Gestion des processus

ps => afficher all.

ps -e => select all.

ps -ef => select all + columns (UID|PID|PPID|c|STIME|TTY|TIME|CMD)

pstree -p => arborescence.

pidof un_pros => find the pid (nom exact).

pgrep un_pros => find the pid.

kill -9 un_pid => tuer par pid.

killall name => tuer par nom.

./ un_pros & => lancer en bg.

jobs => lister les processus en bg.

fg num_job => reprendre en fg.

bg num_job => mettre en bg.

top => processus en temps réel.

Gestion des packages

Redhat : rpm / Debian : dpkgapt

rpm ihv pak.rpm => Installer un package (il doit exister sur la machine).

--nodeps => no relations de dépendances.

rpm -Uhv pak.rpm => update.

rmp -e pak => désinstaller.

rpm -qa => lister li installés.

rpm -q pak => installé ou non ?

rpm -qa | grep pak => chercher pak.

rpm -ql pak => fichiers de pak.

rpm -qc pak => fichiers de config de pak.

rpm -qd pak => fichiers de doc de pak.

rpm -qi pak => infos sur pak.

rpm -qip pak => same pour pak non installé.

rpm -qf cheminFichier => pak origine de Fichier.

rpm -qfi Fichier => infos sur pak origine de Fichier.

rpm -Va => vérifier les fichiers de tous les paks.

(M : permissions/types - 5 : contenu - S : taille - D : type majeur/mineur - L : nbr de liens - U : propriétaire - G : group - T : date d'accès - C : config).

rpm -V pak => vérifier les fichiers de pak.

rpm -Vf pak => vérifier le fichier et tout son pak.

yum install pak => installer un pak (needs Internet) dans une bdd locale (interrogée par -q).

Ubuntu :

apt-get install package

apt-get remove/purge package

dpkg -l => afficher les packages installés.

dpkg–i => install le ou les packages indiqué(s).

dpg–r => supprimer le ou les packages indiqué(s).

Plannification des tâches

crontab -e => éditer. || **crontab -r** => supprimer.

crontab -l => modifier.

/etc/cron.allow : users autorisés.

/etc/cron.deny : users non autorisés.

Format :

min h j_mois mois j_sem tâche

0-59 0-23 1-31 1-12 0-6

Réseau

ifconfig -a => infos de toutes les interfaces réseaux. || **ifconfig eth0** => infos sur eth0.

ifconfig eth0 up (ou ifup eth0) => activer eth0. || **ifconfig eth0 down (ou ifdown eth0)** => désactiver eth0.

ifconfig intf @ip => affecter @ip à l'inerface intf. || **ifconfig eth0 netmask 255.255.255.224** => affecter un mask à eth0.

dmesg | grep eth => infos sur port Ethernet de votre machine et son driver.

dhclient nom_intf => @ip @passerelle @DNS nom_domaine => configuration dynamique de l'interface.

ping @ip => envoyer packets et retourner temps de réponse. || **ping -C nbr @ip** => de même, mais envoyer just nbr packets. || **ping -b @broadcast** => pinger une @ broadcast. || Fichier **/etc/hosts** => on donne des noms aux @ip.

routeadd default gw @passerelle eth0 || ajouter dans yum si proxy : **proxy=http://10.23.201.11:3128**

Les permissions étendues

SUID (4000) : Lorsque le droit SUID est appliqué à un exécutable et qu'un utilisateur quelconque l'exécute, le programme détiendra alors les droits du propriétaire du fichier durant son exécution. Son flag est la lettre s ou S qui vient remplacer le x du propriétaire. C'est un s si le droit d'exécution du propriétaire est présent, ou un S sinon. Exemple : - r w s r - x r - x correspond à 4755 (= 4000 + 755). Commandes : **chmod u+s file || chmod u-s file || find / -perm +4000**

SGID (2000) :

Ce droit fonctionne comme le droit SUID, mais appliqué aux groupes. Il donne à un utilisateur les droits du groupe auquel appartient le propriétaire de l'exécutable et non plus les droits du propriétaire. Son flag est la lettre s ou S qui vient remplacer le x du groupe (la majuscule fonctionne aussi de la même façon). Exemple : - r w x r - S r - - correspond à 2744 (= 2000 + 744). Commandes : **chmod g+s file || chmod g-s file || find / -perm +2000**

STICKY BIT (1000) : Le bit collant

Le droit d'écriture signifie que l'on peut créer et supprimer les fichiers d'un répertoire. Le sticky bit permet de faire la différence entre les deux droits. Pour les répertoires : Il interdit la suppression d'un fichier qu'il contient à tout utilisateur autre que le propriétaire du fichier. Néanmoins, il est toujours possible pour un utilisateur possédant les droits d'écriture sur ce fichier de le modifier. La création de nouveaux fichiers est toujours possible pour tous les utilisateurs possédant le droit d'écriture sur ce répertoire. Pour les fichiers : Il indique alors que ce fichier doit encore rester en mémoire vive après son exécution. Son flag est le t ou T, qui vient remplacer le droit d'exécution x des "autres" (la majuscule fonctionne aussi de la même façon). Exemple : - r w x r - x r - t correspond à 1755 (= 1000 + 755). Commandes : **chmod o+t directory || chmod o-t directory || find / -perm +1000**

Système de fichiers LVM

Rôle : modifier la taille d'un système de fichier d'une façon dynamique.

Groupe de volume = ensemble de volumes physiques (disques/partitions) dans lequel on peut créer des partitions logiques (volume logique).

pvcreate fichier-spécial => créer un volume physique (**pvcreate /dev/sdb1**).

vgcreate nom liste_volumes => créer un groupe de volumes (**vgcreate data /dev/sdb1 /dev/sdc**).

lvcreate -n nom -L taile nom_volume => créer un volume logique (**lvcreate –n LV1 –L 10G data**, les 10G sont prises à partir du disque (/dev/data/LV1)).

* Pour augmenter la taille d'un LV, il faut :

Démonter la partition : **unmount /dev/data/LV1** ||| Check a linux ext2 file system : **e2fsck –f /dev/data/LV1**

Change the size : **lvresize –L +10G /dev/data/LV1** ||| Recalculer la taille: **resize2fs /dev/data/LV1**

vgextend data /dev/sda2 => augmenter la taille d'un groupe de volume.

vgdisplay –v /dev/data => afficher plus d'infos sur les groupes de volumes.

vgremove data /dev/data => supprimer le groupe de volumes.

Remplacer /dev/sda1 par /dev/sda2 dans le vg data :

#pvcreate /dev/sda2 #pvmove /dev/sda1 /dev/sda2 #vgextend data /dev/sda2 #vgremove data /dev/sda1

pvdisplay => afficher la liste des volumes physiques (Il faut éviter de mettre la partition /boot dans un VL).

* Sauvegarder les données des utilisateurs (/home) dans le fichier /users/data => **tar jcvf /users /data /home**

tar =>archiver toute l'arborescence dans un seul fichier => c : créer l'archive || v : mode bavard || f : dans un fichier || j : ajoute la compression Bzip2. || z : compression Gzip. || t : afficher le contenu || x : extrait l'archive

* Sauvegarder le premier de chaque mois les fichiers des utilisateurs et les mettre dans /users/data.

crontab –e 0 18 1 * * tar jcvf /users /data -'date +%b.%g' /home

* Créer un S.F ext2 dans une partition loop0 de taille 10M : **dd if=/dev/zero of=f bs=10M count=1 ||| losetup /dev/loop0 f ||| mkfs –t ext2 /dev/loop0 ||| mkdir /programmes_c ||| mount /dev /loop0/ programmes_c/**

* Déplacer les programmes c n'appartenant pas à root dans le SF

find / -name "*.c " ! -user root –exec mv{} /programmes_c/

* Désactiver le bit SUID dans ce SF : **mount –o remount,nosuid /dev/loop0**

* Donner la liste de tous les fichiers contenant password : **find / -type f exec grep -il password {} \ ;**

ACL : Access Control Lists

ACL provides an additional, more flexible permission mechanism for file systems.

mount -o remount,acl nom_partition => activer ACL pour la partition donnée.

setfacl -m u:user01:r-w file01 => Grant user01 read and write to file01. (g for group)

s -l file01 => afficher infos sur file01 et ses permissions d'accès.

getfacl file01 => afficher les ACLs définis pour ce fichier.

Si on des problèmes d'accès il faut revoir : Les permissions UNIX + Les attributs + Les ACLs.

mount => afficher les partitions montées avec les options de montage.

df –h => reports the amount of available disk space being used by file systems.

* Créer un fichier de taille 20 Mo.

dd if=/dev/zero of=file01 bs=20M count=1 || Les partitions /dev/loop[0-->7] sont virtuelles. **|| losetup /dev/loop0 file01 || losetup -a => show status of all loop devices. || losetup -d partition => detach the file or device associated with the specified loop device(s)**

mkfs -t ext2 /dev/loop0 => build a Linux filesystem on a device, usually a hard disk partition de type EXT2. || **mount /dev/loop0 /mwt**

Gestion des utilisateurs et des groupes

/etc/passwd : utilisateurs => nom_compte : mdp: UID : GID : comment : rep : prog_demarrage

/etc/group : groupes => nom_groupe : champ_special : GID : membre1, membre2

/etc/shadow : mdps => nom_compte : mdp : nbr_j_drn_chng : min_days: max_days : warn_days

/etc/login.defs : configuration.

UID = 0 => root || 0< UID < UID_MIN => propres à la machine || UID >= UID_MIN => utilisateurs.

grep UID_MIN /etc/login.defs || grep ^root /etc/passwd || echo ~ user_name (où suis-je ?)

cat -ns mon_fichier || cat *.cc || cat fichier1 fichier2 > fichier3 || cat > fichier (mode w) || cat >> fichier (mode a)

which command => chemin de command?

passwd username => changer mdp ("l" => désactiver compte, "u" => activer compte, "d" delete mdp).

PATH contient les répertoires dans lesquels le shell cherche la commande qu'on écrit au clavier. || **echo $PATH (afficher le contenu) || PATH = $PATH:./ (ajouter un rep)**

groupadd nom_groupe || gpasswd -a nom_user nom_groupe ||

gpasswd -M nom_user1,nom_user2, ... nom_groupe

gpasswd -d nom_user nom_groupe || groups user01 || groupdel nom_groupe (-r avec le répertoire personnel)

cut –d : -f1 /etc/passwd => Afficher les premiers champs des lignes.

grep ^root : /etc/passwd |cut –d : -f4 => Afficher le GID de l'utilisateur root.

cut -d : -f7 /etc/passwd|sort|uniq => Afficher les applications de démarrage.

chage -M MAX_DAYS username (-m MIN_DAYS || -W WARN_DAYS)

cut –d : -f1,3 /etc/passwd | grep :0$ |cut –d : -f1 => afficher admins.

awk -F/ '$3>=1000 { print $1 }' /etc/passwd => afficher users ordinaires.

cut –d : -f1,2 /etc/shadow| grep :$ |cut –d : -f1 => afficher users without mdp.

grep root file01 => lignes contenant root. (**grep -v** pour non)

useradd -u 0 admin => créer admin.

usermod -u new_uid username || usermod -l new_username old_username

grep ^username : /etc/passwd /etc/shadow /etc/group /etc/gshadow

cut –d : -f1,2 /etc/shadow| grep:\! |cut –d : -f1 => liste des comptes désactivés.

for x in ' cut –d : -f1,2 /etc/shadow| grep :$ |cut –d : -f1' ; do passwd -l $x ; done => Désactiver les comptes qui n'ont pas de mdp.

for x in {0..10..2} ; do echo bonjour $x times ; done

id -gn username => groupe principal de user (**-g** pour groupe secondaire).

gpasswd –M 'awk-F :' $3>=1000 {print $1}' /etc/passwd |tr "\n" "," ' groupname => Ajouter tous les comptes ordinaires à ce groupe.

grep friends : /etc/promp | ent –d : -f4 | tr ", " "\t" => Afficher la liste des membres.

usermod -g primarygroupname username => changer le groupe principal d'un utilisateur.

usermod -G secondarygroupname username => changer le groupe secondaire d'un utilisateur.

Système de fichiers ext2

/dev/hda : Le disque maître du 1er contrôleur ide.

/dev/hdb : Le disque esclave du 1er contrôleur ide.

/dev/hdc : Le disque maître du 2ème contrôleur ide.

/dev/hdd : Le disque esclave du 2ème contrôleur ide.

/dev/sda - /dev/sdb => disques plus récents.

ls -l /dev/sda

exécution : **8 => majeur / 0 => mineur.**

cat /dev/sda /strings

Sur un disque : **/dev/sdx**

/dev/sdx1 => 1ère partition primaire.

/dev/sdx4 => 4ème partition primaire.

/dev/sdx5 => 1ère partition logique.

On ne peut avoir qu'une seule partition étendue par disque.

fdisk -l => Afficher les partitions.

fdisk /dev/sdb

m => help / p => la table de partition / n => créer une partition / d => supprimer une partition / l => afficher les # types de système de fichiers / w => save / q => quit / t => change sf type.

Un système de fichiers ext2 divise une partition en :

1) Bloc de données : enregistrer les données de fichiers.

2) Super bloc : contient les informations sur le système de fichiers. (**tune2fs -l partition**).

3) inode (**ls -i file**) : contient les informations d'un fichier et les @ de ses blocs de données.

df => combien d'espace il nous reste ?

df -i => combien d'inodes il nous reste ?

mkfs -t type partition (exemple : **mkfs -t ext2 /dev/sdb1**)

Monter une partition = Associer sa racine avec le répertoire de montage.

mkdir /rep

mount /dev/sdb1 /rep

mount => Afficher toutes les partitions montées.

umount /rep => Démonter le répertoire.

remount : modifier les options d'un système de fichier monté.

Pour démonter un système de fichier, il ne doit pas être utilisé.

fuser rep_de_montage => PID des processus qui utilisent cette partition et comment ils les utilisent.

fuser -k rep_de_montage => Tuer tous les processus qui utilisent cette partition pour pouvoir la démonter.

Augmenter la sécurité du serveur Linux avec les options ,**nodev, nosuid et noexec** dans le fichier /etc/fstab

chattr +attribut file (attribut = "a" can only be open for writing, "i" ta7aja, "s" les données écrasées totalement si suppression, "S" modifications synchrones).

lsattr file => afficher les attributs.

find :

find « où » « quoi » « que faire avec »

find /var/log/ -name "*syslog*" || find -size +10M

find -name "*.jpg" -exec chmod 600 {} \; || find -name "*.odt" -atime -7

find / -user user01 -exec rm –rf {} || find ~ ! -user $USER

find / ! -user root -type f -size +1M 2>/dev/null

for x in 'awk -F : '$3>=1000 {print $1}' /etc/passwd'

do find ~ $x ! -user $x -exec rm –rf {}\ ; 2>/dev/null

Droits d'accès :

$ chmod 777 fichier

$ chmod g+w fichier || $ chmod o-r fichier || $ chmod u+rx fichier ||

$ chmod g+w,o-w fichier || $ chmod go-r fichier || $ chmod +x fichier ||

$ chmod u=rwx,g=r,o=- fichier