**Linux en ligne de commande**

1. Organisation et gestion des fichiers.
2. organisation des répertoires :

Les systèmes Unix reposent sur une arborescence de fichiers différente des systèmes Windows. En effet, vous ne trouverez pas sur Unix de périphérique en tant que tel ; la philosophie est la suivante : tout est fichier. Les fichiers peuvent donc représenter des données, des programmes, des périphériques, des processus ... Ils sont organisés suivant une arborescence qui peut varier selon les systèmes. En règle générale nous retrouvons la structure basique suivante :

|  |  |
| --- | --- |
| / | racine des fichiers (ROOT) |
| /etc | répertoire contenant les fichiers de configuration. |
| /usr | répertoire des applications |
| /var | répertoire des données |
| /proc | répertoire des processus |
| /dev | répertoire des périphériques |
| /home | répertoires privés des utilisateurs. |
| /tmp | répertoire des fichiers temporaires. |

1. Type des fichiers :

Les fichiers n'ont pas d'extension particulière et la notion même d'extension, bien que couramment utilisée, n'existe pas. Le type d'un fichier est identifié par son contenu. La commande **file** permet d'en connaître le type.

1. Manipulation :

Les fichiers sont organisés en répertoires et sous-répertoires comme sur un système Windows. La navigation peut se faire au travers de commandes simples :

|  |  |
| --- | --- |
| pwd | Indique le répertoire courant. |
| uname | Affichage des informations relatives UNIX installée |
| date | Affichage de la date et de l'heure instantanés. |
| mkdir rep | Crée dans le répertoire courant un nouveau répertoire appelé rep. |
| ls | Liste les fichiers du répertoire courant. Des options peuvent être ajoutées :  ls -a liste tous les fichiers, y compris ceux cachés  ls -l liste les fichiers en ajoutant les informations telles que les droits et le propriétaire. |
| cd | Change le répertoire courant. Par exemple cd /home vous amène  directement dans le répertoire home. |
| touch fichier | Créer un fichier vide. |
| cat | Affichage complet du contenu d'un fichier |
| more | Affichage progressif du contenu d'un fichier |
| head | Affichage du début d'un fichier |
| tail | Affichage de la fin d'un fichier |
| rm fichier | Supprime le fichier 'fichier'. L'option -f évite d'avoir à confirmer chaque effacement.  L'option -r permet de supprimer un sous-répertoire et son contenu. |
| rmdir rep | Supprime le répertoire 'rep' s'il est vide. |
| cp source destinataire | Copie le fichier dont le chemin est source vers le chemin destinataire L'option -R indique une copie récursive (incluant les sous-répertoires). |
| mv source destinataire | Déplace le fichier dont le chemin est source vers le chemin destinataire. L'opération réalisée est un couper-coller. |

1. Utilisation des liens

Les systèmes Unix permettent la création de liens. Un lien est en quelque sorte un raccourci vers un fichier. C'est est en apparence un fichier comme un autre. Il existe deux principales sortes de liens : les liens physiques et les liens symboliques, la commande **ln** permet de créer des liens comme suit :

|  |  |
| --- | --- |
| ln –s fichier fichier1 | Lien symbolique |
| ln fichier fichier2 | Lien physique |

1. Edition d'un fichier :

L'éditeur de fichier que vous trouverez sur tous les systèmes Unix est **vi**. Cet éditeur est certes peu pratique pour un novice mais connaître quelques commandes de base se révèle très souvent utile. Il prend comme paramètre le nom du fichier à éditer.

**Vi** est un éditeur non graphique et sans menu. Il possède deux modes : édition et commande. Le mode commande permet de sauver, quitter, copier, coller... le document. Le mode édition permet d'ajouter ou modifier du texte. A tout moment, il est possible de passer en mode commande en pressant la touche ESC du clavier. Le passage en mode édition, se fait en pressant la touche 'i' pour une édition avec insertion à la position courante du curseur. 'a' pour une insertion à la position suivant le curseur. 'R' pour le remplacement.

Le mode commande permet de multiples commandes, certaines sont détaillées ci-dessous.

|  |  |
| --- | --- |
| :q | Quitter vi (sans sauvegarde). |
| :w | Sauvegarder le fichier. |
| :q ! | Forcer l’éditeur à quitter sans sauvegarder |
| dd | Supprimer la ligne courante (couper). |
| yy | Copier la ligne courante. |
| p | Coller les lignes précédemment coupées ou copiées |

1. Les métas-caractères :

Les métas caractères sont des caractères qui possèdent une signification particulière pour le shell et qui simplifient parfois pas mal la vie.

|  |  |
| --- | --- |
| **Caractère** | **Fonction** |
| \* | Remplace un ensemble de caractères |
| ? | Remplace un caractère dans une chaine |
| [ ] | Permette d’indiquer un ensemble |
| - | Séparateur d’un ensemble |
| # | Caractère de commentaire |

1. Utilisateurs et groupes :

Pour travailler sur un ordinateur à plusieurs utilisateurs, il convient de créer pour chacun ce que l’on nomme un compte. C’est une entité gérée par le système et qui permet à un utilisateur de se connecter à la machine. Elle se compose essentiellement d’un nom de connexion (login) qui identifie l’utilisateur de manière unique, d’un mot de passe pour éviter les fraudes et d’un répertoire de connexion pour accueillir ses fichiers personnels.

1. La création d’un compte

Des utilitaires permettent de créer un compte pour un utilisateur. Le plus courant est **useradd** ou **adduser**. À titre d’exemple et pour ajouter l’utilisateur toto, il suffit de taper :

**useradd toto**

Pour l’instant, aucun mot de passe n’a été affecté au compte. Mais l’utilitaire n’a pas laissé le compte sans protection. Pour lui affecter un mot de passe, il faut taper :

**passwd xxx**

Le programme demande alors à l’utilisateur de taper le mot de passe une première fois, puis une deuxième fois, sans afficher quoi que ce soit. Il faut simplement taper 'Entrée' après le mot de passe, à chaque fois.

En ce qui concerne le répertoire de connexion qui doit abriter les fichiers personnels de l’utilisateur, il a déjà été créé dans le répertoire /home par la première commande. Il porte le nom de l’utilisateur toto. C’est donc : /home/toto. Nous aurons compris au passage que /home contient tous les répertoires personnels des utilisateurs de la machine.

1. La suppression d’un compte

La suppression d’un compte utilisateur est, comme on peut le deviner, plus facile. Une seule commande suffit, en effet :

**userdel toto**

Les fichiers personnels de l’utilisateur toto ne sont pas supprimés par sécurité, car cela serait définitif. Le répertoire de connexion est conservé afin qu’on puisse avoir le temps de copier ou de déplacer les fichiers importants. Doit-on rappeler que l’on peut supprimer ce répertoire en tapant :

**rm –rf /home/toto**

Si l’on n’est pas dans ce cas et que l’on veut tout supprimer à l’aide d’une seule commande, parce que le répertoire personnel de toto ne contient rien d’important, alors l’option –r convient :

**userdel –r toto**

1. La notion de groupe

L’intérêt de créer un groupe d’utilisateur est multiple et ne peut être abordé ici de façon exhaustive. Notons simplement que cela permet de désigner plusieurs utilisateurs par un seul groupe.

Ajouter le groupe user se fait très simplement avec la commande groupadd

**groupadd user**

Supprimer ce même groupe est également aussi simple avec groupdel :

**groupdel user**

Pour faire appartenir l’utilisateur toto au groupe user, la commande usermod s’impose :

**usermod –g user toto**

L’option –g sert à désigner le nouveau groupe. Elle doit impérativement être suivie du nom du groupe : user. Le login de l’utilisateur concerné par ce changement doit absolument venir en dernier sur la ligne de commande, d’où la position de toto à la fin.

Pour faire en sorte qu’un utilisateur quitte un groupe, il suffit de le faire appartenir à un autre groupe à l’aide de la même commande

Par ailleurs, il est impossible de supprimer un groupe si un seul utilisateur en est encore membre. Il faut donc faire appartenir tous ses membres à un ou plusieurs autre(s) groupe(s) avant de supprimer celui-ci.

Pour créer l’utilisateur toto et le mettre dans le groupe user directement (cela évite d’avoir en plus à supprimer le groupe toto), il suffit d’utiliser l’option –g à la commande useradd pour spécifier le groupe :

**useradd –g user toto**

Sinon, les commandes suivantes s’imposent, en supposant que le groupe user a déjà été crée :

**useradd toto**

**usermod –g user toto**

**groupdel toto**

1. Le fichier des mots de passe :

Tous les changements apportés par les commandes vues précédemment sont enregistrés dans des fichiers contenus dans le répertoire /etc. Il porte en effet le même nom que le programme qui modifie le mot de passe.

Le fichier /etc/passwd est un fichier texte comportant la liste de tous les utilisateurs du système, une ligne par utilisateur. Elle comporte certaines informations séparées par le caractère deux-points ':'. On parle de champs et il y en a sept par ligne et donc par utilisateur.

* Le premier est le login de l’utilisateur, son nom de connexion que nous avons vu précédemment
* Le deuxième est le mot de passe.
* Le troisième est un numéro d’identification unique pour chaque utilisateur. Il est appelé UID (User IDentity)
* Le quatrième est le numéro du groupe auquel appartient l’utilisateur. Il est appelé GID (Group IDentity)
* Le cinquième est un champ facultatif destiné à contenir le prénom et le nom de l’utilisateur
* Le sixième est le répertoire personnel de l’utilisateur, généralement situé dans /home
* Le septième et dernier champ est le shell de l’utilisateur, le programme qui accueillera l’utilisateur à sa connexion et lui servira d’interface avec le noyau.

1. Le fichier des groupes :

Tout comme pour les utilisateurs, les groupes ont eux aussi besoin d’un fichier pour les enregistrer : /etc/group. La structure est quasiment identique : une ligne par groupe et quatre champs spécifiques aux groupes (cf. man group).

* Le premier champ est le nom du groupe
* Le deuxième champ est le mot de passe pour se connecter au groupe si l’on est un utilisateur non membre du groupe
* Le troisième est le GID que nous avons vu dans le fichier /etc/passwd
* La liste des membres supplémentaires du groupe, séparés par une virgule. Chaque utilisateur peut se connecter à des groupes autres que son groupe d’appartenance principal en utilisant la commande newgrp.

Vous pouvez récupérer le nom d’utilisateur et les groupes auxquels nous appartenons grâce aux deux commandes : **whoami** et **groups**

III. Les commandes find et grep :

La commande find permet de retrouver des fichiers à partir de certains critères

La commande grep quant à elle permet d'explorer une série de fichiers d'un même répertoire à la recherche de texte satisfaisant une expression régulière donnée.

1. La commande find et ses option :
   1. Syntaxe de la commande find

La syntaxe est la suivante :

Find <répertoire de recherche> <critères de recherche>

Les critères de recherche sont les suivants :

|  |  |
| --- | --- |
| -name | recherche sur le nom du fichier |
| -perm | recherche sur les droits d'accès du fichier |
| -links | recherche sur le nombre de liens du fichier |
| -user | recherche sur le propriétaire du fichier |
| -group | recherche sur le groupe auquel appartient au fichier |
| -type | recherche sur le type (d=répertoire, c=caractère, f=fichier normal) |
| -size | recherche sur la taille du fichier |
| -atime | recherche par date de dernier accès en lecture du fichier |
| -mtime | recherche par date dernière modification du fichier |
| -ctime | recherche par date de création du fichier |

Il est possible de combiner les critères avec des opérateurs logiques :

* critère 1 critere2 ou critere1 -a critere2 correspond au et (and) logique
* !critère non logique
* \(critère1 -o critère2\) correspond au ou (or) logique

La commande find doit être utilisée avec l'option -print. Sans l'utilisation de cette option, même en cas de réussite dans la recherche, find n'affiche rien à la sortie standard.

* 1. Recherche par nom de fichier :

Pour chercher un fichier dont le nom contient la chaine de caractères chaine1 à partir du répertoire /usr, la syntaxe est la suivante :

Find /usr –name chaine1 –print

Pour rechercher tous les fichiers se terminant par .c dans le répertoire /usr, la syntaxe est la suivante :

find /usr -name "\*.c" –print

* 1. Recherche suivant la date de dernière modification

Pour connaître les derniers fichiers modifiés dans les x derniers jours dans l'arborescence, la syntaxe est  la suivante :

Find / -mtime x –print

1. La commande grep et ses options
   1. Syntaxe de la commande grep

La syntaxe de la commande grep est la suivante :

Grep - <option> <expression> <fichier>

Les options de la commande grep sont les suivantes :

|  |  |
| --- | --- |
| -v | affiche les lignes ne contenant pas la chaine |
| -c | compte le nombre de lignes contenant la chaine sans afficher les lignes |
| -n | chaque ligne contenant la chaine est numérotée |
| -x | ligne correspondant exactement à la chaine |
| -l | affiche le nom des fichiers qui contiennent la chaine |
| ^ | Début de ligne |
| . | Caractère quelconque |
| $ | Fin de ligne |
| […] | Plage de caractères permis |
| [^…] | Plage de caractères interdits |
| \{...\} | Répétition |