Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)

Présenté par : Mohamed Belhassen

2019-2020

Sujet 104 : Disques, systèmes de fichiers Linux ,FHS

- 104.1 Création de partitions et systèmes de fichiers. (Weight 2)
- 104.2 Maintenir l'intégrité des systèmes de fichiers (Weight 2)
- 104.3 Contrôle du montage et du démontage des systèmes de fichiers (Weight 3)
- 104.5 Gérer les permissions et les propriétaires des fichiers (Weight 3)
- 104.6 Créer et changer les liens symboliques et physiques sur les fichiers (Weight 1)
- 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats (Weight 2(Weight 3)

Sujet 104 : Disques, systèmes de fichiers Linux ,FHS

- Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif
  - Gestion des tables de partition MBR et GPT
  - Utilisation des différentes commandes mkfs pour le paramétrage des partitions et la création des différents systèmes de fichiers comme :
    - ext2/ext3/ext4
    - XFS
    - VFAT
    - exFAT
  - Connaissance de base de Btrfs, y compris les systèmes de fichiers sur plusieurs périphériques, la compression et les sous-volumes.

Sujet 104 : Disques, systèmes de fichiers Linux ,FHS

- Partial list of the used files, terms and utilities
  - fdisk
  - gdisk
  - parted
  - mkfs
  - mkswap

Création de partitions et systèmes de fichiers.

Création de partitions et systèmes de fichiers

- Description : Les candidats doivent être capables de créer des partitions et des systèmes de fichiers. Ceci inclut la prise en charge des partitions d'échange (swap).
- Termes, fichiers et utilitaires utilisés :
  - Fdisk
  - Mkfs
  - mkswap

Disques durs et partitionnement

- Un disque dur est composé de plateaux reliés à un moteur central, avec des têtes de lecture de part et d'autre de chacun des plateaux.
- Sur chaque plateau se trouvent des pistes cylindriques découpées en secteurs.
- L'adressage d'un secteur est une référence au cylindre, à la tête de lecture utilisée, à la piste, et enfin au secteur.
- À l'installation, un disque dur n'est ni partitionné, ni formaté.
  - Partitionner signifie définir sur le disque un ou plusieurs espaces, ou partitions,
  - et formater signifie préparer une partition à recevoir des informations en utilisant un système de fichiers défini.

Les partitions

- Une partition est définie par son type, son emplacement de début de partition et enfin soit sa taille, soit son emplacement de fin de partition.
- Un partitionnement est réversible (non physique).
- Une seule partition est activée à la fois au niveau du BIOS : cette activation indique où le BIOS doit aller chercher le noyau du système d'exploitation pour le démarrage.

Les partitions

- Il existe trois sortes de partitions :
  - les partitions principales : leur nombre est limité à quatre et elles supportent tous types de systèmes de fichiers ;
  - la partition étendue : elle ne peut contenir que des partitions logiques et ne peut pas recevoir de systèmes de fichiers. Elle ne peut exister que s'il existe une partition principale ;
  - les partitions logiques : elles sont contenues dans une partition étendue. Elles ne sont pas limitées en nombre et acceptent tout type de systèmes de fichiers

Organisation des partitions sous Linux

- Les descripteurs de disques durs dans le répertoire /dev commencent par **hd** pour les périphériques de type IDE ou par **sd** pour les périphériques de type **SCSI**.
- Une lettre additionnelle est ajoutée au descripteur pour désigner le périphérique.
- Il y a généralement deux contrôleurs IDE en standard sur un PC, chaque contrôleur supportant deux périphériques (disques, lecteur de cédérom/DVD...).

Disques

- /dev/hda
  - périphérique IDE primaire maître (Disque dure)
- /dev/hdb
  - périphérique IDE primaire esclave IDE
- /dev/hdc
  - périphérique IDE secondaire maître (CD-ROM)
- /dev/hdd
  - périphérique IDE secondaire esclave
- /dev/sda
  - premier disque SCSI
- /dev/sdb
  - Second disque SCSI /sata

Disques

Désignation des périphériques IDE		
	Primaire	**Secondaire**
Maître	a	C
Esclave	b	D

- Pour le périphérique maître sur le contrôleur primaire : hda
- Pour le périphérique esclave sur le contrôleur secondaire : hdd.
- Les périphériques SCSI sont désignés en fonction de leur position dans la chaîne SCSI (sda, sdb, sdc, etc.).

Partitionnement de disques

- On utilise la commande fdisk pour configurer une nouvelle partition.
- Par exemple, pour le premier disque IDE :
  - fdisk /dev/hda

- Voici une liste des différentes commandes internes de fdisk :
  - a : (dés)active un indicateur « bootable » ;
  - b : édite le libellé de disque bsd ;
  - c : (dés)active l'indicateur de compatibilité DOS ;
  - d : supprime une partition ;
  - l : répertorie les types de partition connus ;
  - m : affiche la liste des commandes ;

Partitionnement de disques

- n : ajoute une nouvelle partition ;
- o : crée une nouvelle table de partition DOS vide ;
- p : affiche la table de partition ;
- q : quitte le programme sans enregistrer les modifications ;
- s : crée un nouveau libellé de disque Sun vide ;
- t : change l'ID système d'une partition ;
- u : change l'unité d'affichage/saisie ;
- v : vérifie la table de partition ;
- w : écrit la table sur le disque et quitte le programme ;
- x : fonctions supplémentaires (experts seulement)

Partitionnement de disques

- Ci-dessous, un exemple de table de partitionnement obtenu avec l'option « l » de fdisk :
  - sudo fdisk -l

```
Périphérique Amorçage   Début      Fin Secteurs Taille Id Type
/dev/sda1    *           2048  1050623  1048576   512M  b W95 FAT32
/dev/sda2             1052670 42866687 41814018    20G  5 Étendue
/dev/sda5             1052672 42866687 41814016    20G 83 Linux


Disque /dev/loop8 : 346,34 MiB, 363151360 octets, 709280 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
```

Application: Partitionnement et Formatage d'une partition

- Suivre les instructions du tutoriel suivant:
  - https://www.howtogeek.com/106873/how-to-use-fdisk-to-manage-partitions-on-linux/
- NB:
  - ne pas utiliser la commande d'écriture (w) pour ne pas endomager votre machine.
  - Pour quitter sans écrire les modifications sur disque, tapez « q » à la fin des manipulations de la commande fdisk

Partitionnement de disques: partition swap

- Le système d'exploitation utilise une zone d'échange (swap) sur le disque comme une extension de la mémoire physique.
- Selon les besoins, il y aura donc un échange entre la mémoire physique et la zone swap.

```
Disque /dev/hda : 255 têtes, 63 secteurs, 3647 cylindres
Unités = cylindres sur 16065 * 512 octets

Périphérique Amorce    Début       Fin    Blocs   Id  Système
/dev/hda1              1         255   2048256    b  Win95 FAT32
/dev/hda2   *        256         386   1052257+  83  Linux
/dev/hda3            387        1597   9727357+  83  Linux
/dev/hda4           1598        3647  16466625    f  Win95 Etdue (LBA)
/dev/hda5           1598        2744   9213246   83  Linux
/dev/hda6           2745        3509   6144831   83  Linux
/dev/hda7           3510        3647   1108453+  82  Echange Linux
```

Partitionnement de disques

- Linux utilise deux types de partitions : Linux (Linux native) et Echange Linux (swap) comme on peut le constater sur la figure.
- La **première** partition est une partition qui peut contenir un système Windows et la **quatrième** une partition de type étendu qui permet de créer des partitions logiques

```
Disque /dev/hda : 255 têtes, 63 secteurs, 3647 cylindres
Unités = cylindres sur 16065 * 512 octets

Périphérique Amorce    Début       Fin     Blocs   Id  Système
/dev/hda1              1         255    2048256    b  Win95 FAT32
/dev/hda2   *        256         386    1052257+  83  Linux
/dev/hda3            387        1597    9727357+  83  Linux
/dev/hda4           1598        3647   16466625    f  Win95 Etdue (LBA)
/dev/hda5           1598        2744    9213246   83  Linux
/dev/hda6           2745        3509    6144831   83  Linux
/dev/hda7           3510        3647    1108453+  82  Echange Linux
```

Arborescence des fichiers sous Linux

Arborescence des fichiers sous Linux

- **Objectifs**

  ⇒ Connaître l'organisation de l'arborescence standard des fichiers sous Linux.
- **Mots clés:**
  - boot,
  - bin,
  - lib,
  - dev,
  - etc,
  - proc,
  - root,
  - home,
  - tmp,
  - usr,
  - var,
  - local

Les répertoires de base

- Les répertoires de base de l'arborescence standard de fichiers sont les suivants:
- **/boot** : contient principalement le fichier binaire du noyau ainsi que les ressources nécessaires à son lancement au démarrage ;
- **/dev** : contient les fichiers des périphériques (devices) de la machine ainsi que des fichiers spéciaux ;
- **/etc** : répertoire très important où sont stockés tous les fichiers de configuration du système en général et des différents démons en particulier. Il s'agit du répertoire à sauvegarder pour pouvoir restaurer la configuration d'une machine ;

Les répertoires de base

- Les répertoires de base de l'arborescence standard de fichiers sont les suivants (suite):
- /home : répertoire où sont stockés par défaut les répertoires home des utilisateurs du système ;
- /proc : contient les informations nécessaires au noyau. C'est une arborescence virtuelle généralement en lecture seule sauf proc/sys ;
- /root : répertoire home du super-utilisateur (root) ;

Les répertoires de base

- Les répertoires de base de l'arborescence standard de fichiers sont les suivants (suite):
  - **/tmp** : permet aux applications et aux utilisateurs d'avoir un espace d'échange où ils peuvent stocker leurs fichiers temporaires. Il est effacé à chaque redémarrage de la machine (« reboot ») ;
  - **/usr** : contient les fichiers nécessaires aux applications, la documentation, les manuels, les fichiers sources ainsi que des librairies généralement statiques et générées à l'installation des logiciels standards de la distribution ;

Les répertoires de base

- Les répertoires de base de l'arborescence standard de fichiers sont les suivants (suite):
  - **/usr/local** : arborescence qui sert à installer les logiciels supplémentaires ;
  - **/var** : contient les fichiers journaux des différents démons (donc variable) ainsi que les spools de mail, d'impression, de cron, etc.

Les autres répertoires

- **/bin et /sbin** : contiennent l'ensemble des binaires indispensables au démarrage de la machine et les commandes essentielles d'administration ;
- **/lib et /usr/lib** : contiennent les librairies nécessaires aux commandes précédentes.

Formatage et types de systèmes de fichiers

Formatage et types de systèmes de fichiers

- **Objectifs**

  ⇒ Connaître les différents types de systèmes de fichiers reconnus par Linux et leurs spécificités.

  ⇒ Maîtriser la création d'un système de fichiers sur une partition.

- Mots clés ext2, ext3, ext4, reiserfs, vfat, xfs, mkfs, tune2fs

Formatage et types de systèmes de fichiers

- Les principaux types de système de fichiers supportés par Linux sont présentés dans le tableau 4

Système de fichiers	Commande de création
ext2	mke2fs ou mkfs.ext2
ext3	mke2fs -j ou mkfs.ext3
ext4	mkfs.ext4 /dev/sdb1 Ou mke4fs -t ext4 /dev/sdb1
reiserfs	Mkreiserfs
Xfs	mkfs.xfs
vfat	mkfs.vfat

Formatage et types de systèmes de fichiers

- Le système de fichiers ext3 est une simple extension du format standard ext2 de Linux :
  - il intègre un journal qui enregistre toutes les opérations effectuées sur le disque.
  - Ceci permet une récupération plus rapide et sûre du système en cas d'arrêt brutal de la machine.
- L'instruction générale de création d'un système de fichiers est :
  - mkfs –t type de fichier partition

Formatage et types de systèmes de fichiers

- Il existe des commandes équivalentes pour chaque type de systèmes de fichiers, par exemple mkfs.ext3, mkfs.vfat (voir tableau)

Système de fichiers	Commande de création
ext2	mke2fs ou mkfs.ext2
ext3	mke2fs -j ou mkfs.ext3
ext4	mkfs.ext4 /dev/sdb1 Ou mke4fs -t ext4 /dev/sdb1
reiserfs	Mkreiserfs
Xfs	mkfs.xfs
vfat	mkfs.vfat

Formatage d'une partition

- Exemples de formatage de la partition hda1 avec création d'un système de fichiers de type ext3 (les trois commandes sont équivalentes):
  - mkfs.ext3 /dev/hda1
  - mkfs -t ext3 /dev/hda1
  - mke2fs -j /dev/hda1 # création du journal spécifique
- Il est aussi très facile de transformer une partition ext2 en ext3 avec l'instruction tune2fs pour créer le journal :
  - tune2fs -j /dev/hda1

Contrôle de l'intégrité du système de fichiers et réparation

Contrôle de l'intégrité du système de fichiers et réparation

- Objectifs

  ⇒ Maîtriser la remise en état d'un système de fichiers endommagé.

- Mots clés
  - fsck, e2fsck, debugfs, dumpe2fs

Vérification et correction du système de fichiers

- Si le système de fichiers est endommagé ou corrompu, l'utilitaire fsck est utilisé pour vérifier et corriger le système.
- L'instruction générale de vérification du système de fichiers est :
  fsck -t type-de-fichier partition
- De même que précédemment, il existe des commandes équivalentes pour chaque type de systèmes de fichier, par exemple fsck.ext3

Vérification et correction du système de fichiers

- L'option -i permet de laisser l'utilitaire fsck essayer de corriger lui-même les problèmes rencontrés.
- L'exemple qui suit permet de vérifier l'intégrité d'un système de fichiers reiserfs :
  - fsck.reiserfs /dev/hda1
- Une vérification de toutes les partitions est faite au démarrage du système par la commande fsck.

Vérification et correction du système de fichiers

- La commande e2fsck est équivalente à fsck –t ext2.
- En cas de problèmes plus important, il est possible d'utiliser l'utilitaire debugfs. Il est utilisé pour examiner et modifier l'état d'un système de fichiers formaté en ext2. Il permet par exemple de retrouver des inodes de fichiers supprimés (récemment) et de les restaurer.
- L'utilitaire dumpe2fs permet d'afficher les informations d'un système de fichiers formaté en ext2. Il est par exemple ainsi possible de connaître la date du dernier montage d'un système de fichiers.

Montage et démontage d'un système de fichiers

Montage et démontage d'un système de fichiers

- Objectifs

  ⇒ Savoir monter et démonter un système de fichiers sous Linux.

- Mots clés
  - mount, umount, du, df, /etc/fstab

Montage et démontage manuel

- Pour pouvoir utiliser un système de fichiers, celui-ci doit être monté sur un point de montage de l'arborescence Linux : son contenu est alors accessible comme un simple répertoire.
- Le système d'exploitation réalise alors diverses tâches de vérification afin de s'assurer que tout fonctionne correctement.

Montage et démontage manuel

- La commande mount accepte deux arguments :
  - le premier est le fichier spécial correspondant à la partition contenant le système de fichiers ;
  - le second est le répertoire sous lequel il sera monté (point de montage).
- Il peut être nécessaire de spécifier le type de fichiers avec l'option -t au cas où Linux ne parviendrait pas à le déterminer automatiquement.

Montage et démontage manuel

- La commande umount permet le démontage du système de fichiers.
- Voici un exemple de montage et de démontage d'une clé USB de type « flashdisk » décrite par le fichier device sda :
  - mount /dev/sda1 /mnt/flashdisk
  - umount /mnt/flashdisk

Montage et démontage automatique

- Le fichier /etc/fstab est utilisé pour le montage automatique des systèmes de fichiers au moment du démarrage du système.
- Chaque ligne du fichier fstab décrit la manière de montage d'un système de fichiers, et ceci à travers six champs séparés par des espaces.

```
#cat /etc/fstab
LABEL=/     /           ext3    defaults          0 1
/dev/hda5   /home       ext3    defaults          0 2
none        /proc       proc    defaults          0 0
/dev/cdrom  /media/cdrom iso9660 ro,noauto,owner 0 0
/dev/hda3   /usr        ext3    defaults          0 2
/dev/hda6   /var        ext3    defaults          0 2
/dev/hda2   swap        swap    defaults          0 0
```

Format du fichier fichier fstab

- Le premier champ donne le nom de périphérique ou l'étiquette (LABEL) associé à ce périphérique.
  - Ce premier champ peut aussi inclure des systèmes de fichiers distants, dans ce cas la notation d'un chemin NFS (Network File System) est utilisée : serveur:/chemin_distant.
  - Ceci indique que /chemin_distant est un répertoire partagé via NFS sur une machine distante dont le nom est « serveur » ;
- Le second champ indique le point de montage, qui est le chemin d'accès dans l'arborescence Linux ;
- Le troisième champ décrit le type de système de fichiers, par exemple ext2, ext3, reiserfs, iso9660, etc. ;

Format du fichier fichier fstab

- Le quatrième champ indique les options de montage.
  - Il s'agit d'une liste d'options séparées par des virgules, ce sont en fait les options de la commande mount.
  - Le mot clé « default » indique la combinaison des options rw, suid, dev, exec, auto, nouser, et async
  - (voir le manuel de la commande mount pour plus de détails sur chacune de ces options) ;
- Le cinquième champ est destiné théoriquement à être utilisé par l'utilitaire dump pour déterminer les systèmes de fichiers à sauvegarder. Mais en pratique ce champ n'est pas exploité et il est presque toujours à 0 ;

Format du fichier fichier fstab

- Le sixième champ est utilisé par l'utilitaire fsck pour déterminer l'ordre de vérification de l'intégrité des systèmes de fichiers lors du démarrage du système.
  - Le système de fichier racine doit avoir la valeur 1, les autres systèmes de fichiers ont la valeur 2 et seront vérifiés à la suite.
  - Si ce champ vaut 0, fsck ne vérifie pas le système de fichier.

Format du fichier fichier fstab

- Les commandes mount et umount utilisent le fichier fstab.
- Il est important que les données de ce fichier soient complètes et exactes.

Format du fichier fichier fstab

- Par exemple on peut monter un système de fichier en spécifiant seulement le point de montage ou le nom de périphérique.
  - Ainsi, au lieu d'exécuter la commande :

```
#mount –t iso9660 –o ro,noauto,owner, /dev/cdrom /media/cdrom
```

- on peut écrire

```
#mount /media/cdrom
```

  et les autres informations sont extraites automatiquement par la commande mount à partir de fichier fstab.

Format du fichier fichier fstab

- La commande mount -a monte tous les systèmes de fichiers répertoriés dans le fichier fstab.
- Cette commande est généralement exécutée au moment de démarrage du système.

Taux d'occupation de système de fichiers

- La commande df permet de connaître le taux d'utilisation de toutes les partitions montées du système.
  - L'option -h (human readable) facilite la lecture en utilisant des unités de taille plus commodes (Mo, Go, To …).
  - La commande du (disk usage) est très pratique pour connaître l'espace occupé par une arborescence.
  - L'option -s permet d'afficher le total pour chaque élément et l'option -k de l'afficher en kilo-octets:
    - du -ks /usr/local

Les droits sur les fichiers et les répertoires

Les droits sur les fichiers et les répertoires

- Objectifs
  - ⇒ Comprendre la gestion des droits sur les fichiers et répertoires sous Linux.
  - ⇒ Savoir protéger les fichiers et les répertoires.
- Mots clés
  - chmod, umask, chattr, suid, sgid, sticky bit, chattr

Les droits sur les fichiers et les répertoires

- Linux permet de spécifier les droits dont disposent les utilisateurs sur un fichier ou un répertoire par la commande chmod.
- On distingue trois catégories d'utilisateurs :
  - u : le propriétaire (user) ;
  - g : le groupe ;
  - o : les autres (others).
- Ainsi que trois types de droits :
  - r : lecture (read) ;
  - w : écriture (write) ;
  - x : exécution ;
  - - : aucun droit

Exemples de définition de droits d'accès

- pour donner le droit de lire le fichier liste.txt au propriétaire du fichier :
  - chmod u+r liste.txt
- pour autoriser une personne du groupe propriétaire du fichier à modifier le fichier :
  - chmod g+w liste.txt
- pour autoriser les autres utilisateurs à exécuter le fichier commande.sh :
  - chmod o+x commande.sh

Notation octale des droits d'accès

- Les droits d'accès peuvent aussi s'exprimer en notation octale.
- Les correspondances sont indiquées dans le tableau ci-dessous.

Droit	Notation Octale
---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
rw-	6
rwx	7

Notation octale des droits d'accès

- On peut utiliser la notation octale pour les droits avec la commande chmod,
- cela permet de positionner plusieurs types de droits en une seule commande.
- Exemples :
  - attribuer les droits rw------- à tous les fichiers :
    - chmod 600 *
  - attribuer les droits rw-r--r-- à tous les fichiers :
    - chmod 644 *
  - attribuer les droits rwxr-x--- à tous les fichiers :
    - chmod 750 *

Droits d'accès par défaut

- Lorsqu'on crée un nouveau fichier, par exemple avec la commande touch, ce fichier possède certains droits par défaut.
- La commande umask permet de changer ces droits par défaut.
- Les droits sont indiqués de façon « inverse » en notation octale pour les droits de type r et w seulement.
- Pour créer des fichiers en mode rw-r--r-- :
  - umask 022
- Pour créer des fichiers en mode ------- :
  - umask 666

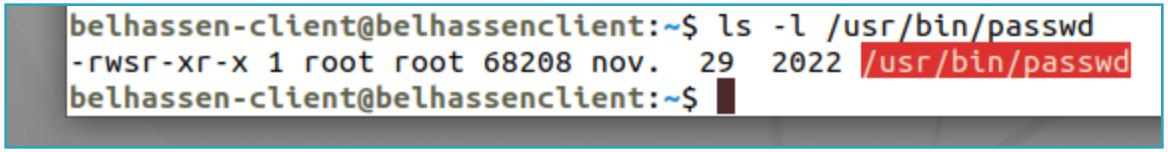
Les droits spéciaux

Les droits spéciaux

- Il existe trois droits spéciaux, SUID, SGID et Sticky Bit.
- Ils peuvent être positionnés par la commande chmod (premier groupe de droits exprimés en octal)

Le droit SUID (Set User ID)

- lorsque le bit SUID est positionné, une commande se lancera avec l'UID de son propriétaire, ce qui permet d'acquérir ses droits durant l'exécution de la commande :
- par exemple la commande /usr/bin/passwd permet d'acquérir les droits de root pour modifier le fichier /etc/shadow.



- Le bit SUID est positionné par l'option s de la commande chmod, ou bien en positionnant à « 1 » le premier bit du groupe des droits spéciaux (d'où la valeur 4 de l'exemple suivant) :
  - chmod 4755 /bin/cat
  - chmod u+s /bin/grep

Le droit SGID (Set Group Id)

- Ce droit fonctionne de la même façon que le droit SUID en ce qui concerne les exécutables mais en donnant le droit du groupe.
- Le SGID peut aussi être attribué à un répertoire : dans ce cas tout fichier créé dans un répertoire portant le SGID aura comme groupe propriétaire le groupe du répertoire.
- Ce bit est positionné par l'option s de la commande chmod, ou bien en positionnant à « 1 » le deuxième bit du groupe des droits spéciaux (d'où la valeur « 2 » de l'exemple suivant) :
  - chmod 2755 /home/lpi102
  - chmod g+s /home/lpi102

Le droit Sticky Bit

- si le Sticky Bit est positionné sur un répertoire, seul le propriétaire d'un fichier contenu dans ce répertoire pourra le renommer ou le supprimer, mais tous les utilisateurs pourront y avoir accès.
- Le Sticky Bit est par exemple toujours positionné sur les répertoires /tmp ou /var/mail/.

Le droit de Sticky Bit

- Pour les fichiers, en particulier les exécutables, le sticky bit permet à l'administrateur de marquer ces fichiers pour qu'ils restent conservés dans la mémoire principale, même lorsque leur exécution se termine, afin de minimiser le temps de leur prochain lancement.
  - Cette fonction est devenue obsolète en raison de l'optimisation du temps d'accès aux disques.

Le droit Sticky Bit

- Ce bit est positionné par l'option t de la commande chmod, ou bien en positionnant à 1 le troisième bit du groupe des droits spéciaux (d'où la valeur 1 de l'exemple) :
  - chmod 1666 /home/lpi/partagé
  - chmod o+t /home/lpi/partagé

Modifier le propriétaire et le groupe sur les fichiers et les répertoires

Modifier le propriétaire et le groupe sur les fichiers et les répertoires

- Objectifs

  ⇒ Être capable de modifier le propriétaire et le groupe d'un fichier ou d'un répertoire.
- Mots clés
  - chown, chgrp Linux permet de spécifier le propriétaire d'un fichier ou d'un répertoire par la commande chown:
  - chown le_propriétaire /home/lpi/monfichier

Modifier le propriétaire et le groupe sur les fichiers et les répertoires

- Linux permet de spécifier le propriétaire d'un fichier ou d'un répertoire par la commande chown.
  - chown le_propriétaire /home/lpi/monfichier
- Linux permet de spécifier le groupe d'un fichier ou d'un répertoire par la commande chgrp.
  - chgrp le_groupe /home/lpi/monfichier

Recherche de fichiers

Recherche de fichiers

- Objectifs

  ⇒ Connaître et maîtriser les différents types d'outils de recherche de fichiers.

  ⇒ Savoir gérer les bases de données d'informations associées à ces outils.
- Mots clés
  - find, locate, slocate, whereis, which, whatis, updatedb, makewhatis, apropos, /etc/updatedb.conf

Recherche de fichiers

- La recherche dans l'arborescence d'un système de fichiers peut se faire grâce à des utilitaires tel que find, locate, which, whereis, whatis et apropos.

find

- La commande find est la plus ancienne commande de recherche de Unix, elle n'utilise pas de base indexée et son exécution peut donc parfois être longue car elle est très complète par ses critères de recherche.
- La syntaxe générale est la suivante :
  - find <chemin> <critères>

find

- Les principales options de recherche sont les suivantes :
  - -name : par nom de fichiers ;
  - -type : par type du fichier (f : fichier, d : répertoire ...) ;
  - -user : utilisateur auquel appartiennent les fichiers recherchés ;
  - -atime : par date de dernier accès aux fichiers ;
  - -mtime : par date de dernière modification du contenu des fichiers ;
  - -ctime : par date de dernier changement des fichiers : contenu mais aussi droits d'accès, propriétaire..

find

- Pour rechercher le fichier Xinitrc dans tout le système (à partir de la racine) :
  - find / -name Xinitrc
- Pour rechercher les fichiers de l'utilisateur 666 dans tout le système (à partir de la racine) :
  - find / -user 666

locate et slocate

- La commande locate cherche tous les types de fichiers dans l'intégralité des systèmes de fichiers comme find, mais elle utilise une base de données.
- La base de données est automatiquement mise à jour par une commande de type cron, généralement la nuit, lorsque la machine est peu sollicitée.
- On peut mettre à jour manuellement la base de données en utilisant la commande updatedb (on doit être root pour lancer cette commande).
- Les options de fonctionnement de la commande updatedb sont décrites dans le fichier /etc/updatedb.conf.
  - On peut y décrire la racine de l'arborescence à indexer, les fichiers à exclure, l'emplacement de la base de données, etc.

locate et slocate

- La recherche est donc très rapide et peut se faire à partir de fragments du nom :
  - locate monfichier_perdu
    /home/nicolas/trucs/monfichier_perdu

locate et slocate

- La commande slocate est la version sécurisée de locate.
- Elle fonctionne de la même manière, mais stocke également les droits d'accès associés aux fichiers ainsi que les informations de propriété (propriétaire et groupe) du fichier de façon à ne pas afficher dans le résultat de la recherche les fichiers auxquels l'utilisateur n'aurait pas accès.

which

- La commande which est utilisée pour trouver l'emplacement d'une commande : elle effectue sa recherche par rapport au contenu de la variable PATH, et retourne le chemin du premier fichier correspondant.
  - which bash

  →Cette commande est très commode pour vérifier quelle version de la commande s'exécute réellement lorsqu'on l'appelle par son nom relatif.

whereis

- La commande whereis fonctionne de façon similaire à which, mais elle peut aussi chercher dans les pages de manuel (man) et les codes sources.

whatis

- La commande whatis cherche des commandes dans l'intégralité des systèmes de fichiers comme which, mais elle utilise une base de données qui contient une courte description ainsi que des mots clés.
- La base de données est créée en utilisant la commande makewhatis (on doit être root pour lancer cette commande).
  - makewhatis
- La recherche est donc plus rapide et peut se faire à partir du nom ou d'un mot clé.
- La réponse contient une description rapide de la commande
  - whatis who

  → who – show who is logged on

apropos

- La commande apropos utilise la même base de données que whatis, mais donne plus d'informations :
  - apropos who

→w – show who is logged on and what they are doing

→who – show who is logged on

→whoami – print effective userid