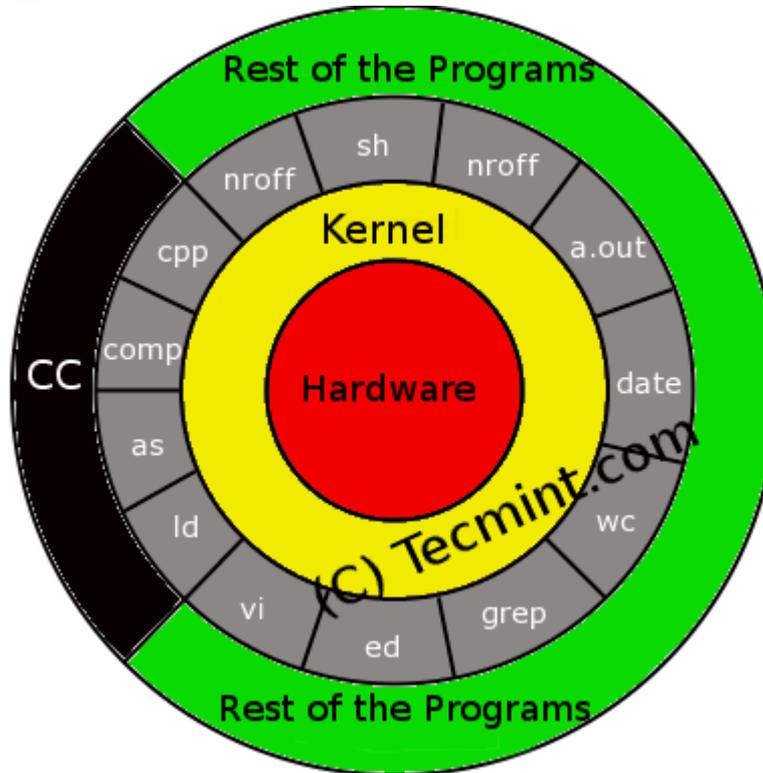


# Démarrage de Linux

<http://www.framboise314.fr/les-droits-linux-sur-le-raspberry-pi/>  
<https://openclassrooms.com/courses/reprenez-le-contrôle-a-l'aide-de-linux>  
[https://doc.ubuntu-fr.org/tutoriel/console\\_commandes\\_de\\_base](https://doc.ubuntu-fr.org/tutoriel/console_commandes_de_base)  
<https://technique.arscenic.org/commandes-linux-de-base/la-gestion-des-fichiers-et-dossiers/>

## Le système Linux

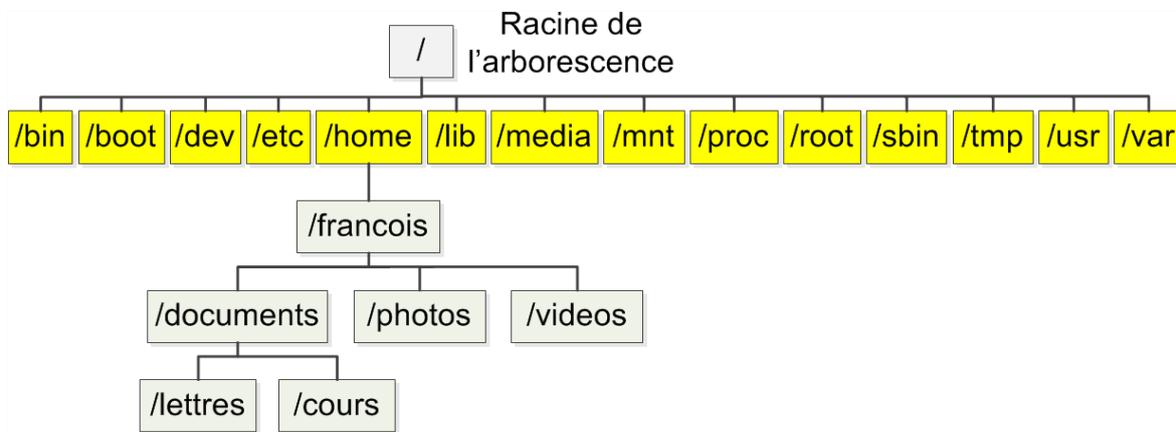
### Architecture du système



L'architecture de Linux est basée sur un noyau (Kernel) chargé de gérer les ressources, dont le matériel (Hardware) fait partie. De nombreux programmes réalisent l'interface entre l'utilisateur et le système. Ils offrent la possibilité de transmettre des ordres au noyau et de recevoir les réponses en mode texte. En Linux, le mode graphique est constitué de couches venant se superposer aux couches de base. Il reste donc toujours possible d'effectuer toutes les opérations en mode texte.

### L'arborescence de Linux

Le système de fichiers de Linux est organisé à partir d'un point de départ appelé root, racine ou encore /. Sous cette racine se déploient des répertoires contenant les fichiers et programmes nécessaires au système d'exploitation. La forme d'arbre inversé de cette structure fait qu'elle a été baptisée arborescence.



Chaque utilisateur peut créer dans son répertoire une arborescence à sa convenance. La connaissance complète de l'arborescence de Linux n'est pas nécessaire à l'utilisation de Linux. Le descriptif ci-dessous peut vous aider à trouver le fichier qui vous intéresse.

<b>Principaux répertoires de l'arborescence Linux</b>	
<b>/</b>	Racine ou root, contient les répertoires de l'arborescence Linux.
<b>bin</b>	Exécutables binaires du système <i>cp, ls, mount, rm...</i>
<b>boot</b>	Fichiers de démarrage de Linux.
<b>dev</b>	Fichiers spéciaux assurant la liaison avec les périphériques.
<b>etc</b>	Fichiers de configuration du système, des services...
<b>home</b>	Répertoire personnel des utilisateurs.
<b>lib</b>	Bibliothèques système partagées.
<b>media</b>	Point de montage des clés USB, CD-ROM...
<b>mnt</b>	Point de montage temporaire de partitions et périphériques.
<b>proc</b>	Informations sur les processus et le noyau Linux.
<b>root</b>	Répertoire personnel du super-utilisateur.
<b>sbin</b>	Binaires système et outils comme <i>fsck</i> .
<b>tmp</b>	Fichiers temporaires.
<b>usr</b>	Fichiers binaires et commandes utilisateurs.
<b>var</b>	Système de fichiers "variables" (modifiables) ; on y trouve le contenu web (répertoire <i>www</i> ), mais aussi les logs (journaux).

Ces répertoires contiennent eux-mêmes d'autres répertoires qui constituent l'arborescence de Linux. Quelques différences peuvent exister entre les distributions, mais la plus grande partie de l'arborescence est commune.

Lors d'un démarrage en mode texte, le login à utiliser est **pi** et le mot de passe **raspberry**.

# Si vous ne savez pas comment faire : R.T.F.M.

## Gestion de l'arborescence

### **pwd**

affiche le répertoire actuel

### **cd**

Changement de dossier actif ou dossier de travail

### **ls**

Lister le contenu d'un répertoire

### **mkdir**

Créer un répertoire

### **rm**

Suppression fichier ou répertoire

### **mv**

Déplacer ou renommer un fichier

### **cp**

Copier un fichier

### **chown**

Changer le propriétaire

### **chmod**

Changer les droits

## Les principales commandes

### **man**

Sans doute la commande la plus importante. C'est le manuel qui explique comment se servir de toutes les instructions.

### **sudo**

Permet à utilisateur de prendre momentanément le rôle de « super-utilisateur »

### **ls**

liste les fichiers et répertoires du répertoire en cours

### **cd**

Change de répertoire

### **mv**

Déplace un fichier ou change le nom d'un fichier

**cp**  
Copier un fichier ou un répertoire

**rm**  
Supprimer un fichier

**mkdir**  
Créer un répertoire

**rmdir**  
Supprimer un répertoire

**top**  
Donne des informations sur le système, ainsi que les processus et leur consommation de ressources. (quitter avec q)

**find**  
Recherche un fichier

**grep**  
Recherche une chaîne de caractères dans les fichiers, souvent utilisé en filtre

**cat**  
Affiche le contenu d'un fichier

**less**  
Gère l'affichage page par page

## Les droits sous Linux

### Gérer les droits

La commande `ls -l` affiche des informations sur les fichiers et répertoires, en particulier les droits attachés à chaque fichier, ainsi que le propriétaire du fichier et le groupe propriétaire.

```
root@raspberrypi:/home/user# ls -l
total 12
-rwxrw-rw- 1 user user 120 Jul 24 13:14 fichier1.txt
-rwxrw-rw- 1 user user 180 Jul 24 13:14 fichier2.txt
drw-r--r-- 2 user user 4096 Jul 24 13:15 lettres
root@raspberrypi:/home/user#
```

Le début de chacune des lignes donne des informations sur le fichier ou le répertoire dont le nom figure à l'extrémité droite de la ligne. Le premier bloc de 10 caractères concerne les droits attachés au fichier ou au répertoire listé.

Type	Propriétaire			Groupe propriétaire			Les autres		
d	r	w	x	r	w	x	r	w	x
	4	2	1	4	2	1	4	2	1

La première lettre indique le type du fichier :

**d** pour un répertoire (*directory* en anglais).

- pour un fichier ordinaire.

**c** pour un périphérique de type caractère (liaison série, modem...).

**b** pour un périphérique de type bloc (disque dur).

Les neuf caractères suivants sont répartis en trois blocs :

– Les droits du propriétaire du fichier.

– Les droits du groupe propriétaire du fichier.

– Les droits de tous les autres.

Chaque bloc comprend trois lettres qui, lorsqu'elles sont présentes, indiquent que le droit correspondant est disponible. Quand le droit n'existe pas, la lettre est remplacée par un tiret -.

**r** droit de lire.

**w** droit d'écrire.

**x** droit d'exécuter.

- le droit n'existe pas.

Les droits peuvent également être indiqués en utilisant les lettres ou les chiffres inscrits sous chacune des cases.

Suivent ensuite deux noms : **user user** qui indiquent pour le premier le nom du propriétaire du fichier, pour le second le nom du groupe propriétaire du fichier. Ici l'utilisateur *user* appartient au groupe *user*, d'où la répétition du nom. Ces informations proviennent de l'utilisateur qui a créé le fichier : son login et son groupe principal.

## Analyse des droits sur un fichier

```
-rwxrw-rw- 1 user user 120 Jul 24 13:14 fichier1.txt
```

Type	Propriétaire			Groupe propriétaire			Les autres		
-	r	w	x	r	w	-	r	w	-
	4	2	1	4	2	1	4	2	1
	7			6			6		

Le fichier appartient à l'utilisateur *user* et au groupe *user*.

Les droits correspondent à un fichier car la première lettre est un -.

Le propriétaire possède les droits *r*, *w* et *x* : il peut lire, écrire (modifier, supprimer) et exécuter ce fichier (ce qui a peu d'intérêt dans le cas présent d'un simple fichier texte). Les droits *rwx* correspondent à 4+2+1 et peuvent s'exprimer par le chiffre 7.

Les membres du groupe *user* autres que le propriétaire *user* ont le droit de lire et écrire ce fichier (*rw*). Les droits *rw* correspondent à 4+2 et peuvent s'exprimer par le chiffre 6.

Les utilisateurs qui ne sont ni *user* ni membres du groupe *user* (tous les autres) ont également le droit de lire et écrire ce fichier (*rw*). Les droits *rw* correspondent à 4+2 et peuvent s'exprimer par le chiffre 6.

Les droits de ce fichier peuvent s'écrire *rwxrw-rw-* ou *766*, qui est équivalent.

---

*Les droits sous Linux sont exprimés en octal, système de numération à base 8. Chaque groupe de 3 signes représente 1 bit et la valeur numérique est la traduction en octal du nombre binaire correspondant. *rwxrw-rw-* est équivalent à 111110110 en binaire et 766 en octal.*

## Droits sur un répertoire

Les droits sur un répertoire ont une signification différente des droits sur un fichier.

- r** afficher le contenu du répertoire avec **ls**.
- w** créer ou supprimer des fichiers dans le répertoire avec **mv**, **cp**, **rm**.
- x** accéder au répertoire avec la commande **cd**.
- le droit n'existe pas.

Si le droit *w* est positionné sur un répertoire, le droit *x* doit être positionné pour autoriser l'accès au répertoire, et *r* doit également être positionné pour pouvoir accéder aux fichiers contenus dans le répertoire. Lorsque *w* est positionné sur un répertoire, un utilisateur peut supprimer tous les fichiers qu'il contient, même s'il n'a pas de droits explicites sur ces derniers. Le Sticky Bit permet de limiter ce droit en limitant au seul propriétaire d'un fichier le droit de le supprimer.

## Droits étendus

SUID (*Set owner User ID up on execution* = exécuter avec l'identité du propriétaire) permet d'exécuter un programme avec les droits de son propriétaire au lieu des droits de celui qui lance l'exécution. Il n'a pas d'effet sur les répertoires. L'application de SUID à un fichier dont root est propriétaire pourrait permettre à un attaquant d'exécuter un programme avec les droits de root, ce qui constitue un risque pour la sécurité.

SGID (*Set Group ID up on execution* = exécuter avec l'identité du groupe) appliqué à un fichier a le même effet que SUID mais au niveau du groupe, avec le même risque pour la sécurité. Appliqué à un répertoire, SGID fait hériter son groupe propriétaire à tous les fichiers et répertoires créés dans ce répertoire. Dans un partage, ceci permet à tous les membres d'un groupe de modifier des fichiers communs.

Sticky Bit positionné sur un répertoire, empêche un utilisateur qui ne serait pas propriétaire d'un fichier de le supprimer.

## Changement du propriétaire et du groupe propriétaire

Le propriétaire et le groupe propriétaire d'un fichier ou d'un répertoire sont positionnés à la création, avec le login et le groupe principal de celui qui les crée.

Le changement de propriétaire se fait avec la commande `chown`, celui du groupe propriétaire avec `chgrp`. Seul le super-utilisateur peut changer le propriétaire et le groupe d'un fichier ou d'un répertoire dont il n'est pas propriétaire.

Lorsque le propriétaire d'un fichier est changé, l'ancien propriétaire n'a plus de droits sur ce fichier. S'il appartient au même groupe que le nouveau propriétaire, il aura les droits du groupe. S'il n'appartient pas au même groupe que le nouveau propriétaire, il aura les droits de tous les autres.

### Syntaxe

```
chown nouveau_propriétaire nom_du_fichier
chgrp nouveau_propriétaire nom_du_fichier
```

### Exemple de changement de l'utilisateur propriétaire d'un fichier

L'utilisateur *user* essaye de rendre *mozart* propriétaire de son fichier. Le système indique que cette opération n'est pas permise.

```
user@raspberrypi ~ $ chown mozart fichier1.txt
chown: changing ownership of `fichier1.txt': Operation not
permitted
```

### Seul le super-utilisateur peut changer le propriétaire d'un fichier

```
root@raspberrypi:/home/user# chown mozart fichier1.txt
root@raspberrypi:/home/user# ls -l
total 12
-rw-r--r-- 1 mozart user 120 Jul 24 13:14 fichier1.txt
-rw-r--r-- 1 user user 180 Jul 24 13:14 fichier2.txt
drw-r--r-- 2 user user 4096 Jul 24 13:15 lettres
```

### Exemple de changement du groupe propriétaire d'un fichier

Le super-utilisateur change le groupe propriétaire de *fichier1.txt*.

```
root@raspberrypi:/home/user# chgrp mozart fichier1.txt
root@raspberrypi:/home/user# ls -l
total 12
-rw-r--r-- 1 mozart mozart 120 Jul 24 13:14 fichier1.txt
-rw-r--r-- 1 user user 180 Jul 24 13:14 fichier2.txt
drw-r--r-- 2 user user 4096 Jul 24 13:15 lettres
```

## Modification des droits

La modification des droits des fichiers et répertoires se fait avec la commande `chmod`. Le propriétaire d'un fichier et le super-utilisateur peuvent tous deux exécuter la commande `chmod`.

`chmod` fonctionne de façon littérale (mode symbolique) en faisant des additions et des soustractions de droits, ou de façon numérique en appliquant directement l'ensemble des droits.

### Syntaxe

```
chmod [u g o a] [+ - =] [r w x] nom_du_fichier
chmod NNN nom_du_fichier
```

u	Appliquer les modifications à l'utilisateur propriétaire ( <i>user</i> ).
g	Appliquer les modifications au groupe propriétaire ( <i>group</i> ).
o	Appliquer les modifications aux autres ( <i>others</i> ).
a	Appliquer les modifications aux trois blocs ( <i>all</i> ).
+	Ajouter un droit.
-	Retirer un droit.
=	Écraser un droit.
r	Droit de lire.
w	Droit d'écrire.
x	Droit d'exécuter.

### Exemple d'utilisation de chmod en mode symbolique

Un fichier nommé *fichier1.txt* possède les droits *rw-rw-rw-*.

```
root@raspberrypi:/home/user# ls -l
total 4
-rwxrwxrwx 1 user user 180 Jul 24 13:14 fichier1.txt
```

Pour tous les utilisateurs (*a*), le super-utilisateur écrase les droits (=) et les remplace par lecture (*r*).

```
root@raspberrypi:/home/user# chmod a=r fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-r--r--r-- 1 user user 180 Jul 24 13:14 fichier1.txt
```

Pour le groupe seulement (*g*), le super-utilisateur ajoute (+) le droit d'exécuter (*x*).

```
root@raspberrypi:/home/user# chmod g+x fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-r--r-xr-- 1 user user 180 Jul 24 13:14 fichier1.txt
```

Pour le propriétaire seulement (*u*), le super-utilisateur ajoute (+) les droits d'écrire et d'exécuter (*wx*).

```
root@raspberrypi:/home/user# chmod u+wx fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-rwxr-xr-- 1 user user 180 Jul 24 13:14 fichier1.txt
```

Pour le propriétaire et le groupe (*ug*), le super-utilisateur retire (-) le droit d'exécuter (*x*).

```
root@raspberrypi:/home/user# chmod ug-x fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-rw-r--r-- 1 user user 180 Jul 24 13:14 fichier1.txt
root@raspberrypi:/home/user#
```

## Exemple d'utilisation de `chmod` en mode numérique

Avec un peu d'habitude cette méthode est très rapide. Elle permet de configurer en une seule fois tous les droits sur un fichier ou un répertoire. Combinée avec le rappel des commandes précédentes, elle permet un gain de temps appréciable.

Le fichier *fichier1.txt* a les droits *rw-r--r--*.

```
root@raspberrypi:/home/user# ls -l
total 4
-rw-r--r-- 1 user user 180 Jul 24 13:14 fichier1.txt
```

Le super-utilisateur donne tous les droits à tout le monde.

```
root@raspberrypi:/home/user# chmod 777 fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-rwxrwxrwx 1 user user 180 Jul 24 13:14 fichier1.txt
```

Le super-utilisateur donne tous les droits mais seulement au propriétaire, le groupe et les autres n'ont aucun droit.

```
root@raspberrypi:/home/user# chmod 700 fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-rwx----- 1 user user 180 Jul 24 13:14 fichier1.txt
```

Le super-utilisateur laisse tous les droits au propriétaire, et le droit de lire et d'écrire au groupe et aux autres.

```
root@raspberrypi:/home/user# chmod 766 fichier1.txt
root@raspberrypi:/home/user# ls -l
total 4
-rwxrw-rw- 1 user user 180 Jul 24 13:14 fichier1.txt
```

---

Lors de la création d'un fichier, un masque est appliqué aux droits par défaut de Raspbian. La commande *umask* retourne la valeur *0022* qui correspond à *---w--w-*. Ce masque est soustrait des droits fondamentaux de la distribution. Les fichiers créés ne sont modifiables ni par le groupe propriétaire ni par les autres.

## Proposition de manipulations

- Se positionner dans le répertoire de l'utilisateur pi (par défaut) **/home/pi**
- Voir la liste des fichiers « visibles »
- Voir la liste de tous les fichiers, y compris les fichiers cachés
- Créer un fichier **fichier1** dans le dossier **/home/pi** et y mettre du texte, avec **nano**
- Créer un dossier **dossier1** dans le dossier **/home/pi**, et le faire devenir répertoire de travail
- Copier le fichier **fichier1** dans le répertoire **dossier1**
- Créer un fichier **fichier2** dans le dossier **/home/pi** et y mettre du texte, avec **nano**
- Déplacer le fichier **fichier2** dans le répertoire **dossier1**
- Supprimer le fichier **fichier2**
- Dans le répertoire **/home/pi** créer un fichier **test.sh**
- Dans ce fichier, écrivez avec **nano** :  

```
#!/bin/bash
ls - al
```
- Exécuter le fichier **test.sh** : **./test.sh** (pourquoi mettre **./** devant le nom de fichier ? Que se passe-t-il si on ne le met pas ?)
- Rendre le fichier **test.sh** exécutable : **chmod 755 test.sh** (que signifie le 755 ? Vérifiez que la modification a bien été réalisée avec la commande **ls**)
- Exécuter à nouveau le fichier **test.sh** : **./test.sh** (Qu'affiche-t-il ?)
- Modifiez **test.sh** de la façon suivante :  

```
#!/bin/bash
ls - al > fichier3
```
- Exécuter à nouveau le fichier **test.sh** : **./test.sh** (Qu'affiche-t-il ?)
- Regarder la liste des fichiers. Que contient le fichier **fichier3** ? Quelle est la signification du signe **>** ?
- Positionnez-vous dans le répertoire **/etc**
- Lister les fichiers que contient ce répertoire **/etc**, y compris les fichiers cachés
- La liste défile rapidement et n'est pas lisible dans sa totalité. Utilisez **|less** (le signe **|** s'obtient avec **Alt Gr 6**). Que se passe-t-il ? Avancez avec les touches **ESPACE** ou **ENTREE** : Quelle est la différence d'action entre ces 2 touches ? Sortez avec la touche **q** quand le système affiche (**END**).
- Supprimer le dossier **dossier1** dans **/home/pi**.
- Dans **/home/pi**, créer un fichier **demo**. Le rendre exécutable par son propriétaire, lisible par tout le monde, modifiable par son propriétaire et par le groupe propriétaire. Quelle valeur prennent les droits ?
- Changer le propriétaire du fichier **demo**, remplacer **pi** par **toto**. Que se passe-t-il ?
- Créer l'utilisateur **user** avec la commande **adduser user**. Que se passe-t-il ?
- Créer l'utilisateur **user** avec **sudo adduser user**. Pourquoi cette commande fonctionne-t-elle ? Répondez aux questions que pose le système. Je vous conseille le mot de passe **user** qui sera facile à retenir et à associer au nom d'utilisateur (notre Raspberry Pi n'étant pas utilisé en production connecté à Internet, ce n'est pas problématique).
- Changer le propriétaire du fichier **demo**, remplacer **pi** par **user**. Vérifiez la modification avec la commande **ls**
- Connectez-vous comme l'utilisateur **user** : **su user =>** Que devient le prompt ?
- Saisissez le mot de passe de **user** et vérifiez les groupes auxquels **user** appartient avec la commande : **groups** A quels groupes appartient l'utilisateur **user** ?
- Redevenez l'utilisateur **pi** avec la commande : **exit =>** regardez à nouveau le prompt
- Changer le groupe propriétaire du fichier **demo**, remplacer **pi** par **user**. Vérifiez la modification avec la commande **ls**

Maintenant... à vous de jouer !