**Laboratoire systèmes**

Laboratoire 4 :

Linux : Introduction à la ligne de commande + Gestion des utilisateurs locaux



# Accès à la ligne de commande

C'est un laboratoire primordial pour la compréhension des manipulations suivantes.

1. **Objectifs**

* Découverte de la ligne de commande (CLI – command line) sous Linux
* Se familiariser avec certains raccourcis sous Linux

1. **Préparation**

Afin de disposer de suffisamment de temps pour réaliser l'ensemble de la manipulation, vous devez lire la manipulation au préalable.

Les fichiers nécessaires à la manipulation sont les images disques Linux Debian. Vous pouvez les télécharger à ces adresses :

* Debian : https://www.debian.org/distrib/netinst → Choisir AMD64 pour petits CD ou clefs USB

Idéalement vous devriez déjà avoir votre machine virtuelle Debian avec interface graphique prête.

1. **Evaluation**

Durant ce laboratoire vous serez évalués au travers de différentes questions/exercices posés dans ce document. Il sera nécessaire de noter vos réponses dans un document à transmettre au professeur à la fin de la séance par mail ou via clé USB.

**N’oubliez pas de remettre ce document, il sera essentiel pour votre évaluation.**

1. **Répartition des points**

Exercice 1 : chaque question vaut pour 1 point

Exercice 2 : chaque question vaut pour 1 point

Exercice 3 : chaque question vaut pour 2 points

**Appelez le professeur après chaque section d’exercices.**

# Le shell bash

Une ligne de commande est une interface en mode texte qui permet de saisir des instructions dans un système informatique. La ligne de commande de Linux est fournie par un programme appelé *Shell* (interpréteur de commandes). Au cours de la longue histoire des systèmes UNIX et apparentés, de nombreux *Shell* ont été développés. Le *Shell* par défaut des utilisateurs des distributions Debian est le GNU Bourne-Again *Shell* (bash). Bash est une version améliorée de l'un des Shell les plus populaires sur les systèmes UNIX, le shell Bourne (sh).

Lorsqu'un *Shell* est utilisé de manière interactive, il affiche une chaîne lorsqu'il attend une commande de la part de l'utilisateur. On l'appelle l'invite du *shell* (prompt). Lorsqu'un utilisateur standard lance un *shell*, l'invite par défaut se termine par le caractère $.



Le caractère $ est remplacé par un # si le *shell* est exécuté par le super-utilisateur, root.

Le super utilisateur (root) sous Linux est un compte administratif doté de privilèges complets, lui permettant de contrôler et de modifier l'ensemble du système. Son utilisation nécessite la plus grande prudence pour éviter des altérations accidentelles ou malveillantes du système.



Le bash peut s'avérer puissant pour l'exécution de commandes. Le shell bash fournit un langage de script qui permet l'automatisation des tâches.

Le shell bash offre des fonctionnalités supplémentaires qui peuvent simplifier ou rendre possibles des opérations difficiles à accomplir efficacement avec des outils graphiques.

Le shell bash repose sur le même concept que l'interpréteur de commande que l'on trouve dans les dernières versions de cmd.exe de Microsoft Windows, bien que le shell bash offre un langage de script plus évolué. Il s'apparente également au Windows PowerShell de Windows.

# Consoles virtuelles

Les utilisateurs accèdent au shell bash via un terminal. Un terminal est doté d'un clavier pour la saisie des commandes par l'utilisateur, et d'un écran pour les résultats. Sur les installations en mode texte, il peut s'agir de la console physique, le clavier et l'écran de la machine qui exécute Linux. L'accès au terminal peut être également configuré par l'intermédiaire de ports série.

On peut aussi accéder à un shell à partir d'une console virtuelle. La console physique d'une machine qui exécute Linux prend en charge plusieurs consoles virtuelles, qui se comportent comme des terminaux distincts. Chaque console virtuelle prend en charge une session connectée indépendante.

Si l'environnement graphique est disponible, il s'exécutera sur la première console virtuelle dans Debian.

# Concepts élémentaires du shell

Les commandes entrées sur l'invite shell comportent trois parties principales :

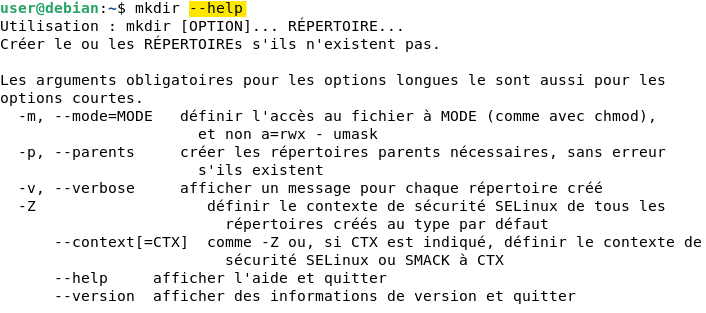
* La commande à exécuter
* Les options qui ajustent le comportement de la commande
* Les arguments, qui sont, le plus souvent, les cibles de la commande



Par exemple, la ligne de commande **mkdir -v Labo2** comprend une commande (mkdir), une option (-v) et un argument (labo 2).

La commande est le nom du programme à exécuter. Elle peut être suivie d'une ou plusieurs options, qui affinent le comportement de la commande ou ce qu'elle va faire. Les options commencent normalement par un ou deux tirets (-a ou --all, par exemple) pour les distinguer des arguments. Les commandes peuvent également être suivies d'un ou plusieurs arguments qui indiquent souvent une cible sur laquelle la commande doit opérer.

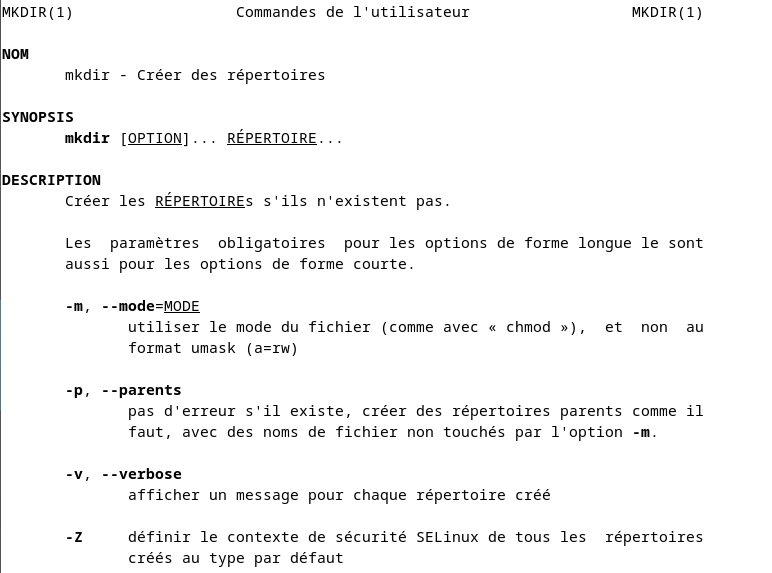
Pour utiliser une commande efficacement, l'utilisateur doit connaître les options et les arguments qu'elle accepte, et dans quel ordre elle les attend (la syntaxe de la commande). La plupart des commandes ont une option --help. Elle provoque l'affichage par la commande d'une description de sa fonction, une « instruction d'utilisation » qui décrit la syntaxe de la commande, et une liste des options qu'elle accepte et de leur fonction.



Quand l'utilisateur a fini d'utiliser le shell et qu'il souhaite le quitter, il peut mettre fin à la session de plusieurs manières. La commande **exit** met fin à la session shell en cours.

Une alternative pour demander de l’aider est d’utiliser la commande « **man** » suivi du nom de la commande que vous souhaitez utiliser. « Man » est défini ici pour le manuel pour le quitter il suffit d’appuyer sur la touche « q ».





# L'environnement de bureau GNOME

L'environnement de bureau est l'interface graphique de l'utilisateur d'un système Linux. L'environnement de bureau par défaut dans la distribution Debian est fourni par GNOME 3. Il fournit un bureau intégré aux utilisateurs, et une plate-forme de développement unifiée basée sur l'environnement graphique fourni par le système X Window.

Le Shell GNOME fournit les fonctions de base de l'interface utilisateur de l'environnement de bureau GNOME. L'application gnome-shell est hautement personnalisable.

Par défaut, les utilisateurs utilisent pour le gnome-shell le thème « GNOME». Parmi les autres options disponibles, le thème « GNOME classique », est similaire à l'environnement de bureau GNOME 2.utilisé en amont par le projet GNOME. Il est possible de sélectionner l'un ou l'autre thème de manière persistante à la connexion en sélectionnant l'icône en forme d'engrenage (sur la droite), lors de la saisie du mot de passe de l'utilisateur.

À la première connexion d'un utilisateur, un programme de configuration initiale se lance pour l'aider à configurer les réglages de base du compte. Il est possible de lancer rapidement l'Aide GNOME en cliquant sur l’icône « bouée » du lanceur ou en sélectionnant Activités (si le lanceur n’est pas visible). Ce « lanceur d'applications » est également appelé « dash ».

Une image contenant texte, capture d’écran

Description générée automatiquement

# Lancement d'un terminal

Pour lancer une invite de shell dans GNOME, démarrez une application de terminal graphique telle que GNOME Terminal. Il y a plusieurs manières de le faire. Voici les trois méthodes les plus couramment utilisées :

Sélectionner Activités -> dans la zone recherche taper les premières lettres de terminal -> Lorsque l’icône de application apparait, cliquer dessus pour la lancer. Vous pouvez aussi la faire glisser vers le « dash » pour le personnaliser.

Sélectionner Activités -> cliquer sur le bouton grille du lanceur et choisir terminal dans la liste des applications disponibles.

Dès l'ouverture d'une fenêtre de terminal s'affiche une invite de shell au nom de l'utilisateur qui a lancé le programme de terminal graphique. L'invite de shell et la barre de titre de la fenêtre de terminal indiqueront le nom de l'utilisateur, le nom de l'hôte et le répertoire de travail actuels.

Une image contenant texte, capture d’écran

Description générée automatiquement

# Exercices 1

Dans cet atelier, vous allez vous connecter par l'intermédiaire du gestionnaire d'affichage graphique en tant qu'utilisateur standard pour vous familiariser avec l'environnement de bureau GNOME.*.*

Connectez-vous avec votre utilisateur en utilisant votre mot de passe.

Cliquez sur Se connecter après avoir saisi le mot de passe.

1. Remplacez votre mot de passe de votre utilisateur par Tigrou008=.

L'approche la plus simple consiste à ouvrir le Terminal GNOME et à utiliser la commande **passwd** à l'invite du shell.

Ouvrez un terminal.

1. Dans la fenêtre de terminal qui apparaît, tapez **passwd** à l'invite du shell. Suivez les instructions fournies par le programme pour remplacer votre mot de passe de votre utilisateur par Tigrou008=.

Déconnectez-vous.

Fermer la session de user pour retourner à l’écran de login.

Reconnectez-vous en tant que votre utilisateur avec le nouveau mot de passe Tigrou008=.

Dans un terminal, créez un répertoire et un fichier dans ce répertoire.

1. Créez le répertoire Exercice3 avec la commande **mkdir**.
2. Créez un fichier File4 avec la commande **touch**.
3. Affichez l’aide de la commande **date**.
4. Effectuez la commande **ls** et regardez ce qu’elle vous renvoie en sortie**.**

# Exécution de commandes à l'aide du shell Bash

Utiliser le complément par tabulation, l'historique des commandes et les fonctions de modification des lignes de commande du shell bash pour exécuter des commandes plus efficacement.

# Syntaxe des commandes de base

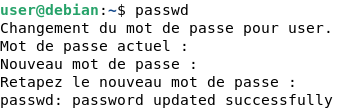
*Rappel*: Le GNU Bourne-Again Shell (**bash**) est un programme qui interprète les commandes saisies par l'utilisateur. Chaque chaîne tapée dans le shell peut comporter trois parties au maximum : la commande, les options (qui commencent par - ou --) et les arguments. Chaque mot tapé dans le shell est séparé des autres par des espaces. Les commandes correspondent au nom des programmes installés sur le système. Chaque commande possède ses propres options et arguments.

La touche **Entrée** est enfoncée lorsque l'utilisateur est prêt à exécuter la commande. Chaque commande est tapée sur une ligne distincte, et la sortie de chaque commande s'affiche avant que le shell n’affiche une nouvelle invite. Si l'utilisateur souhaite taper plusieurs commandes sur une seule ligne, il peut utiliser un point-virgule **;** pour séparer les commandes. Le point-virgule fait partie de la classe des caractères nommés métacaractères, qui ont une signification spéciale pour **bash**.

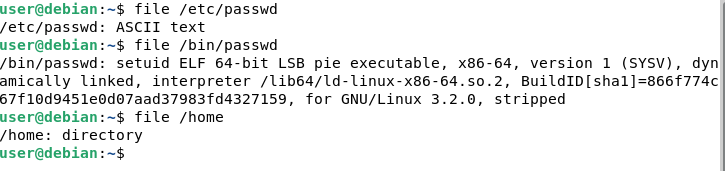
# Exemples de commandes simples

La commande **passwd** change le mot de passe de l'utilisateur courant. Le mot de passe original du compte doit être spécifié avant que le changement soit autorisé. Par défaut, **passwd** est configuré pour exiger un mot de passe fort, composé de minuscules, de majuscules, de nombres et de symboles et qui n'est pas basé sur un mot du dictionnaire. Le super-utilisateur peut utiliser la commande **passwd** pour changer le mot de passe des autres utilisateurs.

La saisie des mots de passe est transparente.

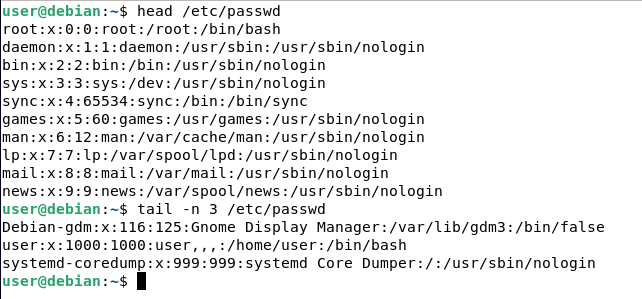


Linux ne nécessite pas d'extensions de nom de fichier pour classifier les fichiers par type. La commande **file** analyse le début du contenu d'un fichier et affiche son type. Les fichiers à classer sont transmis à la commande en tant qu'arguments.



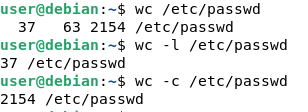
Les commandes **head** et **tail** affichent le début et la fin d'un fichier, respectivement.

Par défaut, ces commandes affichent 10 lignes, mais elles disposent toutes les deux d'une option **-n**, qui permet de spécifier un nombre de lignes différent. Le fichier à afficher est transmis à ces commandes en tant qu'argument.



La commande **wc** compte les lignes, mots et caractères d'un fichier.

Elle accepte les options **-l**, **-w** ou **-c** pour n'afficher respectivement que le nombre de lignes, mots ou caractères.



# Complément par tabulation

La saisie semi-automatique par tabulation vous permet de compléter rapidement les commandes et les noms de fichiers une fois que vous avez entré le nombre de caractères qui réduit les possibilités à une seule. Si les caractères saisis ne sont pas uniques, appuyez sur la touche **Tab** deux fois pour afficher toutes les commandes qui commencent par les caractères déjà tapés.

Première invite : on entre « pas » et on appuie 2 fois sur **Tab**. Ce qui nous donne toutes les possibilités de commandes commençant par « pas ».

2ième invite : on ajoute un « s », on appuie de nouveau sur **Tab** et la commande est complétée



Le complément par tabulation peut servir à compléter les noms de fichiers quand on les saisit comme arguments de commandes. Quand on appuie sur **Tab**, la touche complète le nom du fichier autant qu'il est possible. Une seconde pression sur **Tab** provoque l'affichage par le shell d'une liste de tous les fichiers qui correspondent à l'ensemble de caractères courant. Tapez des caractères supplémentaires jusqu'à ce que le nom soit unique, puis utilisez le complément par tabulation pour terminer la ligne de commande.

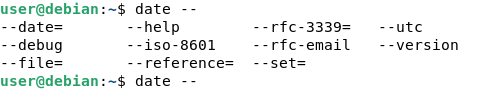
1 appui sur **Tab**: l’argument est complété au maximum (/pas devient /passwd)



Encore 2 appuis sur **Tab**. Cela nous révèle les 2 possibilités (avec ou sans tiret)



Le complément par tabulation (une ou deux fois **Tab** selon les options valides) peut servir avec les arguments et les options de nombreuses commandes. La commande **date** permet l’utilisation de plusieurs options qui peuvent servir à contrôler le comportement de la commande. On peut utiliser le complément par tabulation après une option partielle pour la compléter sans avoir à taper trop de texte.

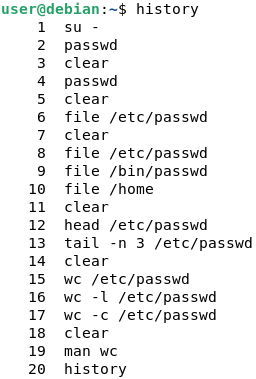


# Historique des commandes

La commande **history** affiche une liste des commandes précédemment exécutées, préfixées par un numéro de commande.



Le résultat est tronqué et montre les dernières commandes exécutées



# Exercices 2

1. Quel est le type du fichier de /usr/bin/cheese ?   
   Est-il lisible par des humains ?  
   Indice : Utilisez la commande « file » pour déterminer son type
2. Afficher les 3 premières lignes de /etc/group  
   Indice : La commande « head » affiche le début du fichier.
3. Afficher les 5 dernières lignes de /etc/group  
   Indice : La commande « tail » affiche la fin du fichier.
4. Afficher uniquement le nombre de mots contenus dans le fichier /etc/group  
   Indice : L’aide de la commande « wc » 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 :

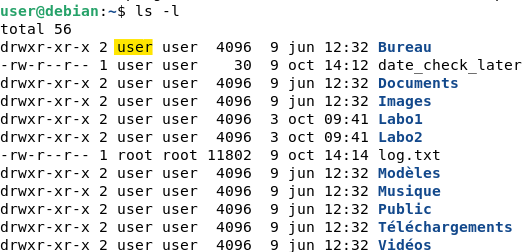
* *L’utilisateur standard* : il peut exécuter des tâches et des commandes communes. Ils ont des droits limités et ne peuvent pas exécuter des opérations critiques (modification réseau, accès aux fichiers système, …)
* *Le super utilisateur* : il est l’administrateur. Il s’appelle, plus communément, root. 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*.

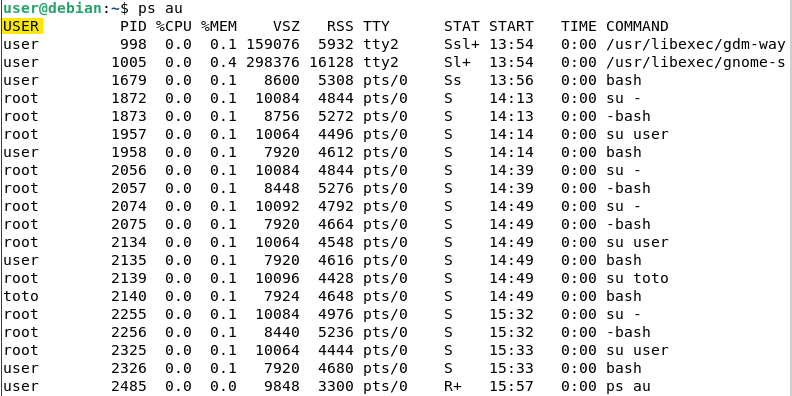


Pour afficher l'utilisateur associé à un fichier ou à un répertoire, utilisez la commande *ls*

*-l*. La troisième colonne indique le nom de l'utilisateur :

 *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 :



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 :

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

|  |  |
| --- | --- |
| **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*



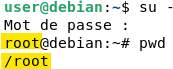
Bien évidemment, l’utilisateur peut se connecter à l’utilisateur root s'il en a les droits. Mais attention, exécuter la commande *su root* permettra à l’utilisateur courant de se connecter avec root mais il ne pourra pas utiliser certaines commandes.

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

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

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



C’est parce qu’avec le tiret, les variables d’environnement de l’user courant sont remplacés 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.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

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



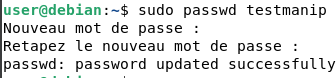
L’option **-c** ou **--comment** permet d’ajouter un commentaire (gecos) pour l’utilisateur. L’option **-m** ou **--create-home** permet de créer le répertoire “home” de l’utilisateur.

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.



Exercices 3

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** en tapant la commande nano /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

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Redémarrez votre machine avec la commande *reboot*.

1. Créez un utilisateur **toto** avec *adduser*. Le mot de passe sera **tttttt**.
2. 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.
3. Tout en restant sur votre utilisateur de base, mettez-lui un mot de passe simple.
4. Faites en sorte que l’utilisateur coco puisse aussi utiliser la commande *sudo*.
5. Affichez les informations de l’utilisateur **coco**