

Systemes d'exploitation

Laboratoire 3 : Introduction aux gestions du traitement de texte et des utilisateurs



Introduction	3
Objectifs	3
Traitement de texte	3
Mode d'utilisation de Nano	5
Mode d'utilisation de Vim.....	6
Les redirections.....	7
Opérateurs principaux de redirection de sortie	8
Exemple de redirection de sortie.....	8
Exercices	9
Gestion des utilisateurs locaux	9
Exercices	13

Introduction

Dans ce laboratoire, nous allons aborder deux sujets différents mais néanmoins importants :

- Le traitement de texte.
- La gestion des utilisateurs.

Ils seront traités de façon introductive. Durant le quadrimestre 2, vous aurez l'occasion d'approfondir ces sujets.

Objectifs

- Créer les fichiers textes en CLI.
- Ecrire dans ces fichiers.
- Rediriger un contenu vers un fichier.
- Créer des utilisateurs et leur environnement de bureau.
- Comprendre les subtilités de « *root* ».

Traitement de texte

Qu'est-ce qu'un traitement de texte ? Un logiciel conçu pour créer, éditer et manipuler des documents textes. Par exemple, LibreOffice, LaTeX,... Mais il existe des commandes en CLI (ligne de commande) qui font la même chose.

Les commandes les plus populaires sont « *nano* » et « *vim* ». Il existe aussi *Emacs*, ...

Nano est un éditeur de texte utilisé exclusivement sous les OS Linux. Comme les logiciels cités ci-dessus, il permet d'éditer, de créer et de manipuler des fichiers. C'est un éditeur simple à prendre en main.

Vim est un éditeur de texte beaucoup plus puissant que nano. Il est principalement utilisé sous les OS Linux mais il est possible de l'installer sur d'autres OS. Son nom complet est Vi Improved signifiant une amélioration de Vi.

Caractéristique	Nano	Vim
Facilité d'utilisation	Plus convivial pour les débutants. Les commandes sont généralement plus simples à comprendre et à utiliser.	Peut sembler complexe pour les débutants en raison de sa large gamme de fonctionnalités et de ses commandes spécifiques.
Interface utilisateur	Interface utilisateur simple et intuitive avec des commandes affichées en bas de l'écran.	Interface utilisateur en mode texte avec des commandes clavier. Pas d'indication visuelle des commandes à l'écran (à moins d'être activée).
Fonctionnalités avancées	Limité en termes de fonctionnalités avancées. Idéal pour l'édition de texte simple.	Offre de nombreuses fonctionnalités avancées, notamment la coloration syntaxique, les macros, les plugins, etc.
Modes de fonctionnement	Mode de fonctionnement en édition directe sans modes distincts.	Fonctionne avec différents modes (normal, insertion, visualisation) pour effectuer différentes tâches.
Personnalisation	Moins de possibilités de personnalisation par rapport à Vim.	Hautement personnalisable avec des fichiers de configuration et des plugins.
Apprentissage	Plus rapide à prendre en main pour les débutants.	Peut nécessiter une courbe d'apprentissage plus longue en raison de ses fonctionnalités avancées.
Popularité	Souvent préféré par les débutants et ceux qui recherchent une simplicité d'utilisation.	Préférent par les utilisateurs avancés et les développeurs pour sa puissance et sa flexibilité.
Commandes fréquemment utilisées	Les commandes courantes sont généralement affichées à l'écran, ce qui facilite leur utilisation.	Les commandes sont basées sur des raccourcis clavier, ce qui peut nécessiter de les mémoriser.
Utilisation dans les scripts	Moins couramment utilisé dans les scripts en raison de sa simplicité.	Souvent utilisé dans les scripts en raison de sa puissance de traitement de texte en mode batch.

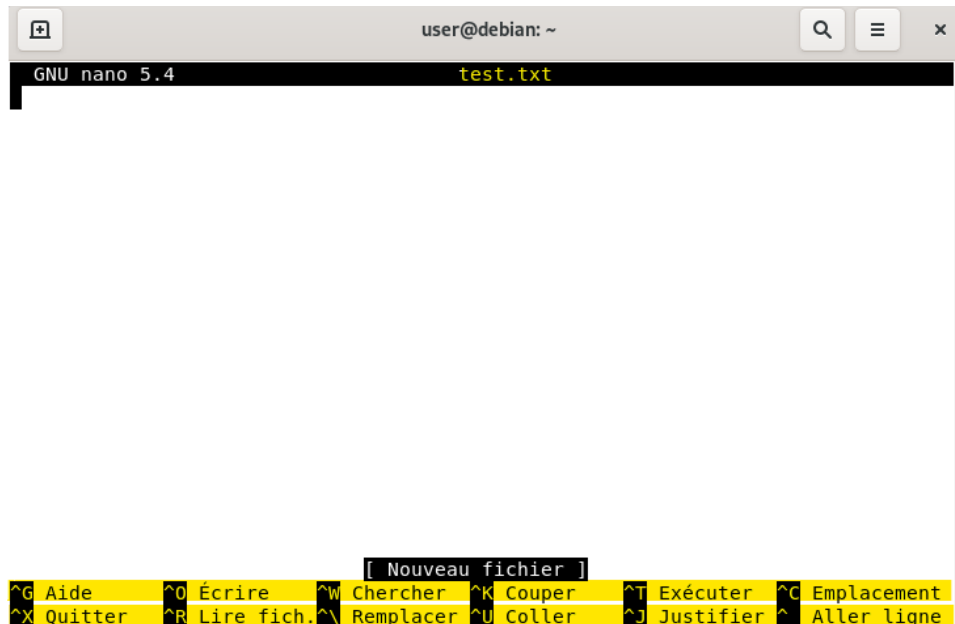
Mode d'utilisation de Nano

Pour utiliser la commande, il n'y a pas de condition. Cela veut dire que l'on peut créer un fichier texte en exécutant la commande ou soit utiliser un fichier déjà créé avec la commande *touch*.

- Pour ouvrir/créer un fichier : il suffit de taper la commande suivi le nom du fichier que l'on veut éditer. Par exemple, pour éditer un fichier nommé **test.txt**, on fera :

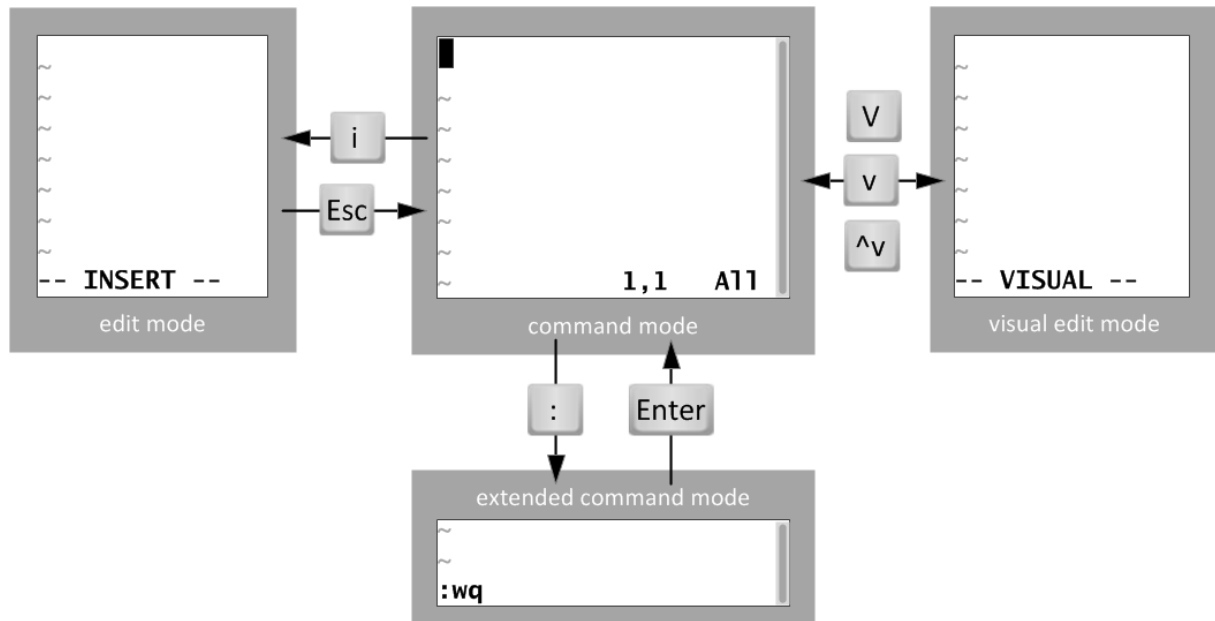
```
user@debian:~$ nano test.txt
```

- Interface utilisateur : une fois le fichier ouvert, l'interface *nano* s'ouvre. Il y a le nom du fichier en haut et son chemin. En bas, il y a un menu affichant les commandes courantes.



- Édition de texte : Vous pouvez commencer à éditer le texte en utilisant les touches de votre clavier. Pour ajouter ou modifier du texte, utilisez simplement votre clavier pour saisir ou supprimer du contenu.
- Sauvegarde du fichier : Pour sauvegarder vos modifications, appuyez sur "Ctrl + O". Vous serez invité à confirmer le nom du fichier. Appuyez sur "Entrée" pour enregistrer.
- Fermeture du fichier : Pour quitter *nano* tout en conservant les modifications, appuyez sur "Ctrl + X". Si vous avez apporté des modifications non enregistrées, *nano* vous demandera de les enregistrer avant de quitter.

Mode d'utilisation de Vim



Lorsqu'il s'ouvre pour la première fois, *Vim* démarre en mode commande, utilisé pour la navigation, les couper-coller et autres manipulations de texte. Les autres modes et des fonctionnalités d'édition spécifiques sont accessibles à l'aide d'une seule frappe sur le clavier :

- Une pression sur **i** ouvre le mode d'insertion, dans lequel tout le texte saisi s'ajoute au contenu du fichier. Appuyez sur **Échap** pour retourner au mode de commande.
- Une pression sur **v** ouvre le mode visuel, dans lequel il est possible de sélectionner des caractères multiples pour manipuler le texte. Utilisez **V** pour sélectionner plusieurs lignes, et **Ctrl+v** pour sélectionner un bloc. Le même raccourci est utilisé pour passer en mode visuel (**v**, **V** ou **Ctrl+v**) et pour en sortir.
- La touche **:** ouvre le mode de commande étendu pour effectuer des tâches telles que l'écriture et l'enregistrement d'un fichier ou pour quitter l'éditeur Vim.
- La combinaison de **:w** pour écrire (enregistrer) le fichier et rester en mode de commande afin de continuer l'édition.
- La combinaison **:wq** pour écrire (enregistrer) le fichier et quitter l'éditeur Vim.
- La combinaison **:q!** pour quitter l'éditeur Vim en annulant toutes les modifications apportées au fichier depuis la dernière écriture.

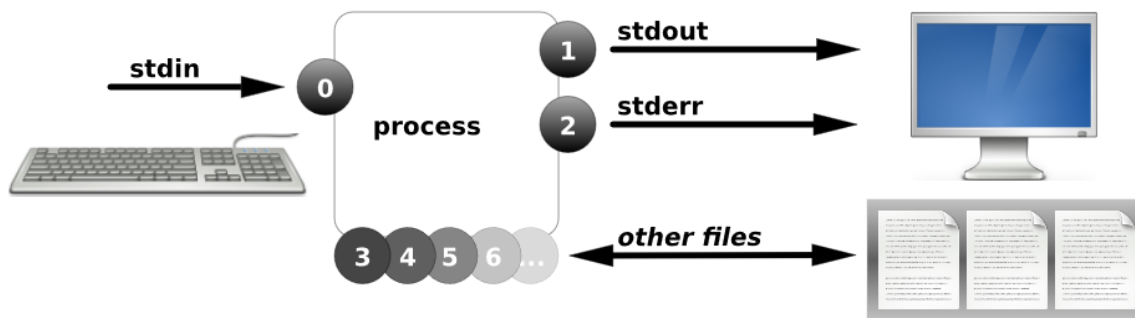
Vim offre des combinaisons de touches efficaces et coordonnées pour effectuer des tâches d'édition évoluées. Alors qu'il est considéré comme très utile une fois maîtrisé, l'éditeur Vim peut désorienter les nouveaux utilisateurs. Le flux de travail suivant présente les touches de base que chaque utilisateur de Vim doit connaître pour accomplir n'importe quelle tâche d'édition.

Les redirections

Un programme d'exécution, ou processus, doit lire l'entrée depuis un emplacement et écrire la sortie sur l'écran ou dans des fichiers. Une commande exécutée depuis l'invite de shell lit habituellement son entrée depuis le clavier et envoie sa sortie vers sa fenêtre de terminal.

Un processus utilise des canaux numérotés appelés descripteurs de fichiers pour obtenir une entrée et envoyer une sortie. Tous les processus disposent d'au moins trois descripteurs de fichiers pour commencer. L'Entrée standard (canal 0) lit l'entrée depuis le clavier. La Sortie standard (canal 1) envoie la sortie normale vers le terminal.

L'Erreur standard (canal 2) envoie les messages d'erreur vers le terminal. Si un programme ouvre des connexions séparées avec d'autres fichiers, il peut utiliser des descripteurs de fichiers portant des numéros plus élevés.

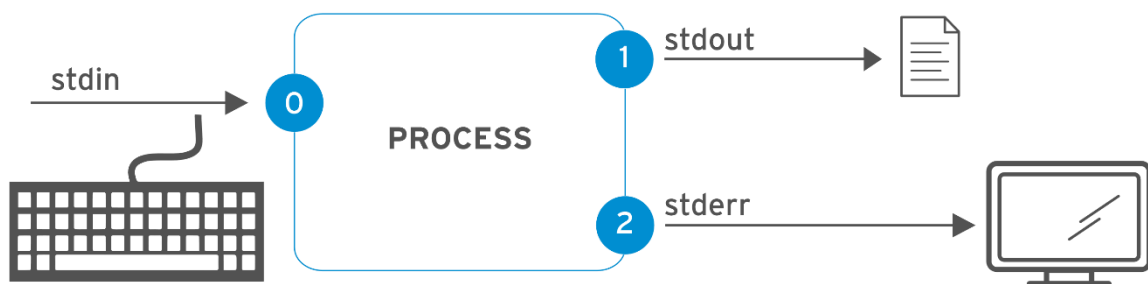


Numéro	Nom du canal	Description	Connexion par défaut	Utilisation
0	stdin	Entrée standard	Clavier	lecture seule
1	stdout	Sortie standard	Terminal	écriture seule
2	stderr	Erreur standard	Terminal	écriture seule
3+	<i>nom de fichier</i>	Autres fichiers	<i>aucune</i>	lecture et/ou écriture

La redirection d'E/S remplace les destinations de canal par défaut par les noms de fichiers qui représentent des fichiers de sortie ou des périphériques. En utilisant la redirection, les sorties de processus et les messages d'erreur normalement envoyés vers

la fenêtre de terminal peuvent être capturés sous forme de contenu de fichier, envoyés vers un périphérique ou supprimés.

La redirection de **stdout** empêche l'affichage de la sortie du processus sur le terminal. Comme indiqué dans le tableau suivant, la redirection de **stdout** **uniquement** n'empêche pas l'affichage des messages d'erreur **stderr** sur le terminal. Si le fichier n'existe pas, il sera créé. Si le fichier existe et que la redirection ne s'ajoute pas au fichier, le contenu du fichier sera écrasé. Le fichier spécial **/dev/null** élimine discrètement la sortie de canal redirigée vers ce fichier et est toujours un fichier vide.



Opérateurs principaux de redirection de sortie

Commande > **fichier**

Redirection du résultat de la commande (stdout) dans un fichier tout en écrasant le contenu du fichier (si contenu existant)

Commande >> **fichier**

Redirection du résultat de la commande (stdout) dans un fichier en l'ajoutant au reste du fichier (si contenu existant)

Exemple de redirection de sortie

- Enregistrer un horodatage pour consultation ultérieure.

```
user@debian:~$ date > date_check_later
user@debian:~$ cat date_check_later
lun 09 oct 2023 14:12:28 CEST
user@debian:~$
```

- Copier les 100 dernières lignes d'un fichier log vers un autre fichier.


```
user@debian:/var/log$ tail -n 100 auth.log > /home/user/log.txt
```

Exercices

1. Créez un fichier **vim_exercice1** avec la commande *vim*. Annotez dedans « *Mais, vous savez, moi je ne crois pas qu'il y ait de bonne ou de mauvaise situation. – Otis* »

Indice : *vim nom_fichier*

2. Créez un fichier *nano_exercice2* avec la commande *nano* et éditez-le en inscrivant vos nom, prénom, groupe et section l'un en dessous des autres. Sauvegardez vos modifications.
3. Listez les 2 premières lignes de votre fichier et redirigez-les dans un fichier nommé **result_exercice3**.
 - a. Faites un *cat result_exercice3* pour afficher le contenu du fichier.
4. Listez les répertoires de la racine et redirigez cette liste dans un fichier **liste_repertoire_racine4**.
 - a. La commande *ls* pour vous aider.
5. Listez les disques sur votre VM et redirigez cela dans un fichier **disk_ex5**.
 - a. La commande *su* - pour vous connecter en root.
 - b. La commande *fdisk -l* peut vous aider.

Gestion des utilisateurs locaux

Qu'est-ce qu'un utilisateur ? Un individu pouvant accéder au système, exécuter des programmes, accéder à des fichiers et des ressources. Il peut aussi effectuer diverses actions sur l'OS.

Il existe plusieurs types d'utilisateurs :

- *L'utilisateur standard* : il peut exécuter des tâches et des commandes communes. Ils ont des droits limités et ne peuvent pas exécuter des opérations critiques (modification réseau, accès aux fichiers système, ...)
- *Le super utilisateur* : il est l'administrateur. Il s'appelle, plus communément, root. Ce dernier a l'accès complet au système et peut effectuer n'importe quelle opération. Il faut faire preuve de prudence en utilisant ce compte.
- *Les utilisateurs systèmes* : ce sont des utilisateurs liés à certains services. Ils sont utilisés pour exécuter des processus spécifiques associés à ces applications ou services. Les utilisateurs système ont souvent des privilèges limités et ne sont généralement pas destinés à être utilisés pour la connexion au système.

La commande *id* sert à afficher des informations sur l'utilisateur actuellement connecté. Il est également possible de demander des informations élémentaires sur un autre utilisateur, en passant son nom d'utilisateur comme premier argument de la commande *id*.

```
user@debian:~$ id
uid=1000(user) gid=1000(user) groupes=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev),113(bluetooth),118(lpadmin),121(scanner)
```

Pour afficher l'utilisateur associé à un fichier ou à un répertoire, utilisez la commande *ls -l*. La troisième colonne indique le nom de l'utilisateur :

```
user@debian:~$ ls -l
total 56
drwxr-xr-x 2 user user 4096 9 jun 12:32 Bureau
-rw-r--r-- 1 user user 30 9 oct 14:12 date_check_later
drwxr-xr-x 2 user user 4096 9 jun 12:32 Documents
drwxr-xr-x 2 user user 4096 9 jun 12:32 Images
drwxr-xr-x 2 user user 4096 3 oct 09:41 Labo1
drwxr-xr-x 2 user user 4096 3 oct 09:41 Labo2
-rw-r--r-- 1 root root 11802 9 oct 14:14 log.txt
drwxr-xr-x 2 user user 4096 9 jun 12:32 Modèles
drwxr-xr-x 2 user user 4096 9 jun 12:32 Musique
drwxr-xr-x 2 user user 4096 9 jun 12:32 Public
drwxr-xr-x 2 user user 4096 9 jun 12:32 Téléchargements
drwxr-xr-x 2 user user 4096 9 jun 12:32 Vidéos
```

Ps au

Pour afficher les informations relatives au processus, utilisez la commande *ps*. Par défaut, seuls les processus du shell courant sont affichés. Ajoutez l'option *a* pour afficher tous les processus liés à un terminal. Pour afficher l'utilisateur associé à un processus, ajoutez l'option *u*. La première colonne indique le nom de l'utilisateur :

```
user@debian:~$ ps au
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
user          998  0.0  0.1 159076 5932 tty2    Ssl+  13:54   0:00 /usr/libexec/gdm-way
user         1005  0.0  0.4 298376 16128 tty2    Sl+   13:54   0:00 /usr/libexec/gnome-s
user         1679  0.0  0.1   8600  5308 pts/0    Ss   13:56   0:00 bash
root         1872  0.0  0.1  10084  4844 pts/0    S    14:13   0:00 su -
root         1873  0.0  0.1   8756  5272 pts/0    S    14:13   0:00 -bash
root         1957  0.0  0.1  10064  4496 pts/0    S    14:14   0:00 su user
user         1958  0.0  0.1   7920  4612 pts/0    S    14:14   0:00 bash
root         2056  0.0  0.1  10084  4844 pts/0    S    14:39   0:00 su -
root         2057  0.0  0.1   8448  5276 pts/0    S    14:39   0:00 -bash
root         2074  0.0  0.1  10092  4792 pts/0    S    14:49   0:00 su -
root         2075  0.0  0.1   7920  4664 pts/0    S    14:49   0:00 -bash
root         2134  0.0  0.1  10064  4548 pts/0    S    14:49   0:00 su user
user         2135  0.0  0.1   7920  4616 pts/0    S    14:49   0:00 bash
root         2139  0.0  0.1  10096  4428 pts/0    S    14:49   0:00 su toto
toto         2140  0.0  0.1   7924  4648 pts/0    S    14:49   0:00 bash
root         2255  0.0  0.1  10084  4976 pts/0    S    15:32   0:00 su -
root         2256  0.0  0.1   8440  5236 pts/0    S    15:32   0:00 -bash
root         2325  0.0  0.1  10064  4444 pts/0    S    15:33   0:00 su user
user         2326  0.0  0.1   7920  4680 pts/0    S    15:33   0:00 bash
user         2485  0.0  0.0   9848  3300 pts/0    R+   15:57   0:00 ps au
```

Le résultat des commandes précédentes affiche les utilisateurs par nom, mais en interne, le système d'exploitation les repère à l'aide de leur *numéro UID*. La mise en correspondance des noms et des numéros est définie dans les bases de données des informations sur les comptes.

Par défaut, les systèmes utilisent un simple « fichier plat », le fichier */etc/passwd*, pour stocker les informations concernant les utilisateurs locaux. Le format de */etc/passwd* (sept champs séparés par des deux-points) se présente comme suit :

```
user:x:1000:1000:user,,,:/home/user:/bin/bash
1 2 3 4 5 6 7
root@debian:~#
```

- 1 *username* est la correspondance entre un UID et un nom d'utilisateur, pour faciliter la lecture par les utilisateurs humains.
- 2 *password* est, historiquement, l'emplacement où les mots de passe étaient conservés sous forme chiffrée. De nos jours, ils sont stockés dans un fichier distinct, appelé */etc/shadow*.
- 3 *UID* est un identifiant d'utilisateur, un numéro qui identifie l'utilisateur au niveau le plus fondamental.
- 4 *GID* est le numéro d'identification du groupe principal de l'utilisateur. Les groupes seront abordés dans un instant.
- 5 Le champ *GECOS* correspond à du texte arbitraire, et contient généralement le nom réel de l'utilisateur.
- 6 */home/dir* est l'emplacement des données et fichiers de configuration personnels de l'utilisateur.
- 7 *shell* est un programme exécuté lors de la connexion de l'utilisateur. Pour un utilisateur standard, il s'agit normalement du programme qui fournit l'invite de ligne de commande de l'utilisateur.

Pour se connecter à un utilisateur (autrement qu'à la connexion), il y a la commande *su*. Cela signifie Substitute User ou SuperUser. Cela permet de changer d'utilisateur en CLI.

Par exemple : je veux me connecter à l'utilisateur toto, j'utilise la commande suivante : *su toto*

```
user@debian:~$ su toto
Mot de passe :
toto@debian:/home/user$
```

Bien évidemment, l'utilisateur peut se connecter à l'utilisateur root s'il en a les droits. Mais attention, exécuter la commande *su root* permettra à l'utilisateur courant de se connecter avec root mais il ne pourra pas utiliser certaines commandes.

Pourquoi ? Faites la commande *pwd*. Votre root est dans le répertoire de l'utilisateur précédent.

```
user@debian:~$ su root
Mot de passe :
root@debian:/home/user# pwd
/home/user
```

Faites la commande `su -`, vous serez toujours avec l'utilisateur root mais en refaisant la commande `pwd`, où êtes-vous à présent ?

```
user@debian:~$ su -  
Mot de passe :  
root@debian:~# pwd  
/root
```

C'est parce qu'avec le tiret, les variables d'environnement de l'utilisateur courant sont remplacées par celle du compte root. Cela peut être utile pour garantir que vous disposez d'un environnement propre et cohérent lorsque vous travaillez en tant que super utilisateur, en particulier si vous avez besoin d'accéder à des variables d'environnement spécifiques au compte root.

En résumé, la différence entre `su root` et `su -` (ou `su - root`) réside dans la gestion des variables d'environnement lors du passage au compte root. Utiliser `su root` conserve l'environnement de l'utilisateur d'origine, tandis que `su -` charge l'environnement du compte root. Le choix dépendra de vos besoins spécifiques en matière d'environnement lors de l'exécution de commandes en tant que root.

Il existe la commande `sudo` qui permet d'exécuter des commandes avec les privilèges du super user tout en restant avec votre utilisateur courant.

Pour créer des utilisateurs, il y a deux possibilités de commandes : `adduser` et `useradd`. Respectivement, l'une est plus conviviale, automatisée et interactive et l'autre est plus orientée CLI. L'une pose des questions pour configurer diverses options tandis que l'autre a besoin que l'on donne des options pour configurer correctement un utilisateur.

```
user@debian:~$ sudo adduser testmanip  
[sudo] Mot de passe de user :  
Ajout de l'utilisateur « testmanip » ...  
Ajout du nouveau groupe « testmanip » (1003) ...  
Ajout du nouvel utilisateur « testmanip » (1003) avec le groupe « testmanip » ...  
Création du répertoire personnel « /home/testmanip »...  
Copie des fichiers depuis « /etc/skel »...  
Nouveau mot de passe :  
Retapez le nouveau mot de passe :  
passwd: password updated successfully  
Changing the user information for testmanip  
Enter the new value, or press ENTER for the default  
  Full Name []: TestManip  
    Room Number []: 05  
    Work Phone []:  
    Home Phone []:  
    Other []: Ceci est un Test  
Cette information est-elle correcte ? [0/n]o
```

Penchons-nous sur la seconde `useradd` qui demande beaucoup plus d'options que le futur nom d'utilisateur.

```
user@debian:~$ sudo useradd testmanip2 -u 1082 -m -d /home/testmanip2 -s /bin/bash -c "  
Test"
```

L'option `-c` ou `--comment` permet d'ajouter un commentaire (gecos) pour l'utilisateur.

L'option `-m` ou `--create-home` permet de créer le répertoire "home" de l'utilisateur.

L'option **-d** ou **--home** permet de spécifier le répertoire personnel de l'utilisateur (préférence dans /home)

L'option **-u** ou **--uid** permet de spécifier un UID (identifiant utilisateur) personnalisé.

L'option **-s** ou **--shell** définit le shell par défaut. La bonne pratique est de définir le shell bash (/bin/bash).

La commande *passwd* vue lors du laboratoire précédent permettra de configurer le mot de passe d'un utilisateur.

```
user@debian:~$ sudo passwd testmanip
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
```

Exercices

Pour utiliser la commande *sudo*, il est possible qu'elle ne soit pas installée. Connectez-vous en *root* via la commande *su -*. Faites un *apt-get install sudo* pour installer le paquet.

Il faut que votre utilisateur de base soit autorisé à utiliser cette commande. Pour ce faire, vous allez éditer le fichier */etc/sudoers*.

Dans le paragraphe : **Allow members of group sudo to execute any command**, ajoutez ceci (et sauvegardez après l'édition) :

User_Name ALL=(ALL:ALL) ALL

Redémarrez votre machine avec la commande *reboot*.

6. Créez un utilisateur **toto** avec *adduser*. Le mot de passe sera **tttttt**.
7. Créez un utilisateur **coco** avec *useradd* tout en restant connecté avec votre utilisateur de base. Son Uid doit être de 1050. Il doit avoir un répertoire personnel dans /home et avoir le shell par défaut. Mettez-lui un mot de passe simple.
8. Connectez-vous avec **coco** et faites en sorte qu'il puisse aussi utiliser la commande *sudo*.
9. Affichez les informations de l'utilisateur **coco** et redirigez-les dans un fichier **exercice10**.
 - a. Vous devez utiliser l'utilisateur **coco**.
10. Tout en restant avec cet utilisateur, vous allez rediriger la toute dernière ligne du fichier */etc/passwd* dans un autre nommé **info_coco**.
 - a. Faites la commande *cat* suivi de **info_coco** pour afficher le contenu.