

Guide utilisation du système linux

Contenu

Introduction.....	3
A. Comment utiliser le système de fichiers Linux	3
Visualiser les répertoires et les fichiers	3
<i>Chemins absous et relatifs.....</i>	5
Afficher les listes de fichiers	6
<i>Répertoires communs pour le système Linux</i>	9
B. Créer et supprimer des fichiers.....	11
Supprimer des fichiers.....	13
C. Créer et supprimer des répertoires	15
Gérer des répertoires	15
Supprimer les répertoires.....	16
D. Déplacer et renommer des fichiers et des répertoires.....	18
Déplacer des fichiers	18
Renommer fichiers et répertoires	20
E. Aide intégrée d'utilisation des commandes Linux	21
Présentation générale	21
Les options de « man ».....	24
Modifier le contenu de fichiers texte	25
F. Commandes de gestion des utilisateurs.....	30
Généralités	30
La commande « sudo ».....	31
G. Gestion des propriétés et autorisations.....	34
Généralités	34
Modifier les autorisations	35
H. Commandes Linux pour le système et le disque	36
Commandes pour le système	36
Commandes pour le disque.....	40
Conclusion	43

Liste des figures

Figure 1 : Commande « pwd ».....	4
Figure 2 : Commande « pwd », « ls » et « cd ».....	5
Figure 3 : Chemins absous et relatifs.....	6
Figure 4 : Commande « ls -l »	7
Figure 5 : Commande « ls -lh »	8
Figure 6 : Commande « ls -lah »	9
Figure 7 : Commande « ls /bin ».....	10
Figure 8 : Commande « touch » sur un répertoire existant	11
Figure 9 : Commande « touch » sur un fichier (ou répertoire) non existant	12
Figure 10 : Commande « rm »	13
Figure 11 : Commande « rm » avec « * »	14
Figure 12 : Commande « mkdir »	15
Figure 13 : Commande « mkdir » avec l'option « -p »	16
Figure 14 : Commande « rmdir » avec les options « -Ri ».....	17
Figure 15 : Commande « mv ».....	18
Figure 16 : Commandes « mv », « cd .. » et « cd ~ »	19
Figure 17 : Commande « mv » utilisée pour renommer des fichiers et des répertoires	20
Figure 18 : Accès au manuel d'aide à l'aide de la commande « man ».....	21
Figure 19 : Accès au manuel d'aide à l'aide de la commande « man » (<i>suite</i>).....	22
Figure 20 : Touches contrôle en mode page par page	23
Figure 21 : Commande « whatis »	23
Figure 22 : Options de la commande « man »	24
Figure 23 : Commande « man -k »	25
Figure 24 : Afficher le contenu d'un fichier texte.....	26
Figure 25 : Exemple de contenu d'un fichier texte	26
Figure 26 : Commande « cat ».....	27
Figure 27 : Interface de l'éditeur de texte « nano » – début de modification.....	27
Figure 28 : Interface de l'éditeur de texte « nano » – fin de modification	28
Figure 29 : Interface de l'éditeur de texte « nano » – sauvegarde du fichier	28
Figure 30 : Interface de l'éditeur de texte « nano » – modification enregistrée	29
Figure 31 : Commande « ls -l » pour fichier modifié	29
Figure 32 : Commande « whoami ».....	31
Figure 33 : Commande « sudo ».....	32
Figure 34 : Modification des autorisations à l'aide de la commande « sudo visudo »	33
Figure 35 : Propriétés et autorisations des fichiers et des répertoires	34

Figure 36 : Modifications des autorisations d'un fichier.....	36
Figure 37 : Commande « cat ».....	37
Figure 38 : Commande « cat /proc/meminfo ».....	38
Figure 39 : Commande « cat /proc/partitions ».....	39
Figure 40 : Commande « cat /proc/version ».....	39
Figure 41 : Diverses commandes système	40
Figure 42 : Diverses commandes pour le disque.....	41
Figure 43 : Commande « du -ha »	42

INTRODUCTION

Ce guide est conçu afin de vous permettre une prise en main rapide du système d'exploitation Linux avec le nano ordinateur Raspberry Pi. Ce guide est aussi un aide-mémoire pour ceux et celles qui ont déjà exploité le système d'exploitation Unix. Linux tirant sa source de Unix, le lecteur ou la lectrice de ce document peut, sans aucune restriction, substituer ces deux termes l'un pour l'autre.

A. Comment utiliser le système de fichiers Linux

Visualiser les répertoires et les fichiers

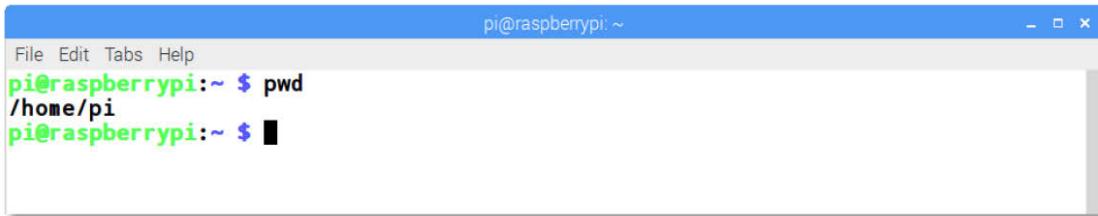
Étape 1

Dans Linux, le chemin vers un répertoire ou un fichier est indiqué par une chaîne de noms de répertoires imbriqués, celui de gauche contenant celui à sa droite et ainsi de suite jusqu'au fichier ou au répertoire concerné. Les noms des répertoires sont séparés par le signe « / ».

Supposons que le système est installé et qu'il vient de démarrer. Après le défilement d'une série de commandes activées par le système lui-même, suivies par des « [ok] » qui indiquent l'exécution correcte de ces commandes, l'« invite » des lignes commandes « pirasberry \$ » apparaît. Avant de commencer, la commande « clear » permettra d'effacer le contenu de l'écran.

Étape 2

La commande « pwd » (*Print Working Directory*) affiche le répertoire sur lequel on travaille actuellement (plus probablement : « /home/pi »).



```
File Edit Tabs Help  
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $ █
```

Figure 1 : Commande « pwd »

Étape 3

La commande « ls » permet de visualiser le contenu du répertoire courant. Les répertoires et les fichiers sont affichés avec des codes couleur différents selon leur type (voir Figure 2 : Commande « pwd », « ls » et « cd »)

) :

- pour les répertoires : bleu (par exemple : *Desktop*, *Documents*, *Downloads* et *Scratch*);
- pour les fichiers ordinaires : blanc;
- pour les fichiers exécutables : vert scintillant;
- pour les fichiers archives : rouge;

et ainsi de suite.

Étape 4

Pour se déplacer entre les différents répertoires, par exemple du répertoire « pi » au répertoire « Documents », la commande « cd Documents » est utilisée (Documents avec un D majuscule, car Linux est sensible à la casse [« case sensitive »]).

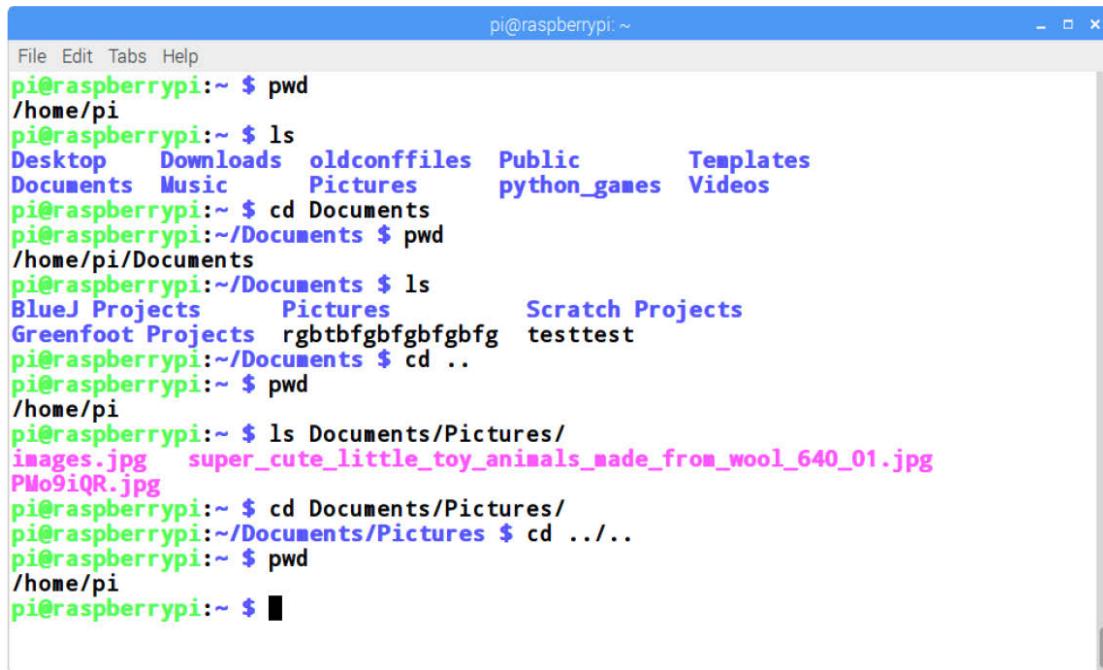
En utilisant la commande « pwd » à nouveau, le résultat sera : « /home/pi/Documents ». La commande « ls » permettra de visualiser le contenu du répertoire « Documents ».

Étape 5

Pour revenir dans le répertoire « pi », on peut utiliser, entre autres, la commande « cd .. » car les deux points dans Linux représentent le répertoire du niveau supérieur, généralement référencé par le « répertoire parent ». Quant à lui, le point seul (« . ») est utilisé pour représenter le répertoire courant.

Étape 6

Les commandes « ls » et « cd » permettent également d'afficher le contenu de chemins plus complexes. Par exemple, « ls Documents/Pictures » affichera le contenu du répertoire « Pictures » contenu dans le répertoire « Documents ».



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls
Desktop Downloads oldconf files Public Templates
Documents Music Pictures python_games Videos
pi@raspberrypi:~ $ cd Documents
pi@raspberrypi:~/Documents $ pwd
/home/pi/Documents
pi@raspberrypi:~/Documents $ ls
BlueJ Projects Pictures Scratch Projects
Greenfoot Projects rgtbtfgbfgbfgbfg testtest
pi@raspberrypi:~/Documents $ cd ..
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls Documents/Pictures/
images.jpg super_cute_little_toy_animals_made_from_wool_640_01.jpg
PMo9iQR.jpg
pi@raspberrypi:~ $ cd Documents/Pictures/
pi@raspberrypi:~/Documents/Pictures $ cd ../..
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $
```

Figure 2 : Commande « `pwd` », « `ls` » et « `cd` »

Chemins absolus et relatifs

Pour résumer, « `ls` » sans aucune autre spécification affiche le répertoire de travail ou le répertoire courant. Pour savoir quel est ce répertoire, on utilise la commande « `pwd` ».

Afin de distinguer la différence entre un chemin absolu et un chemin relatif, il est important de se rappeler que le répertoire racine « `/` » est celui auquel tous les autres répertoires sont subordonnés; donc « `ls /` » affichera le contenu de « racine » et « `cd /` » changera notre répertoire courant pour le répertoire racine, c'est-à-dire « `/` ».

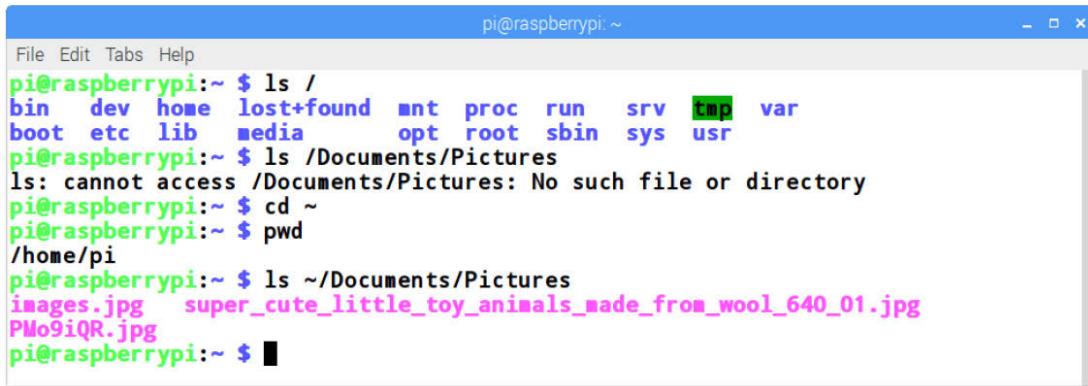
Ainsi, « `ls Documents/Pictures` » et « `ls /Documents/Pictures` » sont des commandes qui ne donneront pas le même résultat : la première commande affiche le contenu du répertoire « `Pictures` », qui se trouve dans le répertoire « `Documents` », qui à son tour, se trouve dans le répertoire courant (de travail), tandis que la seconde commande **devrait** afficher le contenu du répertoire « `Pictures` », qui **devrait** se trouver dans un répertoire « `Documents` », qui a son tour devrait supposément se trouver dans le répertoire racine (selon le signe « `/` » du début de l'argument de la commande « `ls` » (« `/Documents/Pictures` »); mais, comme il n'y a habituellement pas de répertoire « `Documents` » dans le répertoire racine, Linux affichera l'erreur « `No such file or directory` » (« Pas de fichier ou répertoire à ce nom »).

Il faut donc distinguer ici le **chemin absolu** qui débute avec le répertoire racine : « `/` », dans notre exemple : « `ls /Documents/Pictures` », du **chemin relatif** qui, lui, ne débute pas avec le répertoire racine, mais avec le répertoire courant : « `ls Documents/Pictures` ».

Il existe un raccourci pour aboutir dans votre répertoire personnel (« `/home/pi` » dans le cas présent) : en utilisant le caractère « tilde » (« `~` »); avec « `ls ~` » on affiche toujours le

contenu de votre répertoire personnel et avec « cd ~ » on aboutit dans votre répertoire personnel à partir de n'importe quel autre répertoire de travail.

Le caractère « ~ » peut être utilisé dans une adresse plus complexe de répertoire : par exemple, « ls ~/Documents/Pictures » affichera le contenu de « Pictures » de votre répertoire personnel, même si votre répertoire de travail (courant) est différent du répertoire personnel.



```
pi@raspberrypi:~ $ ls /
bin dev home lost+found mnt proc run srv tap var
boot etc lib media opt root sbin sys usr
pi@raspberrypi:~ $ ls /Documents/Pictures
ls: cannot access /Documents/Pictures: No such file or directory
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls ~/Documents/Pictures
images.jpg super_cute_little_toy_animals_made_from_wool_640_01.jpg
PMo9iQR.jpg
pi@raspberrypi:~ $
```

Figure 3 : Chemins absous et relatifs

Afficher les listes de fichiers

La commande « ls » permet d'afficher la liste des fichiers. Celle-ci s'utilise avec différentes options.

Option 1 : -l

La première option « ls -l », affiche les répertoires et les fichiers sous le format long; chaque fichier ou répertoire est caractérisé par une seule ligne :

- au début de la ligne, il y a des caractères (lettres ou traits) : qui désignent les autorisations (abordées à la section 0);
- après les premiers 10 caractères, on trouve un numéro qui représente le nombre de fichiers dans le répertoire :
 - si la ligne se réfère à un fichier : ce nombre sera 1,
 - si c'est un répertoire : le nombre sera au moins 2 (c'est parce que chaque répertoire contient au moins deux fichiers cachés (un « . » et un « .. »)),
 - s'il y a d'autres fichiers (outre ces deux fichiers cachés) dans le répertoire, ce nombre représentera le nombre de fichiers contenus dans le répertoire.

```
pi@raspberrypi:~ $ ls -l /
total 84
drwxr-xr-x  2 root root  4096 Nov 25 17:33 bin
drwxr-xr-x  4 root root 16384 Dec 21 18:09 boot
drwxr-xr-x 15 root root  3400 Nov 25 23:49 dev
drwxr-xr-x 116 root root 4096 Dec 21 18:16 etc
drwxr-xr-x  3 root root  4096 Nov 25 17:24 home
drwxr-xr-x 18 root root 4096 Nov 25 17:33 lib
drwx-----  2 root root 16384 Nov 25 17:59 lost+found
drwxr-xr-x  2 root root 4096 Nov 25 17:19 media
drwxr-xr-x  2 root root 4096 Nov 25 17:19 mnt
drwxr-xr-x  7 root root 4096 Nov 25 17:49 opt
dr-xr-xr-x 161 root root    0 Jan  1 1970 proc
drwx-----  3 root root 4096 Nov 25 17:54 root
drwxr-xr-x 23 root root   760 Dec 21 18:08 run
drwxr-xr-x  2 root root 4096 Nov 25 17:33 sbin
drwxr-xr-x  2 root root 4096 Nov 25 17:19 srv
dr-xr-xr-x 12 root root    0 Dec 21 19:56 sys
drwxrwxrwt 14 root root 4096 Dec 21 19:59 tmp
drwxr-xr-x 11 root root 4096 Nov 25 17:46 usr
drwxr-xr-x 11 root root 4096 Nov 25 18:09 var
pi@raspberrypi:~ $
```

Figure 4 : Commande « ls -l »

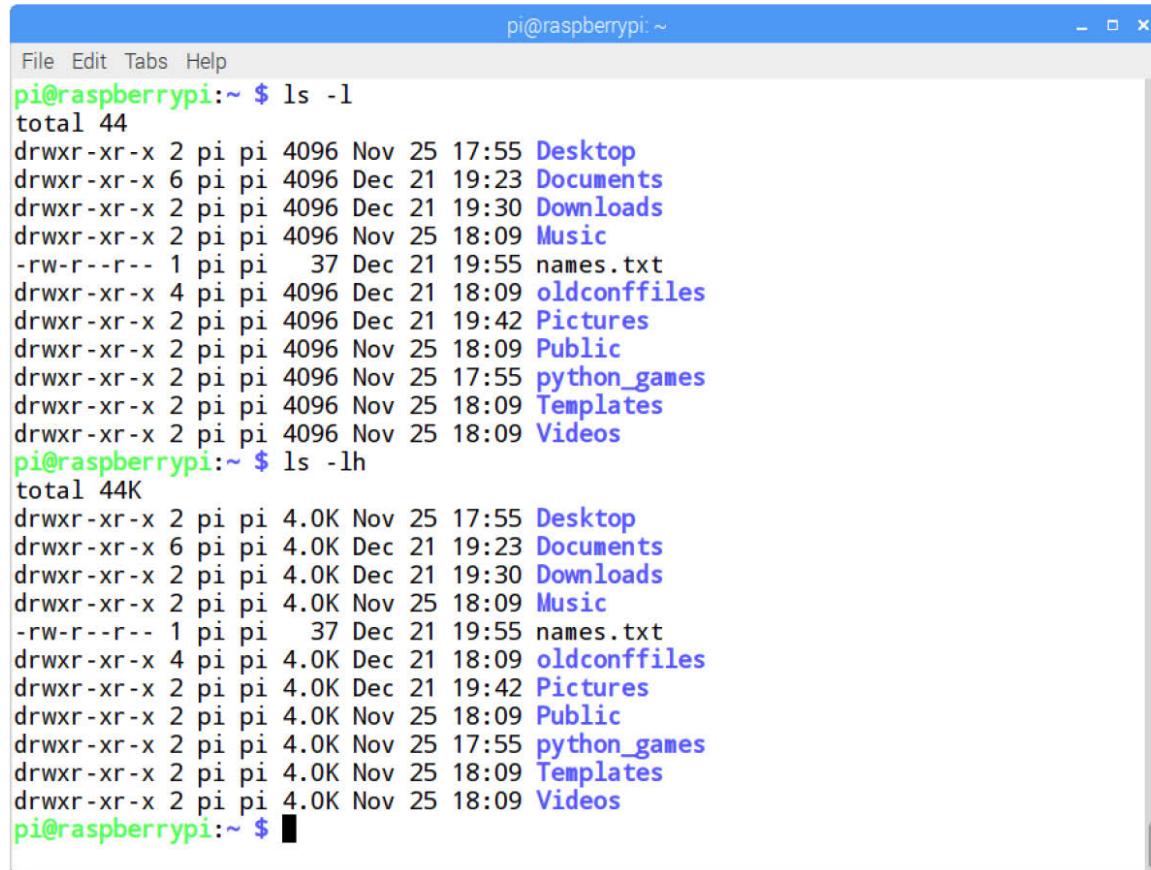
Dans la figure suivante (Figure 5), après le nombre de fichiers, on peut voir le nom « pi » deux fois :

- le premier « pi » se réfère au nom de l'utilisateur qui est propriétaire du fichier (qui est implicitement « pi » dans notre cas);
- le deuxième « pi » se réfère au nom du groupe auquel appartient l'utilisateur et qui est habituellement le même que le nom de l'utilisateur.

On peut introduire « ls -l / » pour afficher les répertoires et les fichiers du **répertoire racine**, et qui **appartiennent à l'utilisateur racine**.

Option 2 : -h

Le nombre qui suit représente le nombre d'octets contenus par le fichier; comme les fichiers peuvent être très volumineux, on utilise la commande « ls » avec l'option « h » : « ls -lh » pour « humaniser » le nombre, c'est-à-dire, le rendre lisible.



```
pi@raspberrypi:~ $ ls -l
total 44
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 19:42 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ ls -lh
total 44K
drwxr-xr-x 2 pi pi 4.0K Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4.0K Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4.0K Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4.0K Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4.0K Dec 21 19:42 Pictures
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4.0K Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:09 Templates
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:09 Videos
pi@raspberrypi:~ $
```

Figure 5 : Commande « ls -lh »

Option 3 : -a

Enfin, il faut se rendre compte que certains fichiers sont cachés, donc on doit utiliser la dernière option « -a ». À noter que les fichiers et répertoires cachés commencent avec un « . ».

En utilisant cette option combinée aux précédentes, on obtient l'image complète d'un répertoire.

The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The main area displays the output of the 'ls -lah' command, listing files and directories in long format. The output includes columns for permissions, number of links, owner, group, size, date modified, and file name. Many file names are colored in blue, indicating they are symbolic links or special files.

```
pi@raspberrypi:~ $ ls -lah
total 132K
drwxr-xr-x 24 pi pi 4.0K Dec 21 19:52 .
drwxr-xr-x 3 root root 4.0K Nov 25 17:24 ..
-rw-r--r-- 1 pi pi 69 Nov 25 18:24 .asoundrc
-rw----- 1 pi pi 178 Nov 25 23:57 .bash_history
-rw-r--r-- 1 pi pi 220 Nov 25 17:24 .bash_logout
-rw-r--r-- 1 pi pi 3.5K Nov 25 17:24 .bashrc
drwxr-xr-x 7 pi pi 4.0K Dec 21 18:17 .cache
drwx----- 16 pi pi 4.0K Dec 21 18:09 .config
drwx----- 3 pi pi 4.0K Nov 25 18:09 .dbus
drwxr-xr-x 2 pi pi 4.0K Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4.0K Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4.0K Dec 21 19:30 Downloads
drwx----- 2 pi pi 4.0K Dec 21 18:15 .gconf
drwx----- 2 pi pi 4.0K Dec 21 18:16 .gnome2
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:10 .gstreamer-0.10
drwxr-xr-x 3 pi pi 4.0K Nov 25 17:55 .local
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4.0K Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4.0K Dec 21 20:00 Pictures
drwx----- 3 pi pi 4.0K Nov 25 19:59 .pki
-rw-r--r-- 1 pi pi 675 Nov 25 17:24 .profile
drwxr-xr-x 2 pi pi 4.0K Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4.0K Nov 25 17:55 python_games
drwxr-xr-x 3 pi pi 4.0K Dec 21 18:45 .shutter
```

Figure 6 : Commande « ls -lah »

Répertoires communs pour le système Linux

bin : contient des fichiers application;

sbin : contient des sous-programmes du système d'opération;

dev : se réfère aux périphériques du système;

etc : contient les fichiers de configuration du système;

lib : contient des bibliothèques de code auxquelles se réfèrent d'autres programmes;

var : divers programmes;

tmp : programmes placés temporairement et qui peuvent être effacés après une courte période de temps.

The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal displays the output of the command 'ls /bin'. The output lists numerous binary executables, many of which are highlighted in red, such as 'fusermount', 'ping6', and 'umount'. Other executables like 'sync', 'systemctl', and 'systemd' are also highlighted.

```
pi@raspberrypi:~ $ ls /bin
bash          fgconsole    mountpoint   sync
bunzip2       fgrep        mt           systemctl
bzcat         findmnt     mt-gnu       systemd
bzcmp         fuser        mv           systemd-ask-password
bzdiff        fusermount   nano         systemd-escape
bzegrep       getfacl      nc           systemd-inhibit
bzexe         grep         nc.openbsd  systemd-machine-id-setup
bzfgrep       gunzip      nc.traditional  systemd-notify
bzgrep        gzexe       netcat      systemd-tmpfiles
bzip2         gzip         netstat     systemd-tty-ask-password-agent
bzip2recover  hciconfig   nisdomainname tailf
bzless        hostname   open         tar
bzmore        ip          openvt      tempfile
cat           journalctl pidof       touch
chacl         kbd_mode    ping        true
chgrp         kill        ping6       udevadm
chmod         kmod        Plymouth    unlockmgr_server
chown         less        ps          umount
chvt          lessecho    pwd         uname
con2fbmap    lessfile    rbash       uncompress
cp            lesskey     readlink   unicode_start
cpio          lesspipe    red         vdir
dash          ln          rm          wdctl
date          loadkeys   rmdir      which
dd            login      rnano      ypdomainname
```

Figure 7 : Commande « ls /bin »

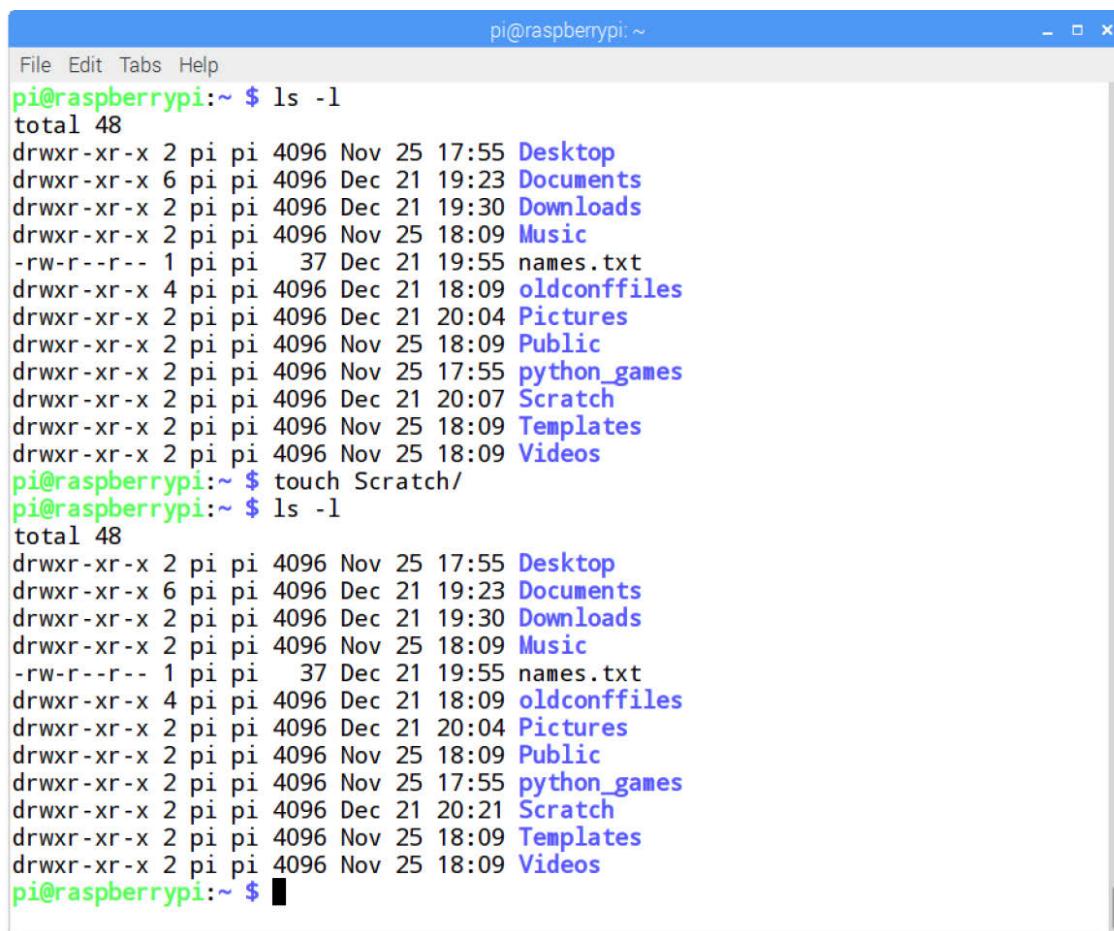
B. Créer et supprimer des fichiers

Cette section concerne la création et la suppression de fichiers. Par contre, une action sur un fichier peut se traduire par une action sur un répertoire si celui-ci existe. Vous référez à la section C pour la création et la suppression de répertoires.

Étape 1

Pour créer un fichier, on utilise la commande « touch ». Cette commande a plusieurs effets : si elle est appliquée à un fichier ou à un répertoire existant, elle change la date et l'heure relatives au fichier ou au répertoire avec la date et l'heure courantes, de la même manière que si le fichier (ou un répertoire) avait été créé.

Prenons par exemple, un répertoire « Scratch » existant. En utilisant la commande « ls -l », on constate l'heure et la date de la création du répertoire Scratch (21 décembre à 20:07). Ensuite, on saisit la commande « touch Scratch », qui normalement, devrait créer un nouveau fichier vide, mais comme il s'agit d'un répertoire existant, la date et l'heure seront changées. La commande « ls -l » démontre que l'heure fut changée pour 20:21 à la même date.



```
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:04 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:07 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ touch Scratch/
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:04 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $
```

Figure 8 : Commande « touch » sur un répertoire existant

Étape 2

La commande « touch testfile » crée un nouveau fichier vide.

Attention!

- Comme Linux est sensible à la casse, « testfile » et « Testfile » sont des fichiers différents.
- Le nom des fichiers ne peut pas contenir d'espaces. Ainsi, « touch test file » créera deux fichiers différents : « test » et « file ». Utiliser alors un « _ » dans le nom.

The screenshot shows a terminal window titled "pi@raspberrypi: ~". The terminal displays the following session:

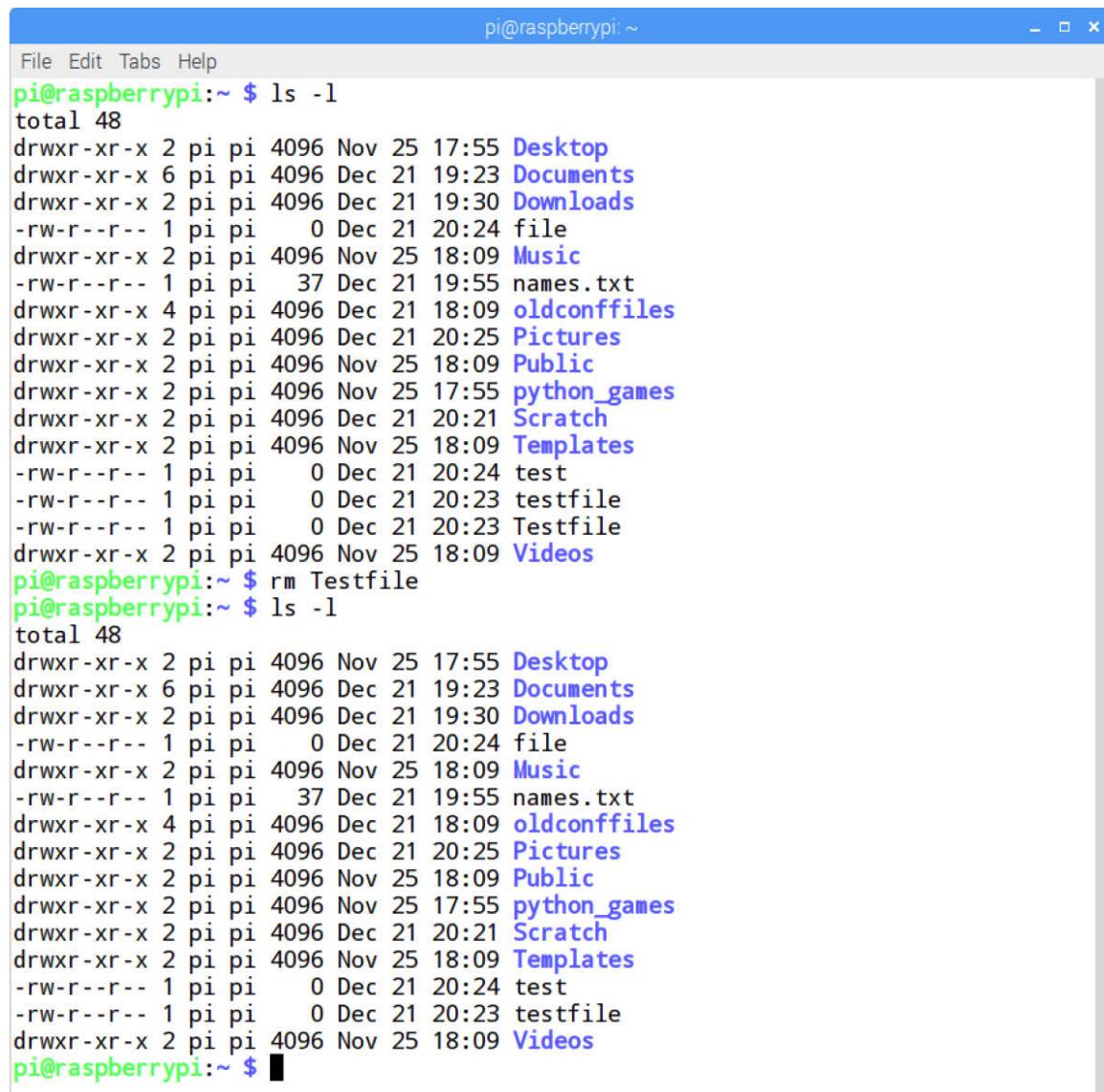
```
pi@raspberrypi:~ $ touch testfile
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:22 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
-rw-r--r-- 1 pi pi 0 Dec 21 20:23 testfile
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ touch Testfile
pi@raspberrypi:~ $ ls
Desktop  Downloads  names.txt  Pictures  python_games  Templates  Testfile
Documents  Music  oldconffiles  Public  Scratch  testfile  Videos
pi@raspberrypi:~ $ touch test file
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 file
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:22 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 test
-rw-r--r-- 1 pi pi 0 Dec 21 20:23 testfile
-rw-r--r-- 1 pi pi 0 Dec 21 20:23 Testfile
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $
```

Figure 9 : Commande « touch » sur un fichier (ou répertoire) non existant

Supprimer des fichiers

Étape 1

On suppose qu'on a créé les fichiers « test », « testfile » et « Testfile ». Avec la commande « rm Testfile », on supprimera le fichier « Testfile ». On vérifie ensuite qu'il n'existe plus dans la liste en utilisant la commande « ls -l ».



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The user runs three commands: "ls -l" to show all files, "rm Testfile" to delete the file, and another "ls -l" to verify its removal. The terminal output is as follows:

```
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 file
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:25 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 test
-rw-r--r-- 1 pi pi 0 Dec 21 20:23 testfile
-rw-r--r-- 1 pi pi 0 Dec 21 20:23 Testfile
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ rm Testfile
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 file
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:25 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 test
-rw-r--r-- 1 pi pi 0 Dec 21 20:23 testfile
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ █
```

Figure 10 : Commande « rm »

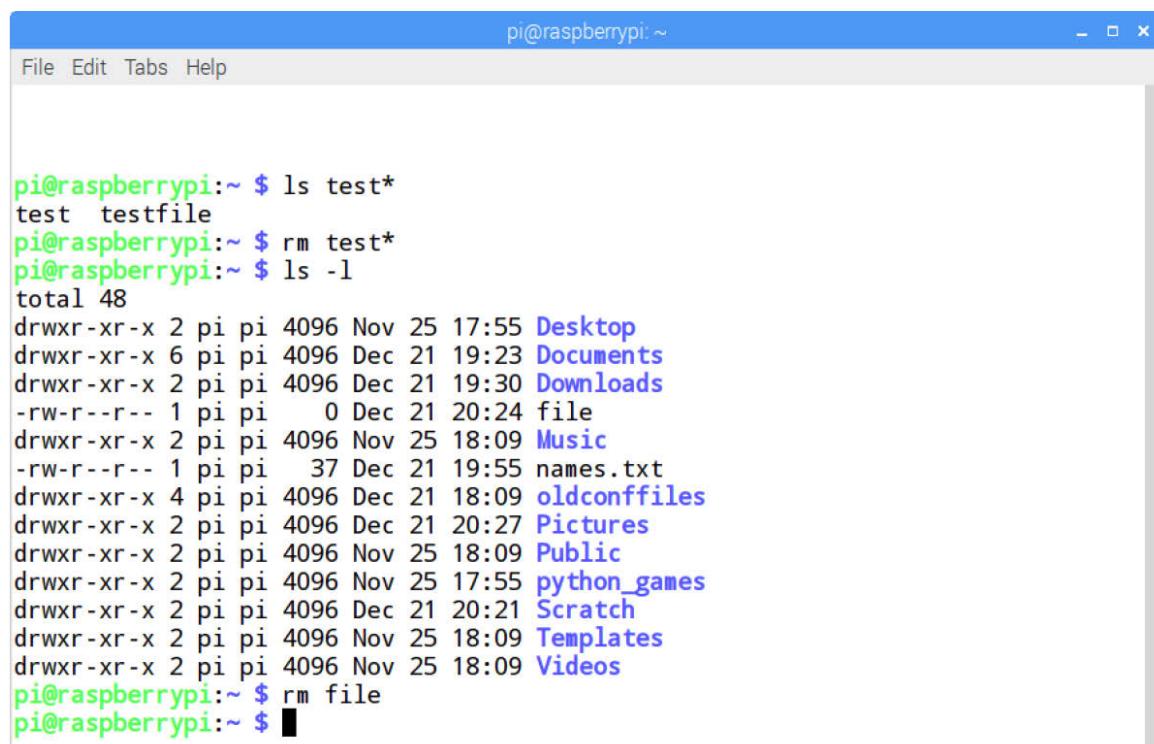
Attention!

Un fichier effacé ne peut plus être récupéré, comme c'est le cas pour Windows!

Étape 2

Si on veut supprimer plusieurs fichiers avec des noms similaires, on peut utiliser le caractère générique « * ». Il faut être prudent et éviter de supprimer des fichiers par mégarde.

Dans la figure qui suit (Figure 11), avant de supprimer plusieurs fichiers, on utilisera d'abord la commande « ls test* », où « * » représente n'importe quel nombre de caractères. Le résultat est l'affichage des fichiers « test » et « testfile ». Ainsi, si on utilise ensuite la commande « rm test* », les fichiers « testfile » et « test » seront tous deux effacés.



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window has a blue header bar with the title and standard window controls. Below the title is a menu bar with "File", "Edit", "Tabs", and "Help". The main area of the terminal displays the following command sequence:

```
pi@raspberrypi:~ $ ls test*
test  testfile
pi@raspberrypi:~ $ rm test*
pi@raspberrypi:~ $ ls -l
total 48
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 6 pi pi 4096 Dec 21 19:23 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
-rw-r--r-- 1 pi pi 0 Dec 21 20:24 file
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 2 pi pi 4096 Dec 21 20:27 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ rm file
pi@raspberrypi:~ $ █
```

Figure 11 : Commande « rm » avec « * »

C. Créer et supprimer des répertoires

Gérer des répertoires

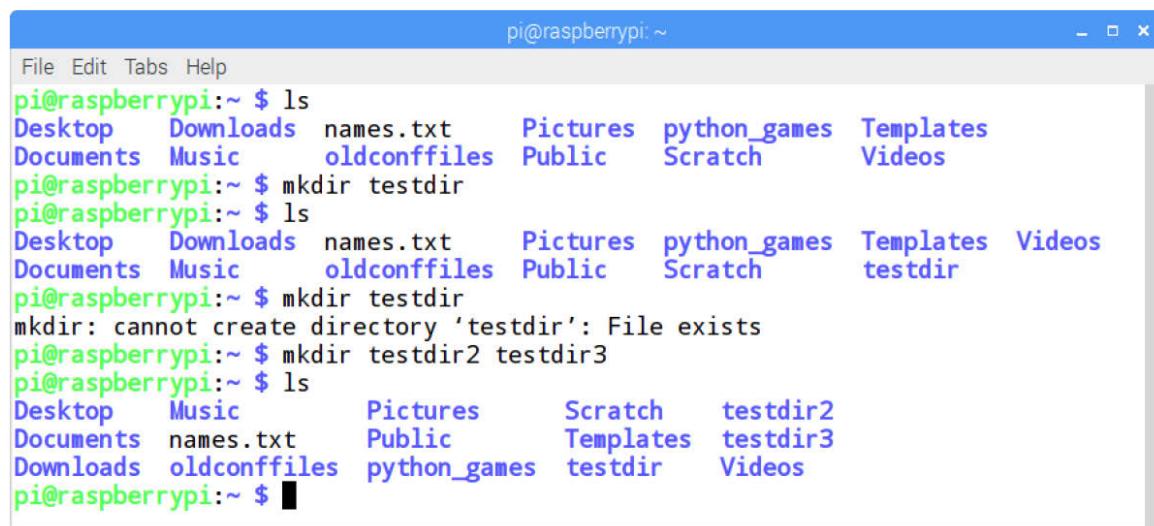
Étape 1

Pour créer des répertoires, on utilise la commande « `mkdir` ». Par exemple, « `mkdir testdir` » crée le répertoire « `testdir` ». Contrairement à la commande « `touch` », la commande « `mkdir` » ne changera pas la date/l'heure du répertoire.

Dans l'exemple à la figure suivante (Figure 12), si on fait à nouveau la commande, on obtient un message d'erreur : « `mkdir : cannot create directory 'testdir' : File exists` » (« `mkdir : on ne peut pas créer le répertoire 'testdir' : Le fichier existe déjà` »).

Étape 2

À l'instar de « `touch` », on peut créer plusieurs répertoires avec une seule commande « `mkdir` ». Par exemple : « `mkdir testdir2 testdir3` » crée deux répertoires, « `testdir2` » et « `testdir3` ». De même, le nom d'un répertoire ne peut pas contenir d'espaces.



```
pi@raspberrypi:~ $ ls
Desktop  Downloads  names.txt  Pictures  python_games  Templates
Documents  Music  oldconffiles  Public  Scratch  Videos
pi@raspberrypi:~ $ mkdir testdir
pi@raspberrypi:~ $ ls
Desktop  Downloads  names.txt  Pictures  python_games  Templates  Videos
Documents  Music  oldconffiles  Public  Scratch  testdir
pi@raspberrypi:~ $ mkdir testdir
mkdir: cannot create directory 'testdir': File exists
pi@raspberrypi:~ $ mkdir testdir2 testdir3
pi@raspberrypi:~ $ ls
Desktop  Music      Pictures      Scratch    testdir2
Documents  names.txt  Public      Templates   testdir3
Downloads  oldconffiles  python_games  testdir    Videos
pi@raspberrypi:~ $ █
```

Figure 12 : Commande « `mkdir` »

Étape 3

On peut créer des répertoires contenus dans des répertoires déjà existants avec une seule commande « `mkdir` ». Par exemple, avec la commande « `mkdir Documents/photos` », on crée le répertoire « `photos` » dans le répertoire `Documents`.

Mais si on essaie de créer un répertoire à l'intérieur d'un répertoire qui n'existe pas, il y aura un message d'erreur. Par exemple, « `mkdir articles/reports` » engendre une erreur parce qu'il n'y a pas de répertoire « `articles` ».

Pour remédier à ce problème, on peut introduire l'option « -p » dans la commande « mkdir ». En utilisant cette commande : « mkdir -p articles/reports », le répertoire « articles » sera créé et dans ce répertoire, le répertoire « reports » sera également créé. En utilisant la commande « ls », on constate qu'il existe un répertoire « articles »; de même, en utilisant la commande « ls articles », on visualise que le répertoire « reports » a été créé à l'intérieur du répertoire « articles ».

```
pi@raspberrypi:~ $ ls
Desktop  Music      Pictures   Scratch  testdir2
Documents names.txt  Public     Templates testdir3
Downloads oldconffiles python_games testdir   Videos
pi@raspberrypi:~ $ mkdir Documents/photos
pi@raspberrypi:~ $ mkdir articles/reports
mkdir: cannot create directory 'articles/reports': No such file or directory
pi@raspberrypi:~ $ mkdir -p articles/reports
pi@raspberrypi:~ $ ls
articles  Downloads oldconffiles python_games testdir   Videos
Desktop   Music      Pictures   Scratch  testdir2
Documents names.txt  Public     Templates testdir3
pi@raspberrypi:~ $ ls articles/
reports
pi@raspberrypi:~ $
```

Figure 13 : Commande « mkdir » avec l'option « -p »

Observation importante!

Sous Linux, la structure de la ligne de commande est toujours : « commande option arguments ». De plus, l'option commence d'habitude par « - »; quant à lui, l'argument est soit un fichier ou une structure de répertoires.

Supprimer les répertoires

Étape 1

Avant de supprimer par mégarde un répertoire, on utilise la commande « ls » afin d'identifier les répertoires du répertoire courant. On peut ensuite supprimer le répertoire « testdir3 » en utilisant la commande « rmdir testdir3 »; puis on vérifie la réussite de l'opération en utilisant à nouveau la commande « ls »; on constate la disparition du répertoire « testdir3 ».

Étape 2

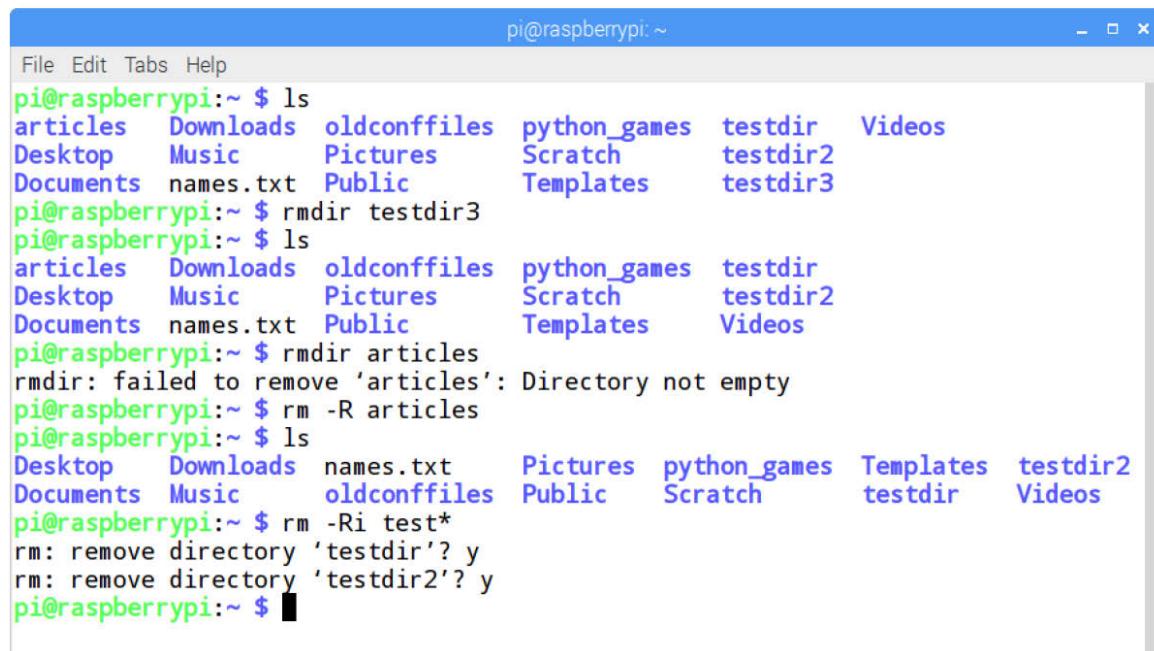
Essayons maintenant de supprimer le répertoire « articles » (qui contient le répertoire « reports »); la commande « rmdir articles » échoue avec le message d'erreur : « rmdir : failed to remove articles : Directory not empty » (traduction libre : « rmdir : élimination des articles échouée : Répertoire non vide »). À noter que la commande « rmdir » supprime seulement les répertoires vides (qui ne contiennent pas de fichiers ou d'autres répertoires).

Étape 3

Pour supprimer un répertoire qui n'est pas vide, on utilise la commande « rm », mais avec l'option « -R »; R pour « recursive » (traduction libre : « récursif »).

Étape 4

Pour supprimer plusieurs répertoires, on utilise le caractère générique « * ». Par contre, on utilise aussi l'option « -i » (pour « interactive ») qui génère un message demandant la confirmation. Ainsi, la commande « rm -Ri test* » supprimera tous les répertoires dont le nom commence par « test », en demandant pour chaque répertoire la confirmation de l'utilisateur : « rm : remove directory 'testdir' ? » (traduction libre : rm : confirmer suppression de 'testdir'?); si la réponse est "y" (pour « yes »), l'opération de suppression aura lieu.



The screenshot shows a terminal window titled 'pi@raspberrypi:~'. The window contains the following command-line session:

```
pi@raspberrypi:~ $ ls
articles Downloads oldconffiles python_games testdir Videos
Desktop Music Pictures Scratch testdir2
Documents names.txt Public Templates testdir3
pi@raspberrypi:~ $ rmdir testdir3
pi@raspberrypi:~ $ ls
articles Downloads oldconffiles python_games testdir
Desktop Music Pictures Scratch testdir2
Documents names.txt Public Templates Videos
pi@raspberrypi:~ $ rmdir articles
rmdir: failed to remove 'articles': Directory not empty
pi@raspberrypi:~ $ rm -R articles
pi@raspberrypi:~ $ ls
Desktop Downloads names.txt Pictures python_games Templates testdir2
Documents Music oldconffiles Public Scratch testdir Videos
pi@raspberrypi:~ $ rm -Ri test*
rm: remove directory 'testdir'? y
rm: remove directory 'testdir2'? y
pi@raspberrypi:~ $
```

Figure 14 : Commande « rmdir » avec les options « -Ri »

D. Déplacer et renommer des fichiers et des répertoires

Déplacer des fichiers

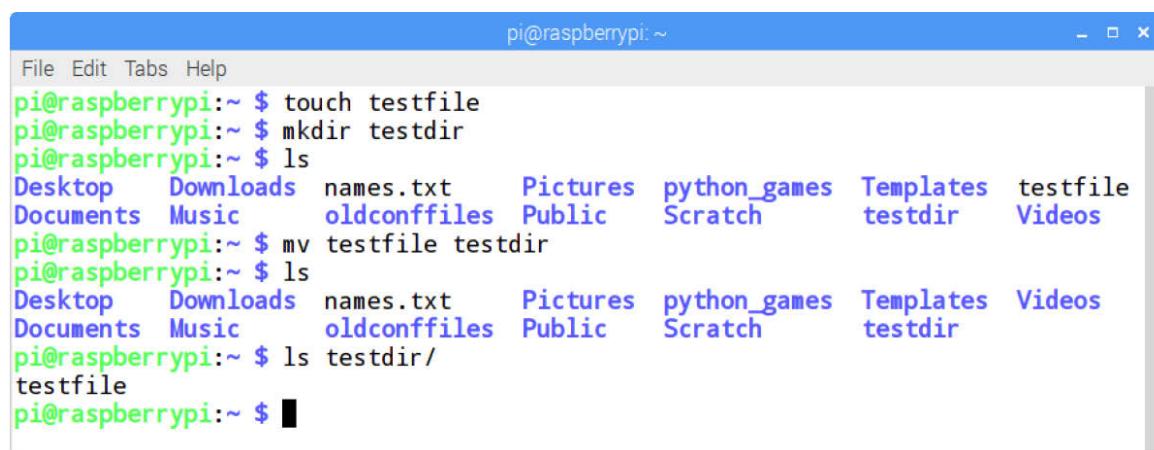
Étape 1

Pour commencer, créez un fichier (avec « touch testfile ») et un répertoire (avec « mkdir testdir ») dans votre répertoire courant. Vérifiez ensuite leur création avec la commande « ls ».

La commande « mv » permet de déplacer les fichiers et les répertoires. Elle est différente des autres commandes utilisées jusqu'à présent, dans le sens qu'elle contient **deux** arguments : le premier étant la source (le fichier ou le répertoire à déplacer) et le second la destination.

Étape 2

En utilisant la commande « mv testfile testdir », le fichier « testfile » sera enlevé du répertoire courant et déplacé ensuite dans le répertoire « testdir ». En utilisant « ls », on valide la disparition du fichier du répertoire courant et avec « ls testdir » l'apparition du même fichier dans le répertoire « testdir ».



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The terminal displays the following sequence of commands:

```
pi@raspberrypi:~ $ touch testfile
pi@raspberrypi:~ $ mkdir testdir
pi@raspberrypi:~ $ ls
Desktop Downloads names.txt Pictures python_games Templates testfile
Documents Music oldconffiles Public Scratch testdir Videos
pi@raspberrypi:~ $ mv testfile testdir
pi@raspberrypi:~ $ ls
Desktop Downloads names.txt Pictures python_games Templates Videos
Documents Music oldconffiles Public Scratch testdir
pi@raspberrypi:~ $ ls testdir/
testfile
pi@raspberrypi:~ $ █
```

Figure 15 : Commande « mv »

Étape 3

Déplacer les répertoires qui contiennent des fichiers se fait de la même manière : entrez « mv testdir newparent » pour déplacer le répertoire « testdir » à l'intérieur du répertoire « newparent »; pour confirmer, on se déplace à l'intérieur du répertoire testdir avec la commande « cd newparent/testdir », puis avec « ls » on constate que « testfile » s'y trouve.

Étape 4

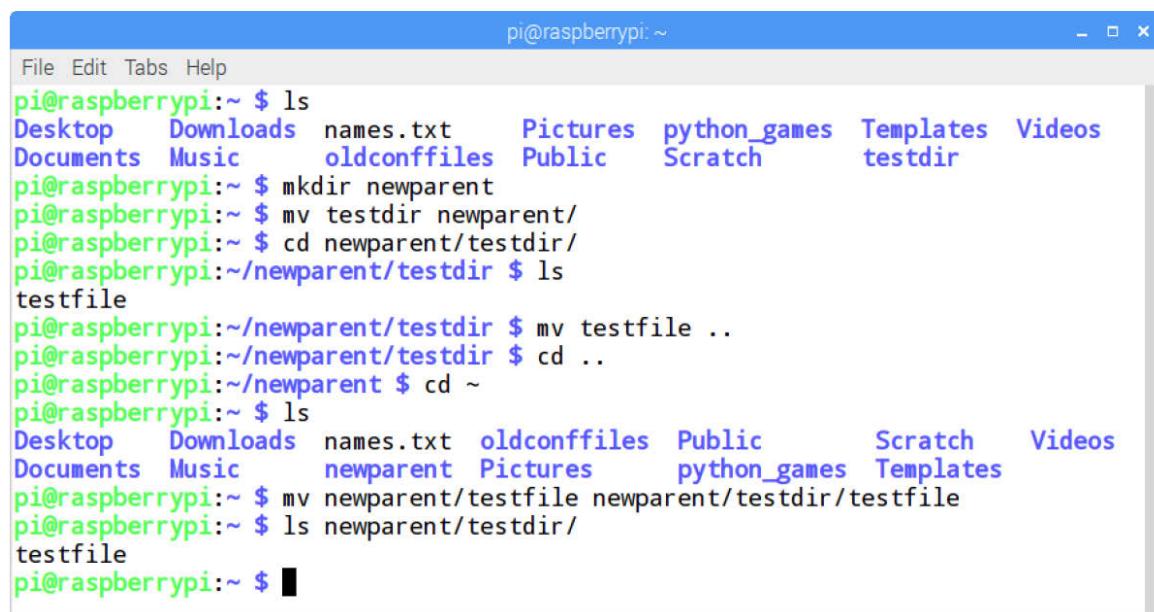
Les fichiers et répertoires peuvent être déplacés au niveau supérieur de l'arbre hiérarchique en utilisant « .. » comme argument.

Ainsi, pour déplacer le fichier « testfile » dans le répertoire parent « newparent », on utilise la commande : « mv testfile .. ». La commande « cd .. » changera le répertoire de travail pour le répertoire de niveau supérieur, c'est-à-dire « newparent » (voir Figure 16).

Étape 5

La commande « cd ~ » changera le répertoire de travail pour notre répertoire personnel.

La commande « mv newparent/testfile newparent/testdir/testfile » déplacera le fichier « testfile » dans son ancien emplacement (répertoire « newparent », à l'intérieur du répertoire « testdir »). La commande « ls newparent/testdir » permet de constater que le fichier « testfile » est à l'intérieur de « testdir ».



The screenshot shows a terminal window titled "pi@raspberrypi:~". The window contains the following command-line session:

```
pi@raspberrypi:~ $ ls
Desktop  Downloads  names.txt  Pictures  python_games  Templates  Videos
Documents  Music    oldconffiles  Public    Scratch      testdir

pi@raspberrypi:~ $ mkdir newparent
pi@raspberrypi:~ $ mv testdir newparent/
pi@raspberrypi:~ $ cd newparent/testdir/
pi@raspberrypi:~/newparent/testdir $ ls
testfile

pi@raspberrypi:~/newparent/testdir $ mv testfile ..
pi@raspberrypi:~/newparent/testdir $ cd ..
pi@raspberrypi:~/newparent $ cd ~
pi@raspberrypi:~ $ ls
Desktop  Downloads  names.txt  oldconffiles  Public    Scratch      Videos
Documents  Music    newparent  Pictures    python_games  Templates

pi@raspberrypi:~ $ mv newparent/testfile newparent/testdir/testfile
pi@raspberrypi:~ $ ls newparent/testdir/
testfile

pi@raspberrypi:~ $
```

Figure 16 : Commandes « mv », « cd .. » et « cd ~ »

Renommer fichiers et répertoires

Étape 1

La commande « mv » peut également servir à renommer les fichiers et les répertoires.

Si on crée d'abord un fichier en utilisant la commande « touch testfile » (et ce, dans le répertoire personnel, ~); la commande « mv testfile names » permettra de changer le nom de « testfile » pour « names ». La même commande sera utilisée pour renommer un répertoire : « mv newparent people » change le nom du répertoire « newparent » pour « people » (voir Figure 17).

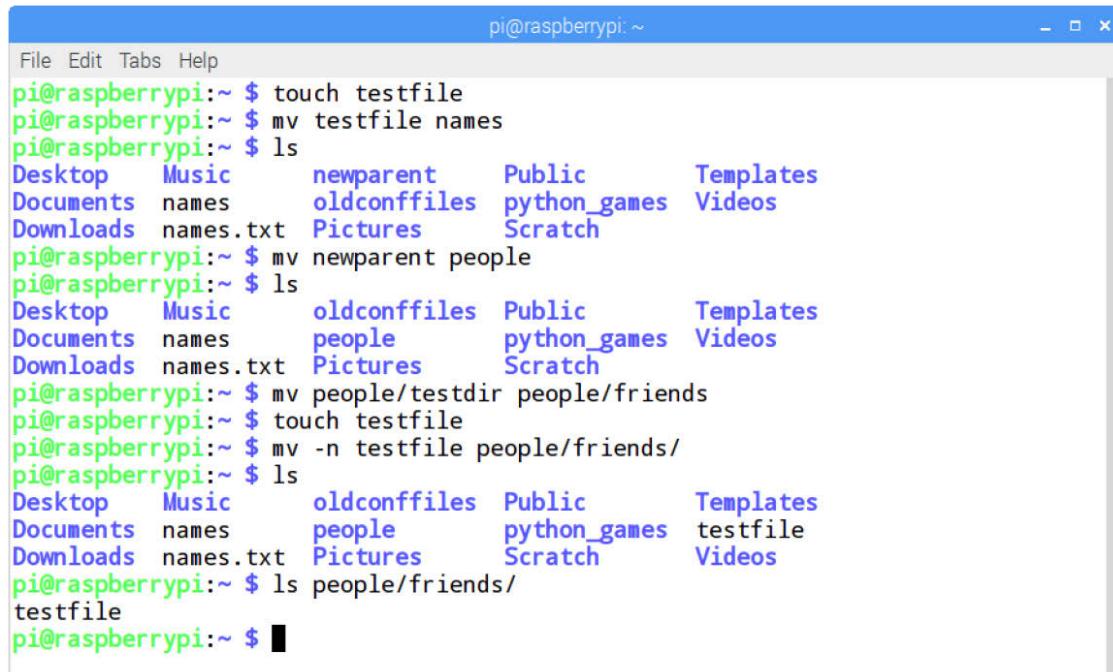
Étape 2

Il est possible de renommer des répertoires qui se trouvent à l'intérieur d'autres répertoires. Pour changer le nom du répertoire « testdir » (qui se trouve maintenant à l'intérieur du répertoire « people ») pour « friends » : « mv people/testdir people/friends » (voir Figure 17).

Étape 3

Il est possible, en voulant renommer un fichier, d'écraser accidentellement un fichier de même nom. Pour éviter cette situation, utilisez l'option « -n » (« no overwrite »).

Dans la figure ci-dessous, Figure 17, la commande « mv -n testfile people/friends/ » n'aura aucun effet puisqu'il existe déjà un fichier « testfile » sous le répertoire « people/friends/ » (voir la commande « ls people/friends »).



```
pi@raspberrypi:~ $ touch testfile
pi@raspberrypi:~ $ mv testfile names
pi@raspberrypi:~ $ ls
Desktop  Music      newparent   Public      Templates
Documents  names    oldconffiles  python_games  Videos
Downloads  names.txt Pictures    Scratch
pi@raspberrypi:~ $ mv newparent people
pi@raspberrypi:~ $ ls
Desktop  Music      oldconffiles  Public      Templates
Documents  names    people      python_games  Videos
Downloads  names.txt Pictures    Scratch
pi@raspberrypi:~ $ mv people/testdir people/friends
pi@raspberrypi:~ $ touch testfile
pi@raspberrypi:~ $ mv -n testfile people/friends/
pi@raspberrypi:~ $ ls people/friends/
testfile
pi@raspberrypi:~ $
```

Figure 17 : Commande « mv » utilisée pour renommer des fichiers et des répertoires

E. Aide intégrée d'utilisation des commandes Linux

Présentation générale

Linux possède une aide intégrée, qui contient des explications sur l'utilisation de toutes les commandes Linux.

La commande « man <nom de la commande> » permet d'accéder à ce manuel d'aide. Pour chaque commande, l'information obtenue contient généralement :

- le nom complet de la commande;
- sa syntaxe;
- une description complète de la commande et de ses options possibles.

```
pi@raspberrypi: ~
File Edit Tabs Help
LS(1) User Commands LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILEs (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
    do not ignore entries starting with .

  -A, --almost-all
    do not list implied . and ..

  --author
    with -l, print the author of each file

  -b, --escape
    print C-style escapes for nongraphic characters
Manual page ls(1) line 1 (press h for help or q to quit)
```

Figure 18 : Accès au manuel d'aide à l'aide de la commande « man »

Pour naviguer dans la page, il faut utiliser les flèches directionnelles (haut et bas) ou les touches page suivante (*page down*) ou page précédente (*page up*). Des exemples d'utilisation, des informations sur de possibles anomalies, ou encore des sites utiles, sont parfois fournis.

```
pi@raspberrypi: ~
File Edit Tabs Help
variable can change the settings. Use the dircolors command to set it.

Exit status:
0      if OK,
1      if minor problems (e.g., cannot access subdirectory),
2      if serious trouble (e.g., cannot access command-line argument).

AUTHOR
Written by Richard M. Stallman and David MacKenzie.

REPORTING BUGS
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report ls translation bugs to <http://translationproject.org/team/>

COPYRIGHT
Copyright © 2014 Free Software Foundation, Inc. License GPLv3+: GNU
GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

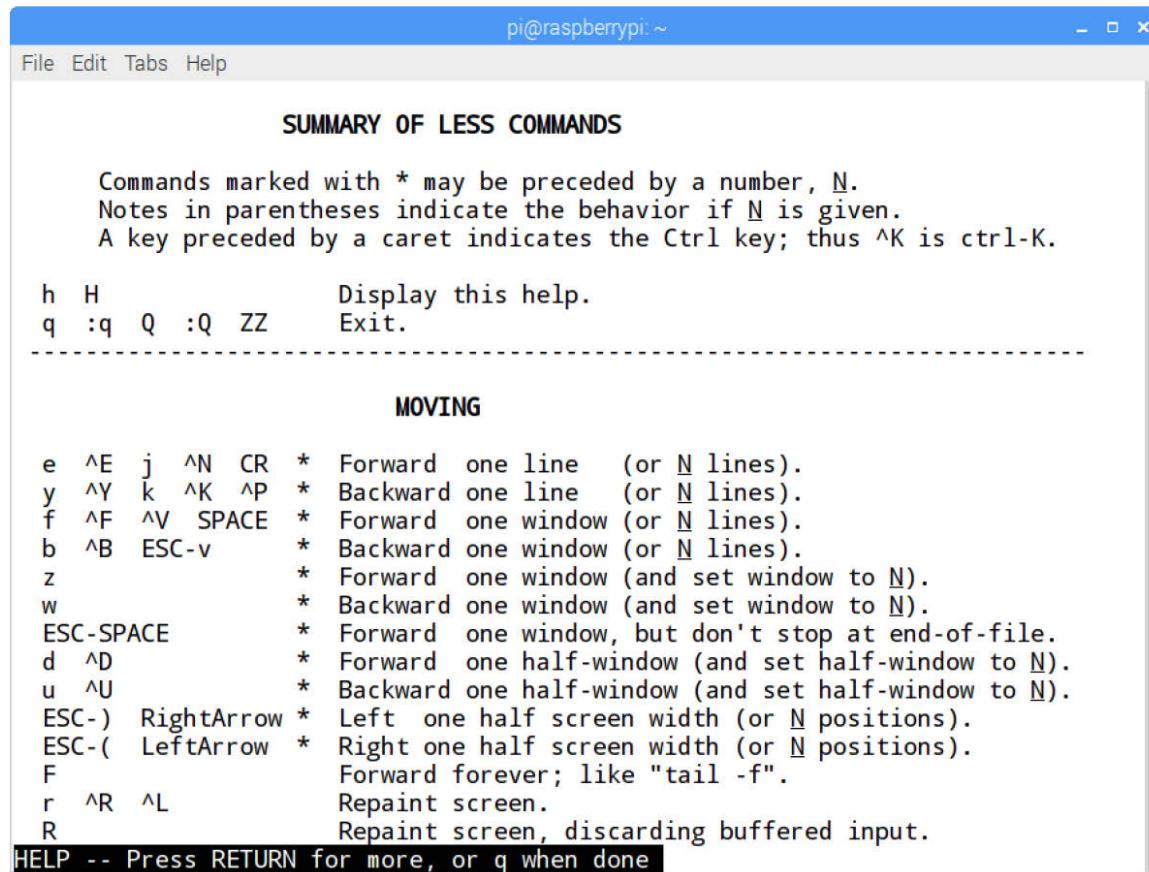
SEE ALSO
Full documentation at: <http://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'

GNU coreutils 8.23          March 2015          LS(1)
Manual page ls(1) line 209/237 (END) (press h for help or q to quit)
```

Figure 19 : Accès au manuel d'aide à l'aide de la commande « man » (suite)

Pour connaître les touches contrôle de la page affichée, appuyez sur « H » (pour « Help ») pour obtenir le « Summary of Less Commands » (traduction libre : Sommaire des commandes page par page) qui contient la description des commandes permettant de se déplacer dans le manuel page par page (voir Figure 20).

Il est possible, en tout temps, d'avancer d'une page (prochaine commande) : touche « z » ou de revenir à la page précédente : touche « w ». La touche « Q » (en majuscule ou minuscule) permet de quitter le manuel.



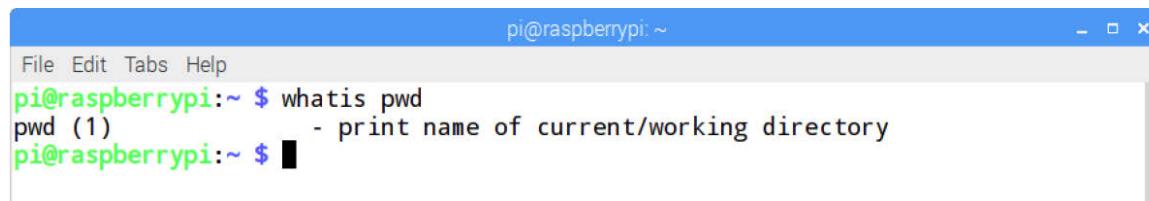
The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window contains a "SUMMARY OF LESS COMMANDS" section. It includes notes about commands marked with *, backward movement, and Ctrl key usage. Below this is a table of key mappings for moving, with a note at the bottom for the "HELP" command.

MOVING						
e	^E	j	^N	CR	*	Forward one line (or <u>N</u> lines).
y	^Y	k	^K	^P	*	Backward one line (or <u>N</u> lines).
f	^F	^V	SPACE		*	Forward one window (or <u>N</u> lines).
b	^B	ESC-v			*	Backward one window (or <u>N</u> lines).
z					*	Forward one window (and set window to <u>N</u>).
w					*	Backward one window (and set window to <u>N</u>).
ESC-SPACE					*	Forward one window, but don't stop at end-of-file.
d	^D				*	Forward one half-window (and set half-window to <u>N</u>).
u	^U				*	Backward one half-window (and set half-window to <u>N</u>).
ESC-)	RightArrow				*	Left one half screen width (or <u>N</u> positions).
ESC-(LeftArrow				*	Right one half screen width (or <u>N</u> positions).
F						Forward forever; like "tail -f".
r	^R	^L				Repaint screen.
R						Repaint screen, discarding buffered input.

HELP -- Press RETURN for more, or q when done

Figure 20 : Touches contrôle en mode page par page

Pour n'obtenir qu'un sommaire d'une commande, ce qui peut s'avérer utile, il suffit d'utiliser la commande « whatis ». Par exemple « whatis pwd » affiche « print name of current/working directory » (traduction libre : « Affiche le nom du répertoire courant / de travail »). La fonction « whatis » permet donc d'effectuer une recherche dans le manuel par mot-clé.



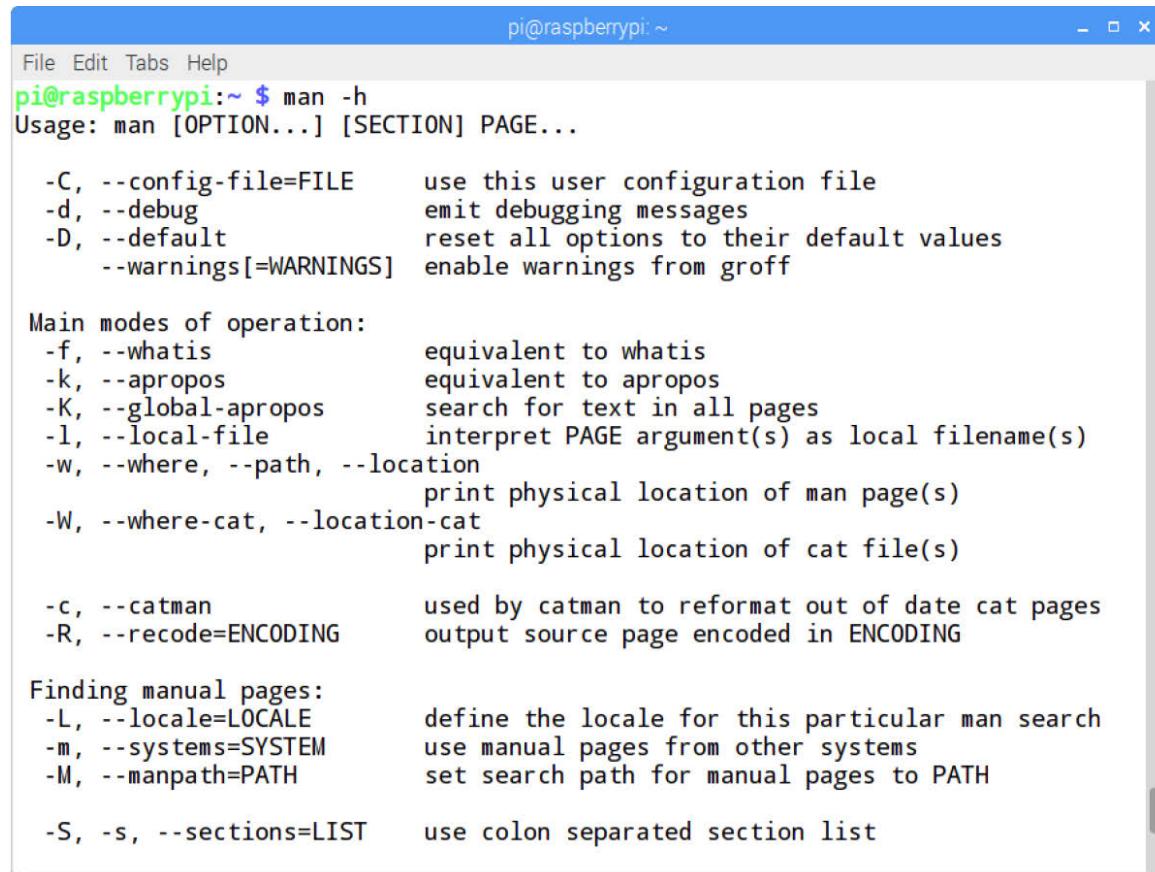
The screenshot shows a terminal window titled "pi@raspberrypi: ~". The user types "whatis pwd" and receives a single-line response: "pwd (1) - print name of current/working directory".

Figure 21 : Commande « whatis »

Quant à elle, la commande « apropos » permet d'effectuer une recherche par mot-clé sur les descriptions. Ainsi, la commande « apropos » permet de retrouver une commande si son nom est inconnu.

Les options de « man »

Pour connaître les options disponibles avec la commande « man », il suffit d'utiliser la commande « man » avec l'option du même nom : « man man ». Par contre, il est plus pratique d'écrire la commande « man -h » ou, encore mieux, d'utiliser la commande « man -h | less » qui permet de lire la description complète un écran à la fois.



The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The window contains the output of the command 'man -h'. The output is as follows:

```
File Edit Tabs Help
pi@raspberrypi:~ $ man -h
Usage: man [OPTION...] [SECTION] PAGE...
      -C, --config-file=FILE      use this user configuration file
      -d, --debug                emit debugging messages
      -D, --default              reset all options to their default values
      --warnings[=WARNINGS]      enable warnings from groff

Main modes of operation:
      -f, --whatis               equivalent to whatis
      -k, --apropos              equivalent to apropos
      -K, --global-apropos       search for text in all pages
      -l, --local-file           interpret PAGE argument(s) as local filename(s)
      -w, --where, --path, --location
                                print physical location of man page(s)
      -W, --where-cat, --location-cat
                                print physical location of cat file(s)

      -c, --catman               used by catman to reformat out of date cat pages
      -R, --recode=ENCODING      output source page encoded in ENCODING

Finding manual pages:
      -L, --locale=LOCALE         define the locale for this particular man search
      -m, --systems=SYSTEM        use manual pages from other systems
      -M, --manpath=PATH          set search path for manual pages to PATH

      -S, -s, --sections=LIST    use colon separated section list
```

Figure 22 : Options de la commande « man »

L'option « -k » est de loin la plus intéressante de la commande « man ». Celle-ci permet de chercher parmi une sélection plus large de pages en rapport avec le sujet. Ainsi, la commande : « man -k directory » affiche toutes les pages relatives aux répertoires.

```
pi@raspberrypi:~ $ man -k directory
alphasort (3)           - scan a directory for matching entries
basename (1)            - strip directory and suffix from filenames
bindtextdomain (3)       - set directory containing message catalogs
chacl (1)               - change the access control list of a file or directory
chdir (2)               - change working directory
chroot (2)              - change root directory
chroot (8)              - run command or interactive shell with special root dir...
closedir (3)            - close a directory
dbus-cleanup-sockets (1) - clean up leftover sockets in a directory
depmod.d (5)             - Configuration directory for depmod
dir (1)                 - list directory contents
dirfd (3)               - get directory stream file descriptor
fchdir (2)              - change working directory
fdopendir (3)            - open a directory
File::BaseDir (3pm)       - Use the freedesktop.org base directory specification
File::Listing (3pm)      - parse directory listing
find (1)                 - search for files in a directory hierarchy
futimesat (2)            - change timestamps of a file relative to a directory fi...
get_current_dir_name (3) - get current working directory
getcwd (2)               - get current working directory
getcwd (3)               - get current working directory
getdents (2)              - get directory entries
getdents64 (2)            - get directory entries
getdirent (3)             - get directory entries in a filesystem-independent format
getwd (3)                - get current working directory
git-clone (1)             - Clone a repository into a new directory
git-mv (1)               - Move or rename a file, a directory, or a symlink
git-stash (1)             - Stash the changes in a dirty working directory away
```

Figure 23 : Commande « man -k »

Modifier le contenu de fichiers texte

Certaines commandes de Linux permettent de modifier le contenu de fichiers texte. Pour ce faire, l'exemple d'un fichier existant sera utilisé.

Étape 1

Avant d'utiliser la commande Linux, on affiche le contenu du fichier texte (voir Figure 24).

- Sous « Menu » de l'interface graphique du Raspberry Pi, cliquez sur « Accessories > Text Editor »;
- dans le menu « File » de la fenêtre (Untitled) qui apparaît, cliquez sur « Open... »;
- cliquez ensuite sur « File System », à gauche de la fenêtre, puis sur le répertoire « home > pi > Documents »;
- finalement, double-cliquez sur le fichier débutant par « rgbt » (ou tout autre fichier texte qu'on y trouvera), ce qui ouvre le fichier et affiche son contenu. Ce fichier contient 15 caractères (voir Figure 25);
- fermez ensuite le fichier.

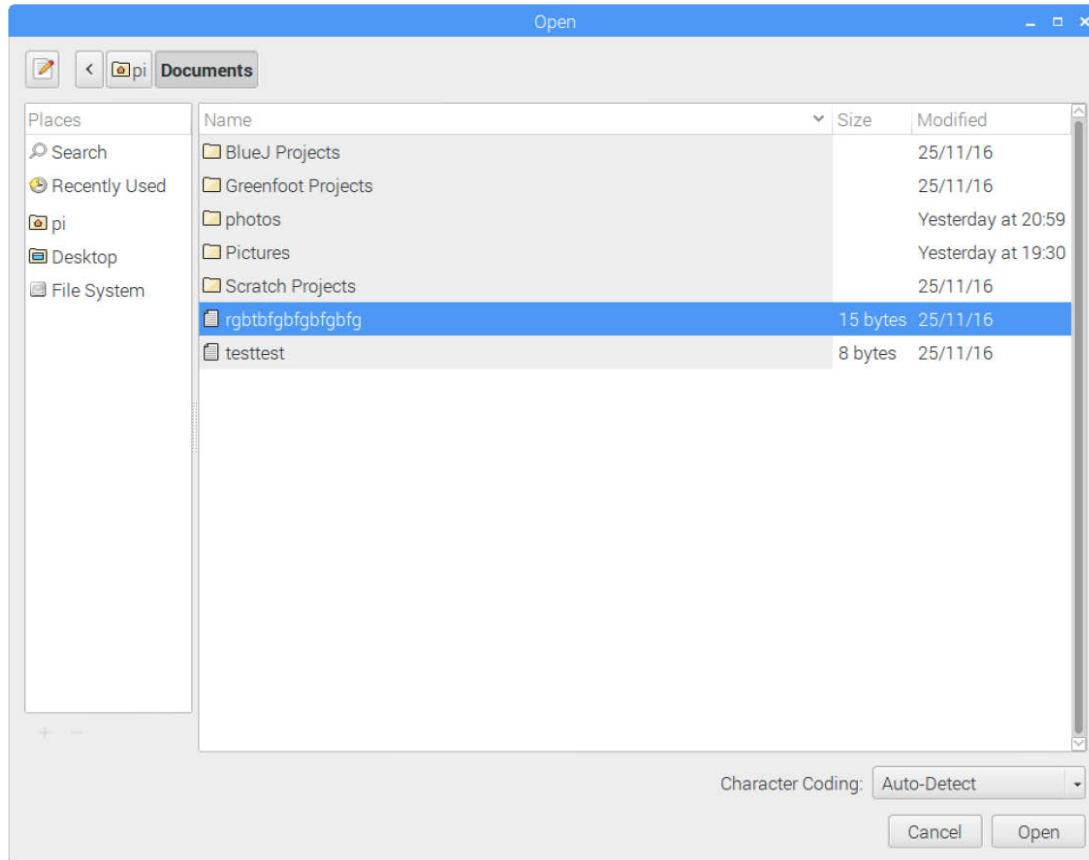


Figure 24 : Afficher le contenu d'un fichier texte

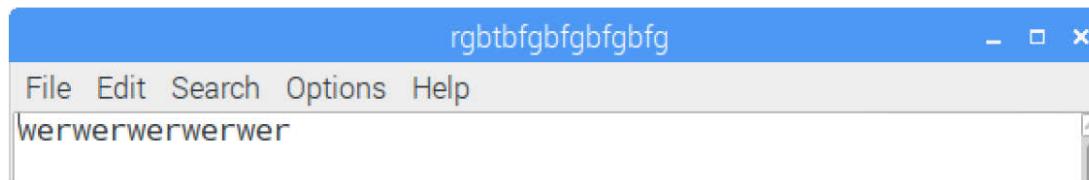


Figure 25 : Exemple de contenu d'un fichier texte

Étape 2

Dans la fenêtre du terminal Raspberry Pi, utilisez la commande « clear » pour effacer l'écran. La commande « ls -l Documents » indique que le fichier débutant par « rgbt » contient 15 caractères, comme obtenu à l'étape 1.

Étape 3

Pour afficher le contenu d'un fichier texte, la commande « cat » est utilisée. Par exemple, « cat Documents/rgbt... » permet de visualiser le contenu du fichier débutant par « rgbt » soit les caractères : « werwerwer... ».

```
pi@raspberrypi:~ $ ls -l Documents
total 28
drwxr-xr-x 11 pi pi 4096 Nov 25 17:49 BlueJ Projects
drwxr-xr-x  5 pi pi 4096 Nov 25 17:49 Greenfoot Projects
drwxr-xr-x  2 pi pi 4096 Dec 21 20:59 photos
drwxr-xr-x  2 pi pi 4096 Dec 21 19:30 Pictures
-rw-r--r--  1 pi pi 15 Nov 25 18:10 rgbtbfgbfbgbfg
drwxr-xr-x  2 pi pi 4096 Nov 25 17:46 Scratch Projects
-rw-r--r--  1 pi pi 8 Nov 25 20:21 testtest
pi@raspberrypi:~ $ cat Documents/rgbtbfgbfbgbfg
werwerwerwerwerpi@raspberrypi:~ $
```

Figure 26 : Commande « cat »

Étape 4

Pour modifier un fichier texte, on peut utiliser le « Text Editor » ou, encore, télécharger un éditeur de texte sous Linux, comme « nano » ou « vim ».



Figure 27 : Interface de l'éditeur de texte « nano » – début de modification

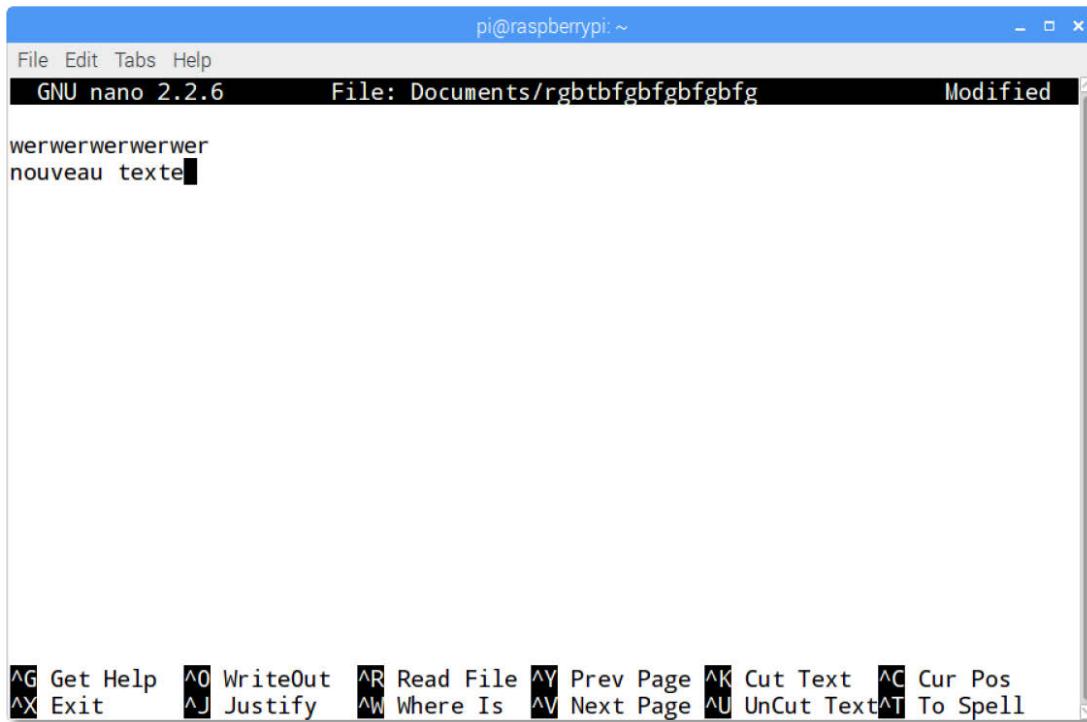


Figure 28 : Interface de l'éditeur de texte « nano » – fin de modification

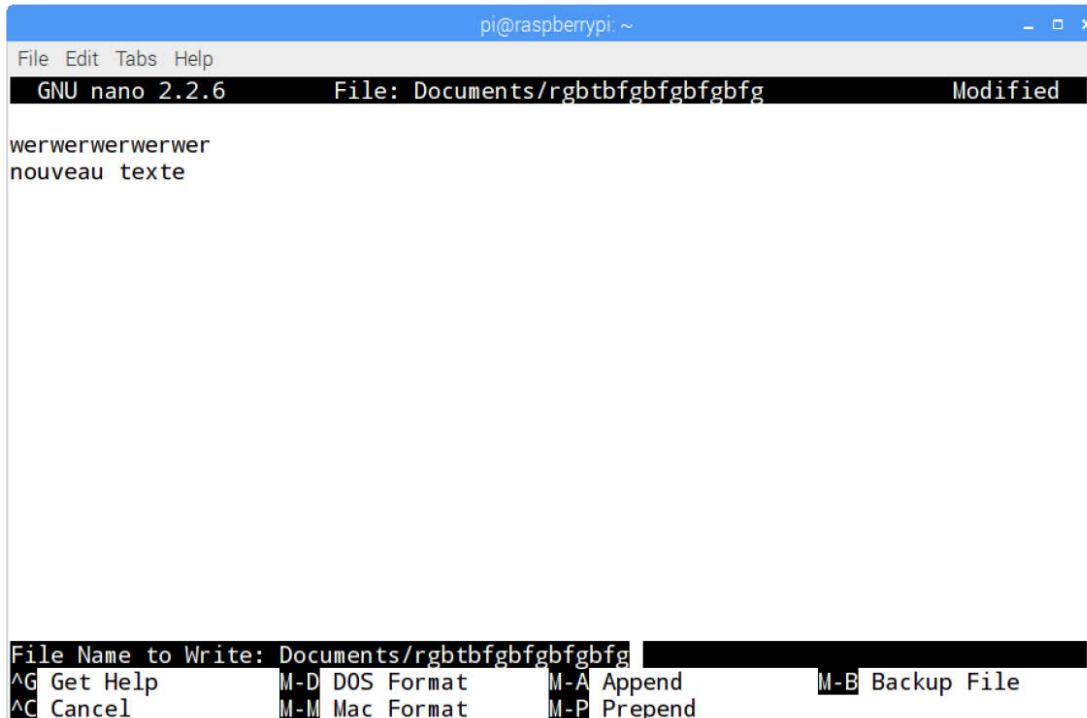


Figure 29 : Interface de l'éditeur de texte « nano » – sauvegarde du fichier

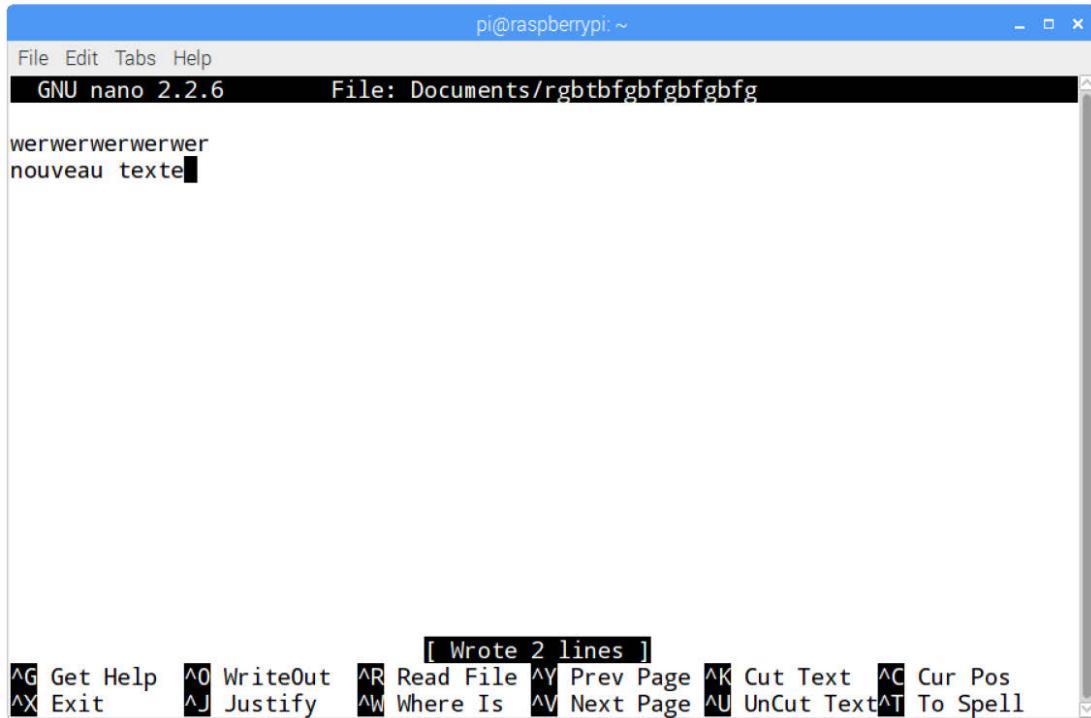


Figure 30 : Interface de l'éditeur de texte « nano » – modification enregistrée

```
pi@raspberrypi:~ $ nano Documents/rgbtbfgbfgbfgbfg
pi@raspberrypi:~ $ cat Documents/rgbtbfgbfgbfgbfg
werwerwerwerwer
nouveau texte
pi@raspberrypi:~ $ ls -l Documents
total 28
drwxr-xr-x 11 pi pi 4096 Nov 25 17:49 BlueJ Projects
drwxr-xr-x  5 pi pi 4096 Nov 25 17:49 Greenfoot Projects
drwxr-xr-x  2 pi pi 4096 Dec 21 20:59 photos
drwxr-xr-x  2 pi pi 4096 Dec 21 19:30 Pictures
-rw-r--r--  1 pi pi   30 Dec 22 14:40 rgbtbfgbfgbfgbfg
drwxr-xr-x  2 pi pi 4096 Nov 25 17:46 Scratch Projects
-rw-r--r--  1 pi pi    8 Nov 25 20:21 testtest
pi@raspberrypi:~ $
```

Figure 31 : Commande « ls -l » pour fichier modifié

F. Commandes de gestion des utilisateurs

Généralités

Étape 1

La commande « whoami » affiche l'identificateur de l'utilisateur au nom duquel on exécute les commandes couramment; dans notre cas, la réponse devrait être « pi » (ou différente, selon l'identificateur d'utilisateur établi pendant l'installation du système).

Chaque utilisateur possède des autorisations qui lui sont propres par rapport aux commandes possibles.

Par exemple, la commande « mkdir /testdir », affichera le message : « mkdir : cannot create directory 'testdir' : Permission denied » (traduction libre : « mkdir : le répertoire 'testdir' ne peut être créé : Autorisation non accordée »). Ce message est affiché, car la création d'un nouveau répertoire dans le répertoire racine n'est pas permise.

Étape 2

Dans ce cas, il faut utiliser la commande « sudo » (« super do »), ce qui signifie faire quelque chose avec les autorisations d'un utilisateur supérieur qui est généralement l'utilisateur racine (« root user »).

Par exemple, avec la commande « sudo useradd -m lucy » permet de créer un utilisateur « lucy ». Le système demandera alors de saisir le mot de passe de l'utilisateur racine pour confirmer au système qu'on a le droit de créer de nouveaux utilisateurs.

Étape 3

Le mot de passe pour l'utilisateur « lucy » sera créé avec la commande « sudo passwd lucy ». Pour toute création d'un mot de passe, une validation est requise (double saisie).

La commande « ls -l /home » permettra d'obtenir la liste des répertoires par utilisateur sous « home ». Pour se connecter sous le compte « lucy », on utilise : « su lucy » et on entre le mot de passe créé pour « lucy ». La commande invite est changée pour « lucy@raspberrypi », même si le répertoire de travail est toujours « /home/pi » (voir commande « pwd » dans la Figure 32 ci-dessous). La commande « whoami » permet de vérifier quel est le compte utilisateur (voir commande « whoami » dans la Figure 32 également).

Étape 4

On peut constater, avec la commande « touch testfile », que l'utilisateur « lucy » ne peut créer des fichiers dans le répertoire « /home/pi », qui est le répertoire racine (de travail). Par contre, si on change le répertoire de travail (avec « cd /home/lucy »), on peut créer des fichiers à l'intérieur de « /home/lucy ». De même, la création de fichiers à l'intérieur de « /home/pi » est possible à condition qu'on change l'identificateur de l'utilisateur courant de « lucy » à « pi » (avec « su pi »).

```
lucy@raspberrypi:/home/pi
File Edit Tabs Help
pi@raspberrypi:~ $ whoami
pi
pi@raspberrypi:~ $ mkdir /testdir
mkdir: cannot create directory '/testdir': Permission denied
pi@raspberrypi:~ $ sudo useradd -m lucy
pi@raspberrypi:~ $ sudo passwd lucy
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $ ls -l /home
total 8
drwxr-xr-x  2 lucy lucy 4096 Dec 22 15:24 lucy
drwxr-xr-x 26 pi   pi   4096 Dec 21 21:45 pi
pi@raspberrypi:~ $ su lucy
Password:
lucy@raspberrypi:/home/pi $ pwd
/home/pi
lucy@raspberrypi:/home/pi $ whoami
lucy
lucy@raspberrypi:/home/pi $ touch testfile
touch: cannot touch 'testfile': Permission denied
lucy@raspberrypi:/home/pi $ su pi
Password:
```

Figure 32 : Commande « whoami »

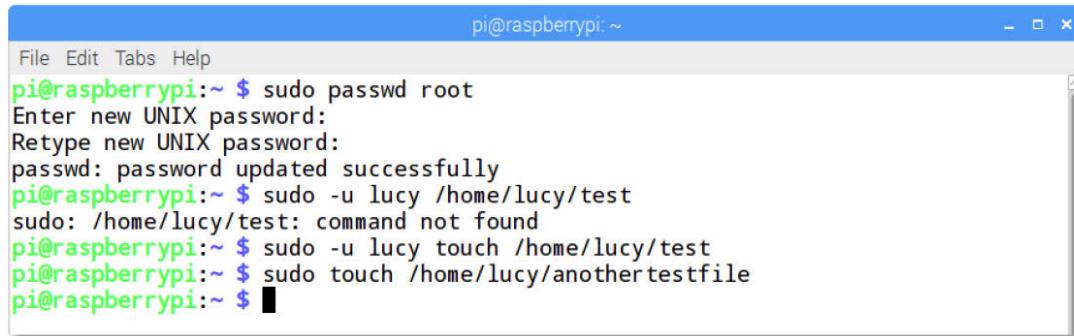
La commande « sudo »

Étape 1

L'utilisateur racine est l'utilisateur qui possède le plus d'autorisations. Il peut, par exemple, créer des fichiers dans n'importe quel répertoire. Pour déterminer ses autorisations, on doit lui attribuer un mot de passe (avec « sudo passwd root »). Pour se connecter sous « root », on utilise la commande « su » comme il est démontré auparavant.

La commande « sudo » permet d'exécuter une commande sous l'identité d'un utilisateur différent; par exemple, « sudo -u lucy touch /home/lucy/test » permet de créer le fichier « test » dans « /home/lucy », même lorsque le répertoire de travail est « /home/pi ».

La commande « sudo » utilisée seule avec l'option « -u » sera exécutée avec l'utilisateur racine, comme si on avait tapé « sudo -u root ». Ainsi, la commande « sudo touch /home/lucy/anotherfile » crée le fichier « anotherfile » dans « /home/lucy », mais avec l'autorité de « root » – même si le répertoire de travail est « /home/pi ».



```
pi@raspberrypi:~ $ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $ sudo -u lucy /home/lucy/test
sudo: /home/lucy/test: command not found
pi@raspberrypi:~ $ sudo -u lucy touch /home/lucy/test
pi@raspberrypi:~ $ sudo touch /home/lucy/another testfile
pi@raspberrypi:~ $
```

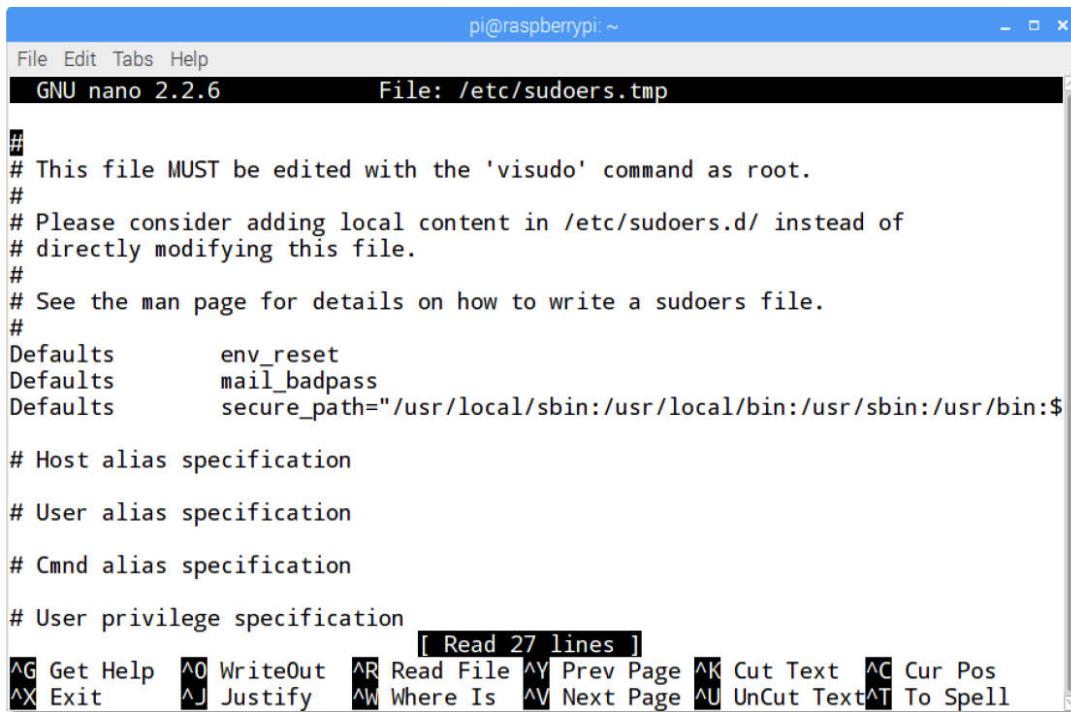
Figure 33 : Commande « sudo »

Étape 2 (cette étape est optionnelle)

La configuration initiale du Raspberry Pi permet seulement à l'utilisateur « pi » d'utiliser « sudo ». Si on veut donner à l'utilisateur « lucy » la même autorisation, on doit introduire l'identificateur « lucy » dans le groupe des « sudoers ».

La commande « sudo visudo » permet d'ajouter des autorisations à un autre utilisateur en éditant le fichier d'autorisations. Pour ce faire, il faut ajouter : « lucy ALL=(ALL) NOPASSWD:ALL » à la dernière ligne du fichier d'autorisations et le sauvegarder avec la touche « Control+O »; supprimez ensuite l'extension « .tmp » du nom du fichier.

Dans la Figure 34 qui suit, on utilise l'éditeur de texte « nano » pour éditer le fichier d'autorisations.



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window title bar also displays "File: /etc/sudoers.tmp". The main area of the window shows the contents of the /etc/sudoers.tmp file, which is a configuration file for the sudo command. The file contains several sections of comments and default settings, followed by sections for host aliases, user aliases, command aliases, and user privilege specifications. At the bottom of the screen, there is a status bar with various keyboard shortcuts for nano editor commands.

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
[ Read 27 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Figure 34 : Modification des autorisations à l'aide de la commande « sudo visudo »

(Tapez la touche « espace » pour voir la suite.)

G. Gestion des propriétés et autorisations

Généralités

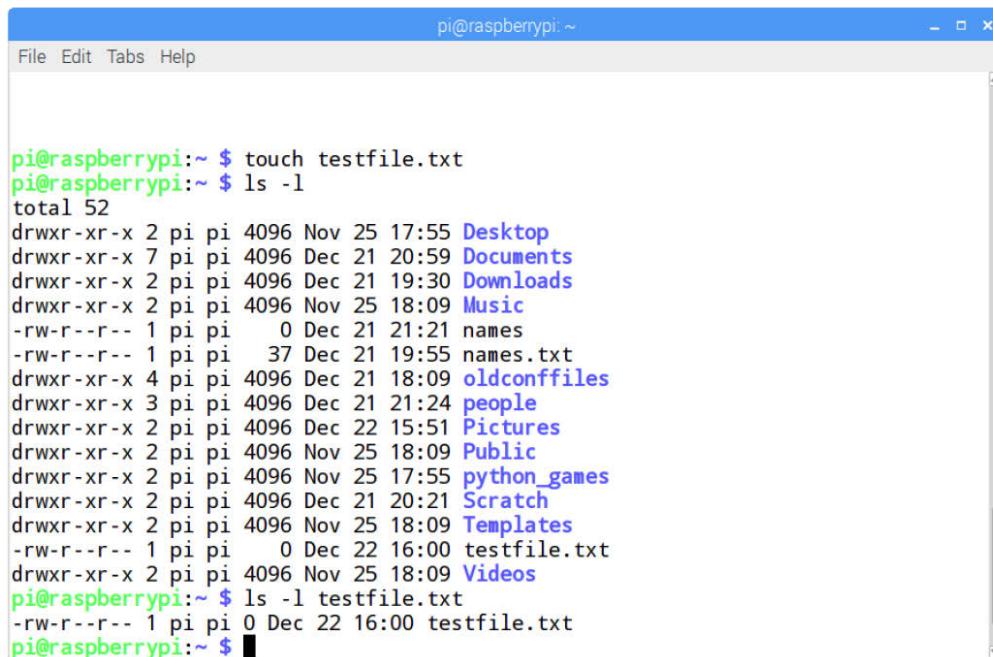
L'utilisation de la commande « ls -l » permet d'afficher la liste des répertoires et fichiers d'un répertoire donné et leurs propriétés. Chaque ligne débute par « - » pour un fichier ou « d », pour un répertoire (voir Figure 35).

S'ensuit un ensemble de neuf caractères décrivant les autorisations que les utilisateurs possèdent vis-à-vis du fichier ou répertoire. Les utilisateurs se divisent en trois groupes :

- le premier groupe (les trois premiers caractères) correspond à l'utilisateur qui a créé le fichier (ou répertoire) et qui en est le propriétaire (*owner*);
- le deuxième groupe (les trois caractères suivants) est celui des utilisateurs qui font partie du même groupe que le propriétaire;
- le dernier (les trois derniers caractères) : celui des autres utilisateurs (*others*).

Ainsi, pour chaque ensemble de trois caractères :

- le premier indique que le groupe concerné peut lire ou non le fichier (ou répertoire); « r » signifie que le groupe possède l'**autorisation** de lire le fichier (ou répertoire) et « - » signifie que le groupe **ne peut pas** lire le fichier;
- le deuxième indique que le groupe peut ou non modifier (*write*) le fichier; elle prend la valeur « w » pour modification permise ou « - » : aucune modification;
- le troisième indique la possibilité de lancer l'exécution d'un fichier exécutable (en vert scintillant) : « r » (pour *run*) si autorisé ou sinon « - ».



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The command "ls -l" is run, displaying the following output:

```
pi@raspberrypi:~ $ touch testfile.txt
pi@raspberrypi:~ $ ls -l
total 52
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 7 pi pi 4096 Dec 21 20:59 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 0 Dec 21 21:21 names
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 3 pi pi 4096 Dec 21 21:24 people
drwxr-xr-x 2 pi pi 4096 Dec 22 15:51 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
-rw-r--r-- 1 pi pi 0 Dec 22 16:00 testfile.txt
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ ls -l testfile.txt
-rw-r--r-- 1 pi pi 0 Dec 22 16:00 testfile.txt
pi@raspberrypi:~ $ █
```

Figure 35 : Propriétés et autorisations des fichiers et des répertoires

Prenons l'exemple de la ligne correspondante au fichier créé soit « testfile.txt » :

```
-rw-r-r-- 1 pi pi 0 May 13 21:14 testfile.txt
```

- le premier caractère « - » indique qu'il s'agit d'un fichier;
- en vert : les trois caractères suivants indiquent que le créateur et propriétaire du fichier peut lire (« r ») et écrire (« w ») dans le fichier, mais pas l'exécuter (« - »);
- en turquoise : les trois caractères suivants indiquent que les utilisateurs qui font partie du même groupe que le propriétaire peuvent seulement lire le fichier « testfile.txt », mais ne peuvent pas le modifier ou l'exécuter;
- en jaune : de même, les trois derniers caractères du groupe indiquent que tout autre utilisateur qui ne fait pas partie des deux premiers groupes peut lire, mais ne peut modifier ou exécuter le fichier « testfile.txt ».

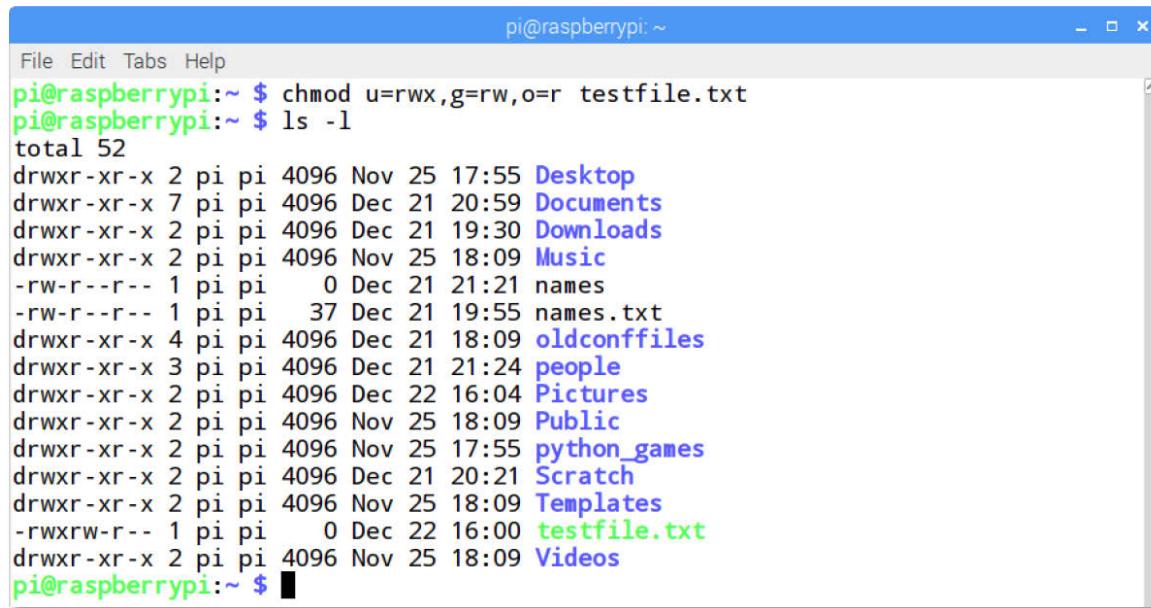
Modifier les autorisations

La modification d'autorisations se fait avec la commande « chmod » et :

- la notation **alpha** pour différencier les différents groupes d'utilisateur soit :
 - « u » pour *owner* (ou propriétaire);
 - « g » pour le groupe (du propriétaire);
 - « o » pour *others*.
- et pour chaque groupe, la notation **octale** : pour assigner les autorisations avec :
 - soit la notation numérique : 4 pour *read*, 2 pour *write* et 1 pour *execute*;
 - soit la notation alpha : « r » pour *read*, « w » pour *write* et « x » pour *execute*.

Par exemple, la commande suivante pour modifier les autorisations du fichier « testfile.txt » : « chmod u=rwx, g=rw, o=r testfile.txt » aura pour effet de changer les autorisations pour les trois groupes d'utilisateurs. Pour les besoins de vérification, la commande « ls » affiche les propriétés suivantes pour « testfile.txt » :

```
-rwxrw-r-- 1 pi pi 0 .....testfile.txt
```



```
pi@raspberrypi:~ $ chmod u=rwx,g=rw,o=r testfile.txt
pi@raspberrypi:~ $ ls -l
total 52
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 Desktop
drwxr-xr-x 7 pi pi 4096 Dec 21 20:59 Documents
drwxr-xr-x 2 pi pi 4096 Dec 21 19:30 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Music
-rw-r--r-- 1 pi pi 0 Dec 21 21:21 names
-rw-r--r-- 1 pi pi 37 Dec 21 19:55 names.txt
drwxr-xr-x 4 pi pi 4096 Dec 21 18:09 oldconffiles
drwxr-xr-x 3 pi pi 4096 Dec 21 21:24 people
drwxr-xr-x 2 pi pi 4096 Dec 22 16:04 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Public
drwxr-xr-x 2 pi pi 4096 Nov 25 17:55 python_games
drwxr-xr-x 2 pi pi 4096 Dec 21 20:21 Scratch
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Templates
-rwxr-wr-- 1 pi pi 0 Dec 22 16:00 testfile.txt
drwxr-xr-x 2 pi pi 4096 Nov 25 18:09 Videos
pi@raspberrypi:~ $ █
```

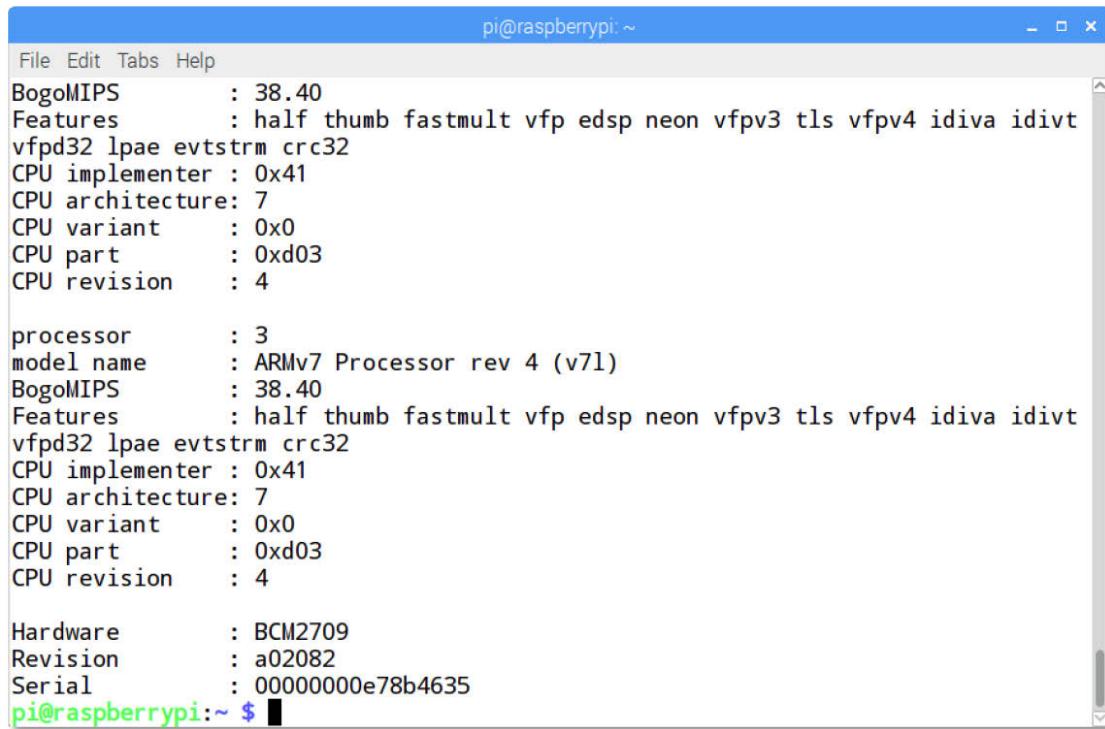
Figure 36 : Modifications des autorisations d'un fichier

H. Commandes Linux pour le système et le disque

Commandes pour le système

La commande « cat » permet d'afficher des informations sur le système. Cette commande affiche diverses informations selon la syntaxe utilisée :

- la commande « cat /proc/cpuinfo » affiche les informations sur les quatre processeurs de Raspberry Pi :



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window contains the following text output from the "cat" command:

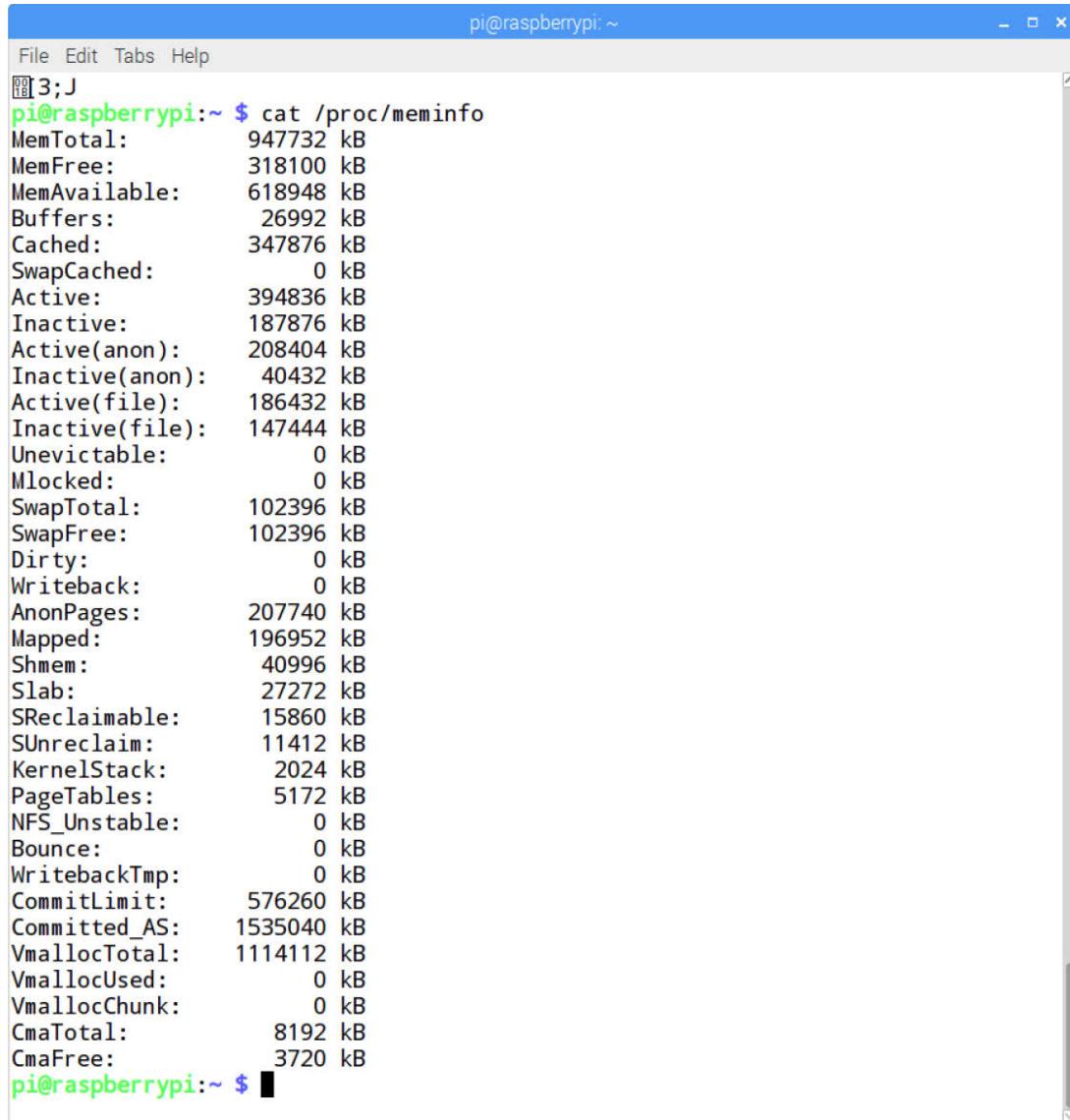
```
BogoMIPS      : 38.40
Features      : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpaes evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4

processor     : 3
model name    : ARMv7 Processor rev 4 (v7l)
BogoMIPS      : 38.40
Features      : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpaes evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4

Hardware      : BCM2709
Revision      : a02082
Serial        : 0000000e78b4635
pi@raspberrypi:~ $
```

Figure 37 : Commande « cat »

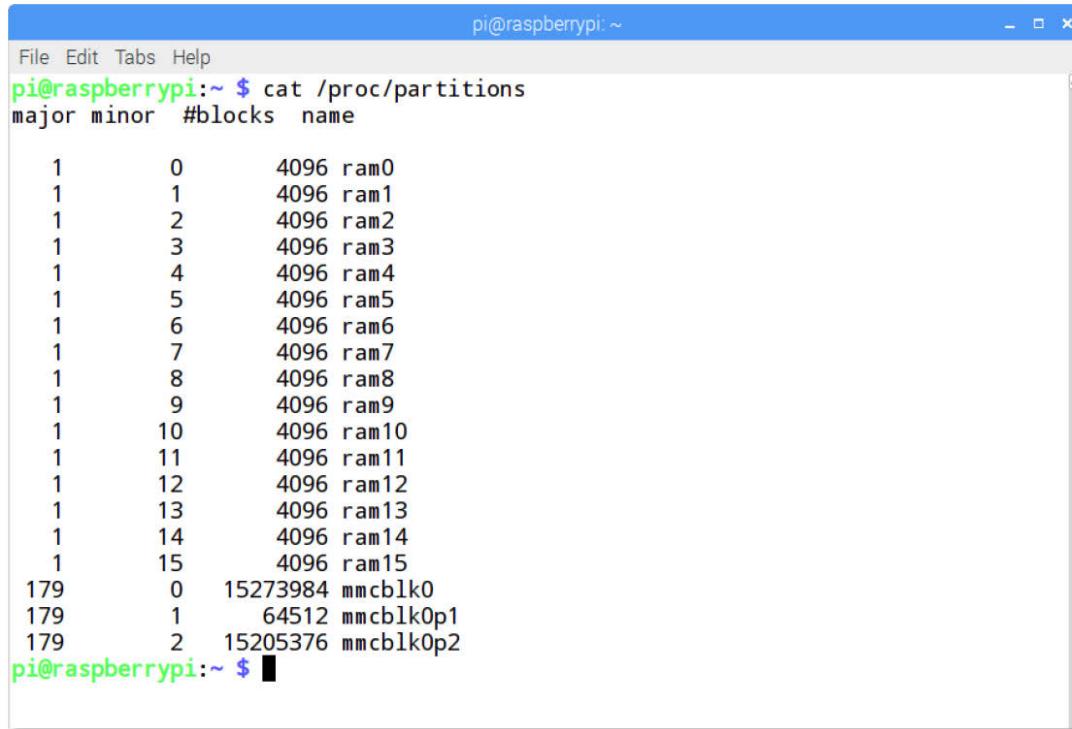
- quant à elle, la commande « cat /proc/meminfo » affiche des informations sur la mémoire :



```
pi@raspberrypi:~ $ cat /proc/meminfo
MemTotal:       947732 kB
MemFree:        318100 kB
MemAvailable:   618948 kB
Buffers:        26992 kB
Cached:         347876 kB
SwapCached:     0 kB
Active:         394836 kB
Inactive:       187876 kB
Active(anon):   208404 kB
Inactive(anon): 40432 kB
Active(file):   186432 kB
Inactive(file): 147444 kB
Unevictable:    0 kB
Mlocked:        0 kB
SwapTotal:      102396 kB
SwapFree:       102396 kB
Dirty:          0 kB
Writeback:      0 kB
AnonPages:      207740 kB
Mapped:         196952 kB
Shmem:          40996 kB
Slab:           27272 kB
SReclaimable:   15860 kB
SUnreclaim:     11412 kB
KernelStack:    2024 kB
PageTables:     5172 kB
NFS_Unstable:   0 kB
Bounce:          0 kB
WritebackTmp:    0 kB
CommitLimit:    576260 kB
Committed_AS:  1535040 kB
VmallocTotal:   1114112 kB
VmallocUsed:    0 kB
VmallocChunk:   0 kB
CmaTotal:       8192 kB
CmaFree:        3720 kB
pi@raspberrypi:~ $
```

Figure 38 : Commande « cat /proc/meminfo »

- la commande « cat /proc/partitions » affiche des informations sur la carte SD :

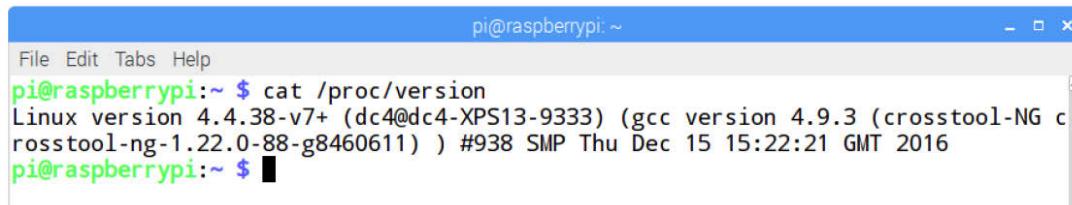


```
pi@raspberrypi:~ $ cat /proc/partitions
major minor #blocks name

1          0      4096 ram0
1          1      4096 ram1
1          2      4096 ram2
1          3      4096 ram3
1          4      4096 ram4
1          5      4096 ram5
1          6      4096 ram6
1          7      4096 ram7
1          8      4096 ram8
1          9      4096 ram9
1         10     4096 ram10
1         11     4096 ram11
1         12     4096 ram12
1         13     4096 ram13
1         14     4096 ram14
1         15     4096 ram15
179        0    15273984 mmcblk0
179        1     64512 mmcblk0p1
179        2   15205376 mmcblk0p2
pi@raspberrypi:~ $
```

Figure 39 : Commande « cat /proc/partitions »

- finalement, la commande « cat /proc/version » affiche la version du Raspberry Pi utilisée :

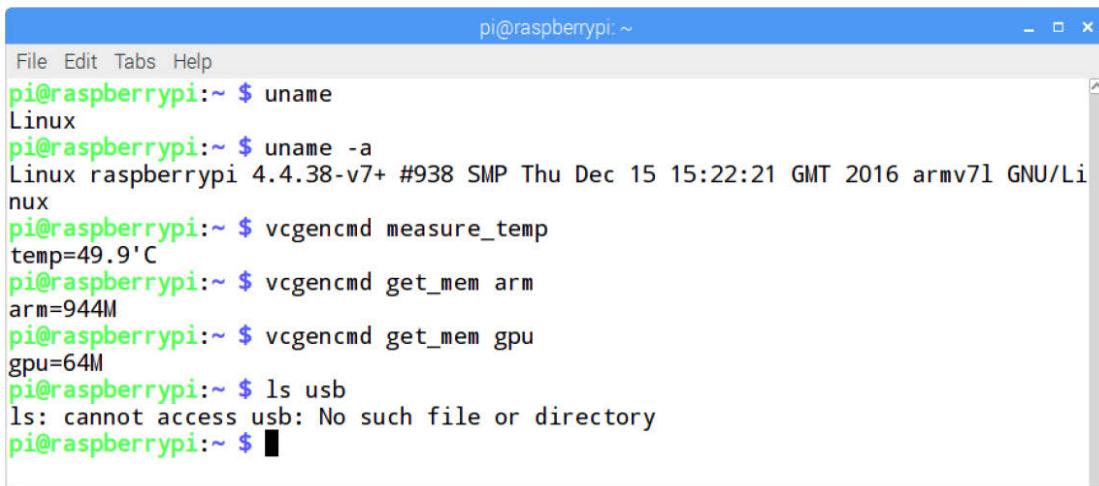


```
pi@raspberrypi:~ $ cat /proc/version
Linux version 4.4.38-v7+ (dc4@dc4-XPS13-9333) (gcc version 4.9.3 (crosstool-NG c
rosstool-ng-1.22.0-88-g8460611) ) #938 SMP Thu Dec 15 15:22:21 GMT 2016
pi@raspberrypi:~ $
```

Figure 40 : Commande « cat /proc/version »

Les autres commandes système couramment utilisées sont les suivantes :

- « uname » : affiche le nom du kernel (noyau) du système d'opération;
- « uname -a » : affiche des détails sur le kernel;
- « vcgencmd measure_temp » : affiche la température du système Raspberry Pi;
- « vcgencmd get_mem arm » : affiche la mémoire RAM disponible;
- « vcgencmd get_mem gpu » : affiche la mémoire de la puce graphique disponible;
- « ls usb » : affiche la liste des dispositifs USB attachés;
- « sudo shutdown -h » : permet d'arrêter Raspberry Pi (h pour *halt*);
- « sudo shutdown -r » : permet de redémarrer Raspberry Pi (r pour *restart*).



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window contains the following command history:

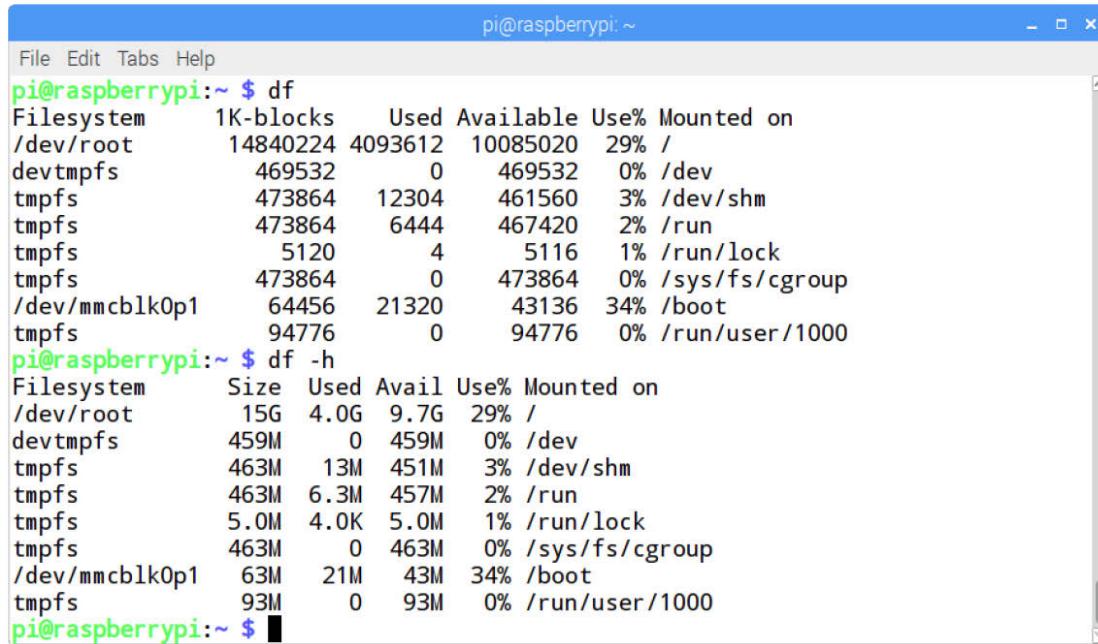
```
File Edit Tabs Help
pi@raspberrypi:~ $ uname
Linux
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 4.4.38-v7+ #938 SMP Thu Dec 15 15:22:21 GMT 2016 armv7l GNU/Li
nux
pi@raspberrypi:~ $ vcgencmd measure_temp
temp=49.9'C
pi@raspberrypi:~ $ vcgencmd get_mem arm
arm=944M
pi@raspberrypi:~ $ vcgencmd get_mem gpu
gpu=64M
pi@raspberrypi:~ $ ls usb
ls: cannot access usb: No such file or directory
pi@raspberrypi:~ $ █
```

Figure 41 : Diverses commandes système

Commandes pour le disque

Il existe également différentes commandes pour obtenir diverses informations sur le disque :

- la commande « df » : affiche une liste des volumes présents sur la carte SD. La carte SD peut être perçue comme un dispositif de mémoire externe qui est divisé en plusieurs partitions pouvant être considérées comme des unités de mémoires externes complètement indépendantes. Chaque partition contient plusieurs volumes, qui sont des aires de stockage des informations;
- la commande « df -h » : affiche la même liste d'une manière plus facile à lire (en G et M octets) :
 - les deux premières lignes correspondent à « rootfs » et « /dev/root » : affichent la taille totale (*Size*), la taille utilisée (*Used*), la taille disponible (*Avail*) et le pourcentage de l'utilisation (*Use%*);
 - ces deux volumes (« rootfs » et « /dev/root ») constituent la partie principale du système d'opération;
 - les autres sont utilisés pour le démarrage et l'initialisation des dispositifs périphériques;



The screenshot shows a terminal window titled "pi@raspberrypi: ~". It displays two command outputs: the first is the standard `df` command output, and the second is the `df -h` command output. Both outputs show disk usage for various file systems.

```
pi@raspberrypi:~ $ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        14840224  4093612  10085020  29% /
devtmpfs          469532      0   469532   0% /dev
tmpfs            473864  12304   461560   3% /dev/shm
tmpfs            473864  6444   467420   2% /run
tmpfs             5120      4   5116   1% /run/lock
tmpfs            473864      0   473864   0% /sys/fs/cgroup
/dev/mmcblk0p1     64456  21320   43136  34% /boot
tmpfs            94776      0   94776   0% /run/user/1000
pi@raspberrypi:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G  4.0G  9.7G  29% /
devtmpfs        459M    0  459M   0% /dev
tmpfs           463M  13M  451M   3% /dev/shm
tmpfs           463M  6.3M  457M   2% /run
tmpfs            5.0M  4.0K  5.0M   1% /run/lock
tmpfs           463M    0  463M   0% /sys/fs/cgroup
/dev/mmcblk0p1     63M   21M   43M  34% /boot
tmpfs           93M    0   93M   0% /run/user/1000
pi@raspberrypi:~ $ █
```

Figure 42 : Diverses commandes pour le disque

- la commande « du » : affiche la taille de mémoire utilisée par les fichiers du répertoire « home » :
 - la commande « du -h | less » permet d'avoir une forme plus compréhensible, car elle affiche une page par écran;
 - de même pour un répertoire, la commande « du -ha python_games » peut être utilisée.

```
pi@raspberrypi:~ $ du -h python_games/
1.8M    python_games/
pi@raspberrypi:~ $ du -ha python_games/
12K    python_games/RedSelector.png
8.0K   python_games/pinkgirl.png
12K    python_games/4row_arrow.png
36K    python_games/match4.wav
84K    python_games/flippybackground.png
32K    python_games/match0.wav
92K    python_games/star_title.png
40K    python_games/match2.wav
68K    python_games/star_solved.png
4.0K   python_games/gameicon.png
40K    python_games/beep3.ogg
4.0K   python_games/grass3.png
4.0K   python_games/blankpygame.py
12K    python_games/Selector.png
12K    python_games/Star.png
8.0K   python_games/boy.png
8.0K   python_games/gem2.png
4.0K   python_games/grass2.png
12K    python_games/slidedepuzzle.py
12K    python_games/inkspillsettingsbutton.png
4.0K   python_games/squirrel.png
16K    python_games/tetrominoforidiots.py
52K    python_games/match1.wav
40K    python_games/badswap.wav
24K    python_games/starpusher.py
8.0K   python_games/Wood_Block_Tall.png
12K    python_games/4row_tie.png
8.0K   python_games/horngirl.png
20K    python_games/4row_computerwinner.png
4.0K   python_games/catanimation.py
12K    python_games/4row_board.png
44K    python_games/match5.wav
```

Figure 43 : Commande « du -ha »

CONCLUSION

Pour en savoir plus sur Linux (et Raspberry), nous vous suggérons de consulter :

1) Les sites suivants :

- <https://www.raspberrypi.org/>
- <http://kali-linux.fr>
- <https://raspbian-france.fr/cours/slides-linux-introduction.html#/11/1>

2) Les revues suivantes :

- [*The Imagine NextTech Series, « Raspberry Pi for Beginners », March 2016.](#)
- [Inside Linux, « Utiliser à fond votre Raspberry Pi : Raspberry Pi Projects », avril, mai-juin 2016, n° 29.](#)

***Note :** Pour illustrer nos propos, certains concepts ou l'usage de certaines commandes, nous nous sommes inspirés de la revue Hors Série de *NextTech* de mars 2016. Tous les exemples illustrés dans ce document ont été réalisés à l'aide de notre réseau informatique personnel expérimental.