez créer un répertoire à l'endroit nécessaire : par exemple ...

```
$ mkdir /mnt/netfolder
```

Vous devez également vous assurer que vous utilisez les options nécessaires, les paramètres et les répertoires. Par exemple, à partir de l'exercice précédent :

```
mount -F smbfs //workgroup;fred:toto@192.168.1.99/homes /mnt/net
```

Ceci, comme nous l'avons vu précédemment, indique au système de monter le disque partagé appelé "homes" sur le serveur Windows à l'adresse 192.168.1.99, dans l'arborescence au point "/mnt/net", en utilisant le système de fichier Server Message Block (Option -F).

Pour en savoir plus sur la commande "mount", voir sa définition Wikipedia :

**[http://en.wikipedia.org/wiki/Mount\\_%28Unix%29](http://en.wikipedia.org/wiki/Mount_%28Unix%29)**

## Poursuite de l'apprentissage

Les commandes présentées ici sont seulement une petite sélection de toutes les commandes disponibles sur votre système. Vous pouvez trouver une liste complète des commandes de Linux sur le site O'Reilly :

**<http://www.oreillynet.com/linux/cmd>**

**OpenSSH** est une application qui vous permet un accès sécurisé à des systèmes Linux distant via le réseau. Vous pouvez utiliser OpenSSH simplement pour le partage de fichiers sécurisé. Mais il vous permet également de vous connecter à un système et de le contrôler par le réseau, même en utilisant l'interface utilisateur graphique, comme si vous étiez assis en face de la machine.

L'installation par défaut de Linux sur votre Raspberry Pi devrait avoir "**SSH daemon**" en cours d'exécution. Cela signifie que votre Raspberry Pi est à l'écoute sur le port 22 pour un ordinateur distant demandant d'établir une connexion avec lui. Dans votre cas, cela signifierait sûrement votre pc de bureau ou un ordinateur portable.

### Exécution d'un accès à distance CLI

Pour vous connecter à votre Raspberry Pi, vous aurez besoin d'un programme client SSH. Linux et Mac OS l'ont déjà d'installé. Pour Windows, vous pouvez télécharger PuTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty>). Reportez-vous au manuel de votre programme client SSH choisi pour plus d'informations sur la façon de l'installer et de l'utiliser.

Afin d'établir une connexion réussie, vous devez avoir le port 22 ouvert à la fois sur l'hôte distant et le Raspberry Pi. Vous aurez également besoin de mettre en place un utilisateur approprié sur le RPi (comme nous l'avons fait plus haut dans ce chapitre).

Si vous utilisez un client Linux ou Mac, vous pouvez indiquer ceci en ligne de commande:

```
ssh <Adresse IP du RPi> -l <Nom d'utilisateur sur le RPi>
```

<Adresse IP du RPi> doit être remplacé par l'adresse IP de votre Raspberry Pi, et <Nom d'utilisateur du RPi> doit être remplacé par le nom d'utilisateur de votre Raspberry Pi, que vous avez configuré précédemment. Le caractère entre eux est un "l" minuscule.

Vous devriez voir ceci :

```
[brian@fc16toshiba ~]$ ssh -X 192.168.1.104 -l brian
The authenticity of host '192.168.1.104 (192.168.1.104)' can't be established.
RSA key fingerprint is 26:a4:a1:ab:c2:ff:50:99:d7:e1:49:6e:f2:90:fb:90.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.104' (RSA) to the list of known hosts.
brian@192.168.1.104's password:
Linux raspberrypi 3.1.9+ #9 Mon Apr 9 20:50:36 BST 2012 armv6l
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

brian@raspberrypi:~$
```

À ce stade, vous êtes opérationnel. Tout ce que vous taperez sera ce qui se passe réellement sur votre Raspberry Pi.

Notes:

## Exécution d'un accès à distance GUI

Pour se connecter à distance à votre Raspberry Pi et utiliser son interface graphique, vous aurez besoin d'installer un **"X server"** sur votre machine. Linux dispose d'un X serveur construit pour Windows, essayez Xming (<http://www.straightrunning.com/XmingNotes>) ou Cygwin (<http://www.cygwin.com>) - vous pouvez les télécharger gratuitement. Pour le système d'exploitation Mac, allez dans "Shell> Nouvelle connexion à distance" et choisissez «Secure Shell (SSH)".

La commande suivante utilise l'option X appeler SSH pour envoyer les commandes sur le serveur X de votre hôte. Entrez la commande suivante :

```
ssh -X <adresse IP du RPi> -l <utilisateur du RPi>
```

## Qu'est ce qu'un serveur X ?

Un ordinateur sous Linux (ou tout autre système de type Unix) exécutant une interface graphique est presque certainement entrain d'exécuter "X" et utilise également un serveur X. Le X envoie des commandes au serveur X sur le genre de choses à mettre sur l'écran, et le serveur X le fait.

Cela signifie que l'interface graphique et le serveur X sont séparés. Cela signifie également que vous pouvez exécuter un programme sur le Raspberry Pi et que la sortie graphique sur l'écran du serveur X apparaisse ailleurs sur le réseau. Cela supprime une grande quantité de demande de traitement du Raspberry Pi. Et cela signifie aussi que vous pouvez simplement brancher votre Raspberry Pi sur votre réseau, plutôt que de lui équiper son propre moniteur, une souris et un clavier.



## Ouvrez un navigateur distant

Voyons ce qui se passe lorsque j'essaie d'exécuter un navigateur Web à distance. Le navigateur Web du Raspberry Pi est appelé **Midori**, donc c'est ce que nous avons besoin d'entrer à l'invite après nous être connecté à distance :

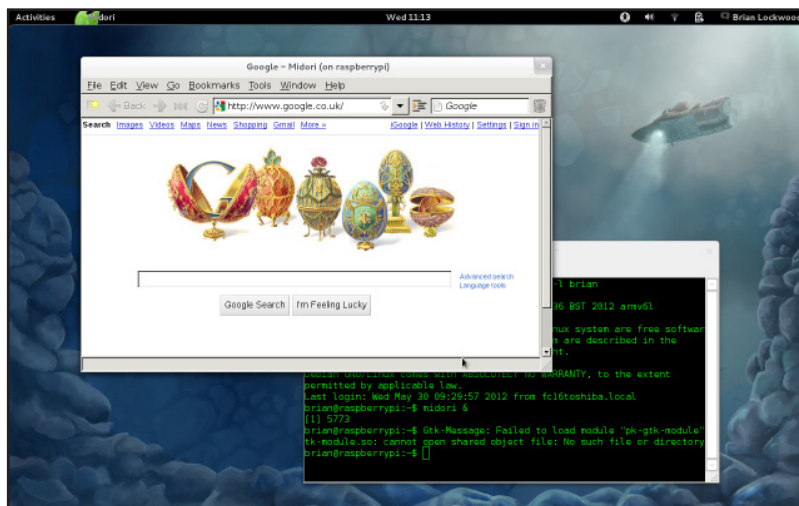
```
brian@raspberrypi:~$ midori &  
[1] 5773
```

Nous avons besoin d'inclure le symbole "&" après le nom du programme, car cela indique au serveur de le lancer comme un processus séparé, ce qui signifie que nous obtenons notre ligne de commande.

Nous obtenons la réponse "[1] 5773", qui est le numéro de processus du programme de Midori maintenant en cours d'exécution.

Parfois, vous obtiendrez des erreurs GTK, qui révèlent des lacunes de la bibliothèque de l'hôte (le RPI) ou le serveur (votre serveur X) mais le processus est assez robuste.

*Midori en cours d'exécution  
à distance sur Gnome 3 UI à  
l'aide du serveur X de  
Fedora 16.*



Notes: