

MEWO

# Module 01 :

Comprendre et utiliser les outils essentiels

Timothee SICCHIA  
11/10/2024

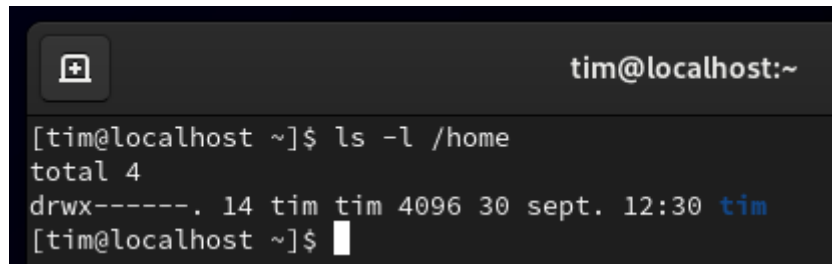
## Table des matières

1. Accéder à un terminal et utiliser les commandes avec la bonne syntaxe .....	2
2. Utiliser la redirection d'entrée/sortie (>, >>,  , 2>, etc.) .....	2
3. Utiliser grep et les expressions régulières pour analyser du texte .....	3
4. Accéder à des systèmes à distance via SSH.....	4
5. Se connecter et changer d'utilisateur sur des cibles multi-utilisateurs .....	5
6. Archiver, compresser, décompresser des fichiers avec tar, star, gzip, bzip2.....	6
7. Créer et éditer des fichiers texte .....	7
8. Créer, supprimer, copier et déplacer des fichiers et dossiers .....	7
9. Créer des liens matériels et logiciels.....	8
10. Gérer les permissions ugo/rwx .....	9
11. Consulter la documentation système (/usr/share/doc).....	9

## 1. Accéder à un terminal et utiliser les commandes avec la bonne syntaxe

Le terminal est l'interface de base pour interagir avec le système Linux. Savoir utiliser les commandes correctement est essentiel.

- **Exemple** : Tapez `ls -l /home` pour afficher une liste détaillée de tous les fichiers et répertoires présents dans le dossier `/home`. Le répertoire `/home` contient les répertoires personnels de chaque utilisateur sur le système. Chaque utilisateur possède un répertoire nommé selon son nom d'utilisateur, où il peut stocker ses fichiers personnels. Cette commande montre les permissions, le propriétaire, la taille et la date de modification de chaque répertoires personnels.



```

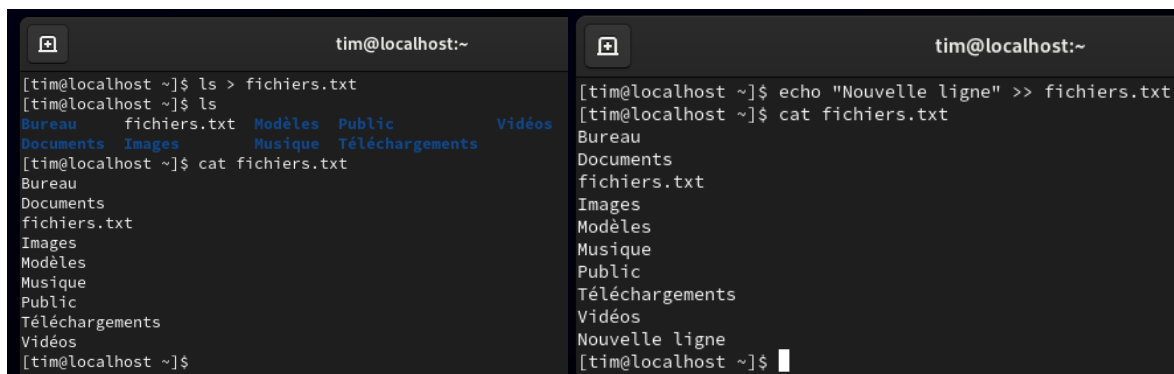
tim@localhost:~
[tim@localhost ~]$ ls -l /home
total 4
drwx-----. 14 tim tim 4096 30 sept. 12:30 tim
[tim@localhost ~]$

```

## 2. Utiliser la redirection d'entrée/sortie (>, >>, |, 2>, etc.)

La redirection permet d'envoyer la sortie d'une commande vers un fichier ou une autre commande.

- **Exemple** : Utiliser `ls > fichiers.txt` pour rediriger la liste des fichiers vers un fichier nommé `fichiers.txt`. Avec `echo "Nouvelle ligne" >> fichiers.txt`, ajoutez une nouvelle ligne à ce fichier.



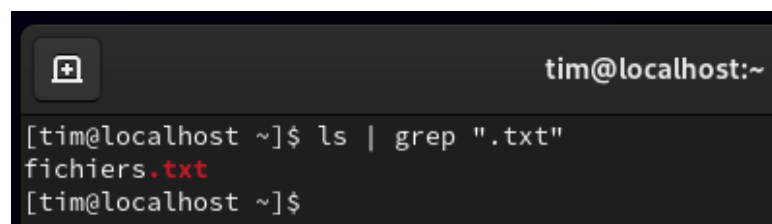
```

tim@localhost:~
[tim@localhost ~]$ ls > fichiers.txt
[tim@localhost ~]$ ls
Bureau  fichiers.txt  Modèles  Public  Vidéos
Documents  Images  Musique  Téléchargements
[tim@localhost ~]$ cat fichiers.txt
Bureau
Documents
fichiers.txt
Images
Modèles
Musique
Public
Téléchargements
Vidéos
[tim@localhost ~]$

tim@localhost:~
[tim@localhost ~]$ echo "Nouvelle ligne" >> fichiers.txt
[tim@localhost ~]$ cat fichiers.txt
Bureau
Documents
fichiers.txt
Images
Modèles
Musique
Public
Téléchargements
Vidéos
Nouvelle ligne
[tim@localhost ~]$

```

Vous pouvez aussi filtrer la sortie d'une commande avec un autre outil, par exemple `ls | grep ".txt"` qui affichera uniquement les fichiers `.txt`. La commande `grep` est utilisée pour rechercher des modèles de texte dans les fichiers ou la sortie d'autres commandes. Elle filtre les lignes qui contiennent une correspondance avec le texte spécifié.



```

tim@localhost:~
[tim@localhost ~]$ ls | grep ".txt"
fichiers.txt
[tim@localhost ~]$

```

Pour rediriger les messages d'erreur vers un fichier, utilisez `ls /dossier_inexistant 2> erreurs.txt`. Étant donné que le chemin `/dossier_inexistant` est erroné ou que le répertoire n'existe pas, la commande `ls` échouera, et le message d'erreur sera capturé dans le fichier `erreurs.txt` plutôt que d'apparaître dans le terminal.

```

tim@localhost:~
[tim@localhost ~]$ ls /dossier_inexistant 2> erreurs.txt
[tim@localhost ~]$ cat erreurs.txt
ls: impossible d'accéder à '/dossier_inexistant': Aucun fichier ou dossier de ce type
[tim@localhost ~]$

```

Opérateur	Description	Exemple
>	Redirige la sortie standard vers un fichier (écrase le contenu existant)	<code>ls &gt; fichiers.txt</code>
>>	Redirige la sortie standard vers un fichier (ajoute au contenu existant)	<code>echo 'nouveau texte' &gt;&gt; fichiers.txt</code>
	Envoie la sortie d'une commande comme entrée d'une autre commande (pipe)	<code>cat fichier.txt   grep 'mot'</code>
2>	Redirige les messages d'erreur (sortie d'erreur standard) vers un fichier	<code>ls /inexistant 2&gt; erreurs.txt</code>
<	Redirige l'entrée d'un fichier vers une commande	<code>sort &lt; liste.txt</code> <i>(La commande sort trie les lignes d'un fichier ou d'une sortie par ordre alphabétique ou numérique)</i>
&>	Redirige à la fois la sortie standard et les erreurs vers un fichier	<code>commande &amp;&gt; sortie_complète.txt</code>

### 3. Utiliser grep et les expressions régulières pour analyser du texte

`grep` permet de rechercher des motifs spécifiques dans un texte, et les expressions régulières permettent de définir ces motifs.

- **Exemple** : Recherchez toutes les lignes contenant "erreur" dans un fichier journal : `grep "erreur" journal.log`. Pour des recherches plus précises, utilisez `grep "^2024" journal.log` pour trouver toutes les lignes commençant par "2024".

```

tim@localhost:~
[tim@localhost ~]$ cat > journal.log
ceci est une ligne d'erreur
erreur une nouvelle fois
2021 était une drôle d'année
2024 on s'amuse plutôt bien
mais 2024 se termine bientôt déjà
on espère sans erreurs
[tim@localhost ~]$ grep "erreur" journal.log
ceci est une ligne d'erreur
erreur une nouvelle fois
on espère sans erreurs
[tim@localhost ~]$ grep "^2024" journal.log
2024 on s'amuse plutôt bien
[tim@localhost ~]$

```

*Le fichier journal.log a été créé et édité en amont*

#### 4. Accéder à des systèmes à distance via SSH

SSH (Secure Shell) est un protocole sécurisé pour se connecter à un autre ordinateur via un réseau. Il faut s'assurer que le serveur dispose du service OpenSSH installé et activé. Vous pouvez installer OpenSSH en utilisant la commande `sudo dnf install openssh-server`.

```

tim@localhost:~
[tim@localhost ~]$ sudo dnf install openssh-server
[sudo] Mot de passe de tim :
Mise à jour des référentiels de gestion des abonnements.
Dernière vérification de l'expiration des métadonnées effectuée il y a 5:50:21 l
e lun. 30 sept. 2024 12:24:15.
Le paquet openssh-server-8.7p1-38.el9_4.4.x86_64 est déjà installé.
Dépendances résolues.
Rien à faire.
Terminé !
[tim@localhost ~]$

```

*Dans mon cas, OpenSSH est déjà installé*

- **Exemple** : Connectez-vous à un serveur distant avec `ssh utilisateur@Nom_ou_IP`. Si vous devez spécifier un port particulier, utilisez `ssh -p 2222 utilisateur@Nom_ou_IP`. Vous pouvez effectuer cette connexion directement depuis l'invite de commande sur votre machine physique, ou en utilisant un émulateur de terminal comme PuTTY ou Tera Term pour vous connecter à distance via SSH. Ces outils sont particulièrement utiles si vous travaillez sur une machine Windows ou souhaitez une interface graphique simplifiée.

```

tim@localhost:~
[tim@localhost ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:43:df:3e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 2a01:cb11:bdd:8e00:a00:27ff:fe43:df3e/64 scope global dynamic noprefixroute
        valid_lft 86372sec preferred_lft 572sec
    inet6 fe80::a00:27ff:fe43:df3e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[tim@localhost ~]$

```

Pour se connecter en SSH, il faut connaître l'IP de notre hôte de destination

```

tim@device-129:~
Microsoft Windows [version 10.0.19045.4894]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Tim.DESKTOP-14KIBAC>ssh tim@192.168.1.50
tim@192.168.1.50's password:
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Mon Sep 30 18:18:42 2024 from 192.168.1.18
[tim@device-129 ~]$ ls
Bureau      erreurs.txt  Images      Modèles     Public      Téléchargements
Documents  fichiers.txt journal.log  Musique     sortie.txt  Vidéos
[tim@device-129 ~]$

```

Connexion SSH via ma machine physique Windows

## 5. Se connecter et changer d'utilisateur sur des cibles multi-utilisateurs

Dans un environnement multi-utilisateurs, il est souvent nécessaire de changer d'utilisateur pour effectuer des tâches administratives.

- **Exemple** : Pour passer à l'utilisateur root, utilisez `su -` et entrez le mot de passe, ou `sudo -i` si vous avez les permissions. Pour exécuter une commande spécifique en tant qu'administrateur, utilisez `sudo`, par exemple `sudo dnf update`.

```

root@device-129:~
[tim@localhost ~]$ su -
Mot de passe :
[root@device-129 ~]# exit
déconnexion
[tim@localhost ~]$ sudo -i
[root@device-129 ~]# dnf update
Mise à jour des référentiels de gestion des abonnements.
Dernière vérification de l'expiration des métadonnées effectuée il y a 5:58:55 l
e lun. 30 sept. 2024 12:24:15.
Dépendances résolues.
Rien à faire.
Terminé !
[root@device-129 ~]#

```

Commande	Description	Utilisation
<b>sudo</b>	Exécute une commande en tant qu'utilisateur root, sans changer de shell ni d'environnement	Pour exécuter une seule commande avec des privilèges root
<b>sudo -</b>	Accède au shell root en conservant partiellement l'environnement utilisateur actuel	Pour accéder au shell root tout en conservant certaines configurations utilisateur ( <i>répertoire personnel HOME</i> )
<b>sudo -i</b>	Accède au shell root avec l'environnement complet de l'utilisateur root, comme une connexion directe	Pour accéder à un shell complet root, chargé comme si vous vous connectiez directement

## 6. Archiver, compresser, décompresser des fichiers avec tar, star, gzip, bzip2

L'archivage permet de regrouper plusieurs fichiers en un seul, et la compression réduit leur taille.

- **Exemple** : Pour créer une archive compressée : `tar -czvf archive.tar.gz dossier/`. Cette commande crée une archive nommée `archive.tar.gz` à partir du répertoire `dossier`. Pour extraire cette archive, utilisez `tar -xzvf archive.tar.gz`.

```

tim@localhost:~
[tim@localhost ~]$ mkdir Test
[tim@localhost ~]$ touch Test/fichier1.txt Test/fichier2.txt
[tim@localhost ~]$ ls Test/
fichier1.txt fichier2.txt
[tim@localhost ~]$ tar -czvf archive.tar.gz Test/
Test/
Test/fichier1.txt
Test/fichier2.txt
[tim@localhost ~]$ ls
archive.tar.gz Documents Modèles Public Test
Bureau Images Musique Téléchargements Vidéos
[tim@localhost ~]$ rm -r Test/
[tim@localhost ~]$ tar -xzvf archive.tar.gz
Test/
Test/fichier1.txt
Test/fichier2.txt
[tim@localhost ~]$ ls
archive.tar.gz Documents Modèles Public Test
Bureau Images Musique Téléchargements Vidéos
[tim@localhost ~]$ █

```

Outil	Utilisation	Vitesse	Taux de compression	Usage	Extension
<b>tar</b>	Compression standard et rapide	Rapide	Modéré	Compression	.gz
<b>star</b>	Archivage avancé avec support des permissions complexes et des noms de fichiers longs	Rapide (meilleure performance que tar dans de nombreux cas)	Dépend de l'option de compression, mais principalement axé sur l'archivage	Archivage (peut être combiné avec des outils de compression comme gzip/bzip2)	.star (mais peut être combiné avec .gz ou .bz2)
<b>gzip</b>	Compression standard et rapide	Rapide	Modéré	Compression	.gz
<b>Bzip2</b>	Compression à haute efficacité mais plus lente	Plus lente	Élevé	Compression	.bz2

**Différence entre archivage et compression :** L'archivage consiste à regrouper plusieurs fichiers ou dossiers en un seul fichier sans réduire leur taille, facilitant ainsi leur stockage et transfert, tandis que la compression vise à réduire la taille des fichiers pour économiser de l'espace ou faciliter le transfert. Ensemble, ces deux processus permettent de mieux gérer les fichiers.

## 7. Créer et éditer des fichiers texte

Savoir manipuler les fichiers texte est fondamental pour gérer les scripts, la configuration, etc.

- **Exemple :** Utilisez `nano monfichier.txt` ou `vi monfichier.txt` pour créer ou modifier un fichier texte. Par exemple, ouvrez nano et tapez du texte, puis utilisez `Ctrl + X, Y`, puis Entrée pour enregistrer et quitter.

## 8. Créer, supprimer, copier et déplacer des fichiers et dossiers

Ces opérations sont essentielles pour organiser et gérer les fichiers du système.

- **Exemple :** Pour copier un fichier : `cp source.txt destination.txt`. Déplacer un fichier : `mv ancien_nom.txt nouveau_nom.txt`. Supprimer un fichier : `rm fichier.txt`. Créer un dossier : `mkdir nouveau_dossier`. Supprimer un dossier : `rm -r dossier_a_supprimer`.

```

tim@localhost:~
[tim@localhost ~]$ cp sortie.txt nouvelle_sortie.txt
[tim@localhost ~]$ ls
archive.tar.gz  erreurs.txt  Musique  sortie.txt
Bureau         Images      nouvelle_sortie.txt  Téléchargements
Documents      Modèles    Public    Vidéos
[tim@localhost ~]$ rm nouvelle_sortie.txt
[tim@localhost ~]$ ls
archive.tar.gz  Documents  Images  Musique  sortie.txt  Vidéos
Bureau         erreurs.txt  Modèles  Public  Téléchargements
[tim@localhost ~]$ mkdir nouveau_dossier
[tim@localhost ~]$ ls
archive.tar.gz  erreurs.txt  Musique  sortie.txt
Bureau         Images      nouveau_dossier  Téléchargements
Documents      Modèles    Public    Vidéos
[tim@localhost ~]$

```

## 9. Créer des liens matériels et logiciels

Les liens permettent de créer des raccourcis ou des références vers des fichiers.

- **Exemple** : Créez un lien symbolique avec `ln -s /chemin/vers/fichier lien_symbolique`. Un lien symbolique pointe vers un fichier ou un répertoire, tandis qu'un lien matériel (`ln fichier1 lien_hard`) est un deuxième nom pour le même fichier.

Vous avez un script `/usr/local/bin/script.sh` que vous souhaitez exécuter depuis votre dossier personnel sans retaper le chemin complet.

- **Lien symbolique**: `ln -s /usr/local/bin/script.sh ~/mon_script` permet d'exécuter le script via `~/mon_script` sans à avoir à retaper tout le chemin.

```

tim@localhost:~
[tim@localhost ~]$ ln -s Documents/journal.log ~/journal.log
[tim@localhost ~]$ echo "Nouvelle ligne" > ~/journal.log
[tim@localhost ~]$ cat ~/journal.log
Nouvelle ligne
[tim@localhost ~]$

```

- **Lien matériel**: `ln /usr/local/bin/script.sh ~/script_copie` crée un lien direct vers le script original, il fonctionnera même si le script original est déplacé à un autre endroit.

```

tim@localhost:~
[tim@localhost ~]$ ln Documents/journal.log ~/journal_copie
[tim@localhost ~]$ mv ~/journal.log Images/
[tim@localhost ~]$ cat ~/journal_copie
Nouvelle ligne
[tim@localhost ~]$ █

```

## 10. Gérer les permissions ugo/rwx

Les permissions contrôlent qui peut lire, écrire ou exécuter un fichier ou un répertoire.

- **Exemple** : Pour attribuer des permissions de lecture et d'exécution à tous les utilisateurs : `chmod 755 script.sh`. Les chiffres (7, 5, 5) représentent les permissions pour l'utilisateur (r pour read, w pour write, x pour execution), le groupe (r-x) et les autres (r-x).

```

tim@localhost:~
[tim@localhost ~]$ ls -l erreurs.txt
-rw-r--r--. 1 tim tim 0 30 sept. 18:03 erreurs.txt
[tim@localhost ~]$ chmod 755 erreurs.txt
[tim@localhost ~]$ ls -l erreurs.txt
-rwxr-xr-x. 1 tim tim 0 30 sept. 18:03 erreurs.txt
[tim@localhost ~]$

```

*Ls -l affiche une liste détaillée des fichiers et dossiers dans un répertoire, montrant des informations comme les permissions, le propriétaire, la taille, et la date de modification de chaque fichier ou dossier.*

Chiffre	Permission (ugo)	Signification
0	---	Aucune permission
1	--X	Exécution uniquement
2	-W-	Écriture uniquement
3	-WX	Écriture et exécution
4	r--	Lecture uniquement
5	r-X	Lecture et exécution
6	rw-	Lecture et écriture
7	rwx	Lecture, écriture et exécution

## 11. Consulter la documentation système (/usr/share/doc)

De nombreux programmes installés sur un système Linux incluent de la documentation.

- **Exemple** : Accédez à la documentation d'un programme spécifique : `cat /usr/share/doc/bash/README`. Vous pouvez également utiliser `man` pour accéder aux pages de manuel, par exemple `man ls` pour en savoir plus sur la commande `ls`.

```

tim@localhost:~
[tim@localhost ~]$ cat /usr/share/doc/bash/README
This directory contains the bash documentation.

FAQ           - a set of frequently-asked questions about Bash with answers
INTRO        - a short introduction to bash
article.ms   - an article I wrote about bash for The Linux Journal
bash.1       - the bash man page
builtins.1   - a man page that documents the builtins, extracted from bash.1
bashref.texi - the `bash reference manual'
bashref.info - the `bash reference manual' processed by `makeinfo'
readline.3   - the readline man page

The `.ps' files are postscript versions of the above. The `.html'
files are HTML versions of the man page and reference manual. The
`.0' files are formatted manual pages. The `.txt' versions are
ascii -- the output of `groff -Tascii'.

The rest of this file explains how to use the `builtins.1' man page.

For each command in the list of builtins create a file in man/man1 called:

${command}.1

eg.

```