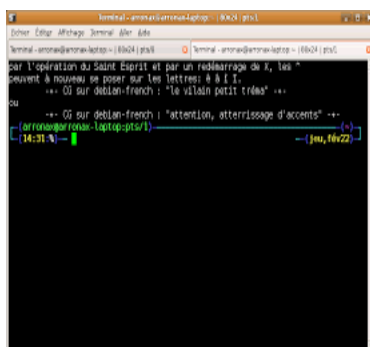


SHELL

Un shell en informatique est un programme qui reçoit des commandes informatiques données par un utilisateur à partir de son clavier pour les envoyer au système d'exploitation qui se chargera de les exécuter. Il se présente sous la forme d'une interface en ligne de commande accessible depuis la console ou un terminal. L'utilisateur lance des commandes sous forme d'une entrée texte exécutée ensuite par le shell. De nombreux shell existent, le plus connu étant « sh », le Bourne Shell . Le deuxième est le « bash » pour Bourne Again Shell , une version améliorée du premier, disponible et installé par défaut sur les systèmes d'exploitation macOS et Linux. « ksh », le « Korn Shell » est un autre exemple de shell, plus puissant et compatible avec bash. D'autres sont également notables comme « csh », « tcsh » ou « zsh ». Lorsque vous vous connectez pour la première fois au système (ou que vous ouvrez une nouvelle fenêtre Utilitaire de commande ou Utilitaire Shell) et que vous voyez apparaître le prompt, cela indique qu'un interpréteur de commande a été lancé automatiquement. Ce programme est appelé Shell de connexion.

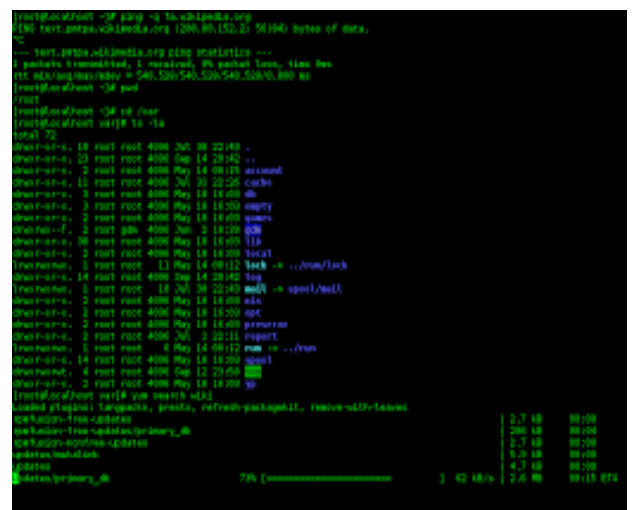
Un terminal est un shell. On en retrouve un ou plusieurs dans tous les systèmes Linux; le terminal



est un programme,

un émulateur de console qui nous permet de lancer des commandes en texte depuis l'interface graphique, c'est une interface de ligne de commande (CLI) par opposition à l'interface graphique utilisateur GUI: graphical user interface, qui nous permet de passer des commandes plutôt qu'en cliquant sur des icônes de programmes, par exemple. La console, elle, est l'écran noir avec du texte qui défile rapidement lorsque l'on démarre l'ordinateur. Dites-vous que tout ce qui se passe dans votre système GNU/Linux est écrit en texte script, quelque part dans le système de fichier.

Dans Linux, tout est fichier, certains sont des fichiers exécutable des scripts, et d'autres non. Les scripts prédéfinis permettent, entre autres, d'automatiser des opérations, des tâches, des processus, etc. Vous pouvez contrôler toutes les composantes de votre système Linux par le terminal. On peut utiliser le terminal pour plusieurs raisons (on pourrait arriver au même résultat désiré par l'interface graphique). C'est souvent plus simple d'entrer une ou deux commandes dans un terminal. Le terminal nous permet également de suivre tout changement qui s'opère dans le système. Les commandes que nous entrons dans le terminal proviennent du même langage informatique qu'Unix. Je n'entrerais pas dans le détail de toutes les fonctions du shell aujourd'hui mais j'ai préparé une petite série de captures



d'écran montrant l'exécution d'une commande simple dans un terminal ainsi que son résultat.

JOB 1:

Une commande est une instruction qu'un utilisateur envoie au système d'exploitation de son ordinateur pour lui faire exécuter une tâche. Il peut s'agir de manipuler des fichiers, d'accéder à des répertoires, de modifier des droits d'accès, etc. Du fait de la complexité des systèmes d'exploitation, il en existe un très grand nombre, et les actions précises de chacune d'elles sont de plus conditionnées par un jeu plus ou moins volumineux d'options et de paramètres. Elles constituent ainsi un outil extrêmement puissant, mais encore faut-il les connaître, et parfaitement comprendre leurs actions. Ces commandes peuvent être déclenchées soit indirectement par le biais d'outils graphiques (outils de l'environnement) soit directement par le biais de lignes de commandes saisies sur un terminal ou une console dans un langage extrêmement concis pour en simplifier la frappe, au risque de les faire apparaître comme extrêmement mystérieuses aux néophytes voire même aux utilisateurs aguerris. Pour afficher le manuel de la commande `ls`, il faut utiliser la commande `man ls`. `man` est une commande disponible sur les systèmes d'exploitation de type Unix. Elle permet de visionner les contenus d'une documentation formatée pour être exploitable par `man`.

Les éléments cachés sont des éléments qui ne sont pas visibles par défaut lorsque l'utilisateur affiche le contenu d'un dossier. Cette possibilité de dissimuler des éléments permet de protéger des fichiers de manipulations involontaires et d'alléger l'affichage des dossiers dans lesquels ils sont stockés. On peut voir les fichiers cachés du home de votre utilisateur en faisant `ls -d`. Les éléments cachés se reconnaissent au point précédant leur nom. Un fichier nommé `“test”` sera donc reconnu par le système comme un fichier caché alors qu'un fichier `“test”` ne le sera pas.

les fichiers cachés plus les informations sur les droits sous forme de liste sont visibles en utilisant la commande `ls -dl`. Leur marquage est une mesure de précaution pour vous empêcher d'endommager par mégarde le système de d'exploitation. Cependant, il peut arriver, pour une raison ou une autre, que vous ayez besoin d'y accéder. Dans ce qui suit, nous présentons différentes façons d'afficher les fichiers et dossiers cachés.

Les commandes de terminal ont généralement des options ou des commutateurs, que vous pouvez utiliser pour modifier la façon dont la commande fait ce qu'elle fait. Les options sont incluses dans la spécification POSIX pour les interfaces de ligne de commande. C'est aussi une convention séculaire établie avec les premières applications UNIX, il est donc bon de savoir comment les inclure dans vos scripts Bash lorsque vous créez vos propres commandes. En 1er mot tapé est une commande. Les lettres tapées après un tiret, et les mots tapés après 2 tirets, sont des options. Le reste ce sont des paramètres. Notez qu'il existe des commandes sans paramètres, d'autres sans options, et même certaines qui

prennent une commande en paramètres, donc pour ajouter des options à une commande il faut ajouter un tiret. Les 2 indispensables syntaxes sont première syntaxe un tiret et pour la deuxième syntaxes 2 tirets.

JOB 2:

De nombreux types de commandes sont disponibles dans bash pour afficher le contenu d'un fichier. Les commandes les plus couramment utilisées sont les commandes "cat", "more", "less", "head" et "tail". Pour lire l'ensemble du fichier, les commandes « cat », « more » et « moins » sont utilisées. Mais lorsque la partie spécifique du fichier doit être lue, les commandes « tête » et « queue » sont utilisées pour effectuer cette tâche.

La commande « head » est utilisée pour lire le fichier depuis le début et la commande « tail » est utilisée pour lire le fichier depuis la fin. Dans ce tutoriel, le présent tutoriel montre comment vous pouvez utiliser les commandes « tête » et « queue » avec différentes options pour lire la partie particulière d'un fichier. Vous pouvez utiliser n'importe quel fichier existant ou créer n'importe quel nouveau fichier pour tester les fonctions des commandes « head » et « tail ». Créez deux fichiers texte nommés products.txt et employee.txt avec le contenu suivant pour afficher l'utilisation des commandes « head » et « tail ». Pour afficher les 10 premières lignes du fichier ".bashrc" on utilise la commande **head .bashrc** et pour afficher les 10 dernières **tail .bashrc** . De plus pour les 20 dernières lignes du fichier on ajoute entre le "head" et ".bashrc" ce qui donne **head -n20 .bashrc**. Pareil pour les 20 dernières nous utilisons la commande **tail n-20 .bashrc** Appelée de cette manière, la commande head de Linux affiche les dix premières lignes du fichier spécifié. De cette façon, on peut jeter un coup d'œil rapide sur le début d'un fichier. Souvent, cela vous indique sur quoi porte le fichier. Dans l'usage quotidien, la commande head de Linux est souvent appelée avec l'option « -n » Avec le nombre qui le suit, on détermine le nombre de lignes à afficher. Par convention, la commande head de Linux est contrôlée par des paramètres optionnels. Dans le cadre des GNU Coreutils, il existe une forme longue pour chaque option ; pour les options les plus fréquemment utilisées, il existe également une forme courte, souvent déterminée historiquement.

JOB 3:

apt est un outil en ligne de commande qui permet de gérer des paquets en provenance de dépôts APT . Il permet donc principalement d'installer et de désinstaller des logiciels en ligne de commande sur Ubuntu. Il rassemble des fonctionnalités qu'implémentent les outils apt-get et apt-cache, mais rend leur usage plus convivial pour l'utilisateur final. apt est installé par défaut sur Ubuntu. Avec les versions précédant la version 2.0 d'apt, introduite avec Ubuntu Focal 20.04, il est possible d'indiquer en paramètre une expression régulière à la place des noms de paquets. Dans les commandes ci-dessous, remplacez « mon_paquet » par le nom de votre paquet. Si vous voulez installer plusieurs paquets en même temps, remplacez « mon_paquet » par la liste des paquets à installer, séparés par un espace. Cmatrix est un petit utilitaire qui permet d'afficher la Matrix directement dans votre terminal. Inutile donc indispensable, surtout pour flamber devant vos amis Windowsien. La méthode conseillée est

d'utiliser `sudo apt install mon_paquet` pour installer cmatrix. Pour lancer le paquet c'est tout simple pour utiliser le nom du paquet donc `cmatrix`. Pour mettre à jour un gestionnaire de paquets c'est la commande `apt-get update`, et pour mettre à jour ses différents logiciels c'est `apt-get upgrade`.

Google Chrome n'est pas une application présente dans les dépôts officiels. Elle ne bénéficie ni de la validation officielle Ubuntu, ni du support des développeurs des équipes officielles Ubuntu. Installer une application hors des dépôts officiels peut présenter des risques d'instabilité de votre système. Cette méthode, recommandée pour sa simplicité, permet de bénéficier de la toute dernière version stable de Google Chrome et installera toutes les dépendances nécessaires au fonctionnement de l'application. on va télécharger google chrome en utilisant la commande `sudo apt install google-chrome-stable`. Comme son nom l'indique, google-chrome-stable est la version stable de l'application, recevant les mises à jour de sécurité, et les mises à jour de fonctionnalités éprouvées. Il est conseillé de choisir cette version pour une utilisation quotidienne car c'est cette version qui est la plus sûre.

Tôt ou tard, vous devrez redémarrer votre machine. Que vous souhaitiez démarrer votre système après l'installation des mises à jour ou que vous ayez besoin de résoudre un problème, vous pouvez le faire en utilisant des commandes. La commande recommandée est `Reboot`. Aucun système d'exploitation n'est parfait. Même si c'était le cas, il pourrait y avoir des problèmes avec les pilotes et les applications. Linux ne fait pas exception. Pour éteindre une machine, la commande est `power off`.

JOB 4:

Pour créer un fichier user.txt qui va contenir User1 et User2 alors je vous propose d'utiliser la commande `rm user.txt` et ensuite `cat > user.txt` ce qui donnera les utilisateurs avec un retour à la ligne. Pour pouvoir créer un groupe appelé "Plateformeurs" il va falloir être déjà sur un superutilisateur et utiliser `sudo groupadd Plateformeurs`. La commande groupadd crée un nouveau compte de groupe en utilisant les valeurs spécifiées sur la ligne de commande plus les valeurs par défaut du système. Le nouveau groupe sera entré dans les fichiers système au besoin. On va pouvoir créer un utilisateur en utilisant la commande `sudo adduser user1` et `sudo adduser user2` et puis ça créera les 2 utilisateurs demandés. Pour pouvoir ajouter "User23 au groupe "Plateformeurs" la commande demandée est `sudo groupadd user2 plateformeurs`. Certains dossiers ou fichiers peuvent être protégés. On peut contourner ce problème en ouvrant un terminal. La commande qu'on va utiliser pour copier "user.txt" dans un fichier "droits.txt" est `cp user.txt droit.txt` puis copier sur le fichier "groupes.txt" on rajoute

juste `cp user.txt groupe.txt`. Lorsqu'un utilisateur crée un fichier ou un dossier, il en devient propriétaire et fait partie du groupe portant le même nom. Pour pouvoir changer le propriétaire du fichier "droit.txt" pour mettre à la pca "User1" la commande que l'on va utiliser est `sudo chown user1 droit.txt`. Nous voulons changer les droits du fichier, on peut modifier les valeurs dans l'onglet "permissions". On peut le faire si on a les droits, mais si le fichier n'est accessible que par root, on ne pourra pas le faire puisque par défaut, nous ne sommes pas "sudo". Pour changer les droits du fichier "droit.txt" pour que "User2" ai accès seulement en lecture la commande est `sudo chmod 740 droit.txt`. Pour changer cette fois les droit du fichier "groupe.txt" pour que les utilisateurs puissent accéder au fichier en lecture uniquement on utilise `sudo chmod 744 groupes.txt`. Enfin pour terminer on change les droits du fichier pour que le groupe "Plateformeurs" puissent y accéder en lecture/écriture j'utilise la commande `sudo chmod 760 groupes.txt`

JOB 5:

Les alias sont des substitutions abrégées de commandes répétitives et longues à taper dans la console. Il est possible de définir vos alias dans deux fichiers cachés. C'est un moyen très puissant pour gagner en productivité en personnalisant le shell Bash. Créez vos propres commandes Linux à l'aide de fonctions d'alias pour des tâches répétitives, raccourcir de longue commandes et configurez des commandes standards avec les options que vous utilisez très souvent. Pour ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la" on utilise `alias la='ls -la'`. Maintenant on souhaite ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant

"update" la commande demandé est `alias update='apt-get update'`

Enfin on ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade" on va utiliser `alias upgrade='apt-get upgrade'`

Les variables d'environnement constituent un moyen d'influencer le comportement des logiciels sur votre système. Par exemple, la variable d'environnement « LANG » détermine la langue que les logiciels utilisent pour communiquer avec l'utilisateur. Les variables sont constituées de noms auxquels on assigne des valeurs. Ainsi, le système d'un utilisateur français devrait avoir la valeur « fr_FR.UTF-8 » assignée à la variable « LANG ». La signification d'une variable d'environnement et le type de valeur qui peut lui être assignée sont déterminés par l'application qui utilise celle-ci. Il existe un petit nombre de variables d'environnement bien connues, dont le sens et le type de valeur sont bien déterminés, et qui sont utilisées par de nombreuses applications. Bien que quelques applications de configuration en mode graphique manipulent en réalité des variables d'environnement, la ligne de commande offre un maximum de flexibilité pour créer et modifier ces variables. Pour ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre

nom d'utilisateur on utilise `export USER=(utilisateur)` . Pour Mettre à jour les modifications de votre `bashrc` dans votre shell actuel: `source .bashrc` . Certaines variables sont directement définies par le système. Par exemple le fameux `PATH` ou `PWD` – que l'on récupère généralement avec la commande `pwd`. Lorsque l'on veut afficher la valeur d'une variable, on utilise généralement la commande `echo` suivi du nom de la variable. Néanmoins, il existe une fonction spécifique à cet usage. Cette dernière permet en outre d'afficher toutes les variables d'environnement disponibles, il s'agit de `printenv`. Si vous l'évoquez sans autre paramètre, cette dernière affiche toutes les variables actuellement déclarées. Pour visualiser le contenu d'une variable en particulier, vous lui passez simplement le nom de la variable en argument (sans le "\$"). Ducoup pour afficher les variables d'environnement on utilise la commande `printenv`.

Pour finir le Path est utilisé sur Linux pour contenir des chemins vers des programmes. Ces chemins permettent d'exécuter directement une commande dans le terminal. Pour ajouter à votre Path le chemin `"/home/votre utilisateur/Bureau"`: `export.path=spath'`

JOB 6:

La commande `Tar` de Linux est l'une des plus utilisées pour la compression, archiver, désarchiver, ect... `Tar` est l'abréviation de Tape archive et sert à compresser une collection de fichiers et de dossiers. Dans la plupart des cas, une fois la compression effectuée à l'aide de `tar`, on obtient un fichier `.tar`. Une compression plus poussée est effectuée à l'aide de `gzip`, ce qui donne un fichier `.tar.gz`. Avec `tar`, vous pouvez compresser et décompresser des fichiers.

JOB 7:

Notre première méthode de création de fichiers texte utilise le cat commander. C'est utile si vous souhaitez ajouter immédiatement du texte à votre nouveau fichier. Pour pouvoir créer un fichier "une_commande.txt" avec le texte suivant "Je suis votre fichier texte" on va utiliser la commande `echo "je suis votre fichier"> une_commande.txt`, et pour vérifier ça va être la commande `cat une_commande.txt`

La commande `wc` (word count) sous Linux permet de connaître le nombre de lignes, de mots, d'octets et de caractères dans des fichiers spécifiés par les arguments de fichier. La syntaxe de la commande `wc` est indiquée ci-dessous. Pour compter le nombre de lignes présentes dans votre fichier de source apt on va utiliser une longue commande qui est "`cat /etc/apt/sources.list | wc -l`" et les enregistrer dans le fichier nommé "nb_lignes.txt" `cat -n /etc/apt/sources.list > nb_lignes.txt`

Pour afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save_sources", la commande est `cat sources.list> save_sources`

Pour aller plus loin..

La commande « `tree` » est utilisée pour répertorier les fichiers d'un répertoire dans une structure en forme d'arborescence. La plupart des gens utilisent la commande de base « `ls` » et ne sont pas au courant de la commande « `tree` ». L'avantage de l'utilisation de la commande « `tree` » est qu'elle donne une liste très structurée d'un répertoire d'une manière très facile à saisir et simple.

Tree est disponible pour les principaux systèmes d'exploitation tels que Windows, Linux ou Unix et MacOS. La commande utilisée pour installer la commande tree est `sudo apt-get install tree`. Pour lancer la commande tree en arrière-plan qui aura pour but d'afficher toute l'arborescence en de votre / en enregistrant le résultat dans un fichier "tree.save" est `tree`

puis `tree > tree.save`. Ensuite pour pouvoir lister les éléments présents dans le dossier courant est utilisé directement le résultat de votre première commande pour compter le nombre d'éléments trouvés on utilise la commande `ls | wc -l`. Puis pour terminer on veut lancer une commande pour updater vos paquets, si l'update réussit alors, vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre upgrade ne se lancera pas on utilise d'abord passer en route `su` - tapez votre mot de passe puis la commande `apt update && apt upgrade`