

Commandes labos Linux et Windows

Labo 1

Linux

Les groupes :

Créer un groupe : groupadd nom du groupe

Modifier un groupe : groupmod nom du groupe

Supprimer groupe : groupdel nom du groupe

Connaître tous les groupes de l'utilisateur : groups

Les users :

Créer un user : useradd -m -g [nom du groupe principal] -G [nom du ou des groupes secondaires] -c « commentaire sur le nom » nom du user

-m = créer un home directory pour l'utilisateur

-g = groupe principale

-G = groupe secondaire, ça peut être une liste de groupe séparé par une virgule

-c = commentaire

Supprimer un user : userdel nom du user

Changer information personnelle de l'utilisateur : chfn nom du user

Attribuer le shell bash à l'utilisateur : chsh -s /bin/bash nom du user

-s = shell

Attribuer password : passwd nom du user

Trouver nom utilisateur + groupes : id

Modifier user : usermod

Ajouter groupe à user : usermod -a -G [nom du ou des groupes secondaires] nom du user

-a = ajouter

-G = groupe secondaire, ça peut être une liste de groupe séparé par une virgule

Labo 2

Linux

Les dossiers :

Changer de dossier : cd nom du dossier

Lister les dossiers : ls

Afficher le chemin où on se trouve : pwd

Créer un fichier : touch nom du fichier

Créer un dossier : mkdir

Copier : cp chemin du fichier ou dossier

-r = copie du dossier en entier

Déplacer fichier ou dossier : mv chemin du fichier ou du dossier

Supprimer fichier : rm

-f = forcer la suppression

-i = demande une confirmation de suppression

-r = suppression récursive

Supprimer dossier : rmdir

Commande pour afficher l'arborescence d'un dossier :

sudo apt-get install tree

tree nom du dossier

Labo 3

Windows

Information sur l'ordinateur : Get-ComputerInfo

Afficher le nom du système : Get-ComputerInfo -Property CsName

Afficher la liste des verbes : Get-Verb

Afficher la liste des commandes : Get-Command

Les users :

Création d'un user sans mot de passe : New-LocalUser -Name « user name » -Description « Description du compte » -NoPassword

Création d'un user avec un mot de passe :

\$Password = Read-Host -AsSecureString

New-LocalUser -Name « user name » -Password \$Password -FullName « full user name » -Description « description du compte » -PasswordNeverExpires

Ajouter user à un groupe : Add-LocalGroupMember -Group « nom du groupe » -Member « nom du user »

Supprimer un user : Remove-LocalUser -Name « nom du user »

Ajouter date d'expiration à un user : Set-ADAccountExpiration -Identity nom du user
-DateTime « date »

Ajouter date d'expiration à tous les users d'un groupe spécifique :

Get-ADGroupMember -Identity « nom du user » | where {\$_.objectClass -eq "user"} | Set-ADAccountExpiration « date » [-TimeSpan 60.0 :0]

-TimeSpan 60.0 :0= 60 jours

Ajouter user à un groupe : Add-LocalGroupMember -Group « nom du groupe » -Member « nom du user »

-Member = On peut ajouter plusieurs users. Exemple : « user 1 », « user 2 », ... « user n »

Les groupes :

Créer un groupe : `New-LocalGroup -Name « nom du groupe »`

Lister les groupes : `Get-AdGroup -filter * | sort name | select name`

Lister membre d'un groupe : `Get-AdGroupMember -Identity nom du groupe`

Labo 4

Linux

Obtenir information plus précise sur un fichier : stat nom du fichier

Changer de propriétaire d'un fichier : chown [-R] nouveau-user fichier

-R = mode récursif pour agir sur l'ensemble des sous-répertoires

Changer le groupe propriétaire du fichier : chgrp [-R] nouveau-groupe fichier

Changer le propriétaire et le groupe du fichier en même temps :

chown nouveau-user :nouveau-groupe fichier

Changer droit d'accès :

Ajouter le droit d'exécution au propriétaire et au groupe (ajouter = +, retirer = -) :

chmod ug+x fichier

Autre manière de changer les droits d'accès : chmod u=rwx,g=rw,o=r fichiers

Autre manière de changer les droits d'accès : chmod binaire fichier

Attribuer les permissions par défaut lors de la création d'un fichier : umask binaire

Calculer permission effective :

Masque de protection = 027

777 = 111 111 111 permissions maxi = **rw**x **rw**x **rw**x

- 027 = 000 010 111 masque de protection

= 750 = 111 101 000 permissions effectives = **rw**x **r**-x ---

Bonus :

Script bash :

Les variables :

nom_var = valeur

Les conditions :

&& = et

|| = ou

If [condition1] || [condition2]

If [condition1] && [condition2]

if [condition]

then

Code à exécuter

elif [condition]

Code à exécuter

else

Code à exécuter

fi

Opération sur chaîne de caractères :

\$chaine1 = \$ chaine2 => Vérifie si les deux chaînes sont identiques. Notez que bash est sensible à la casse : « b » est donc différent de « B ».

\$ chaine1 != \$ chaine2 => Vérifie si les deux chaînes sont différentes.

-z \$ chaine => Vérifie si la chaîne est vide.

-n \$ chaine => Vérifie si la chaîne est non vide

Opération sur nombres :

\$num1 -eq \$num2 => Vérifie si les nombres sont égaux

\$num1 -ne \$num2 => Vérifie si les nombres sont différents

\$num1 -lt \$num2 => Vérifie si num1 est inférieur (<) à num2

\$num1 -le \$num2 => Vérifie si num1 est inférieur ou égal (<=) à num2

\$num1 -gt \$num2 => Vérifie si num1 est supérieur (>) à num2

\$num1 -ge \$num2 => Vérifie si num1 est supérieur ou égal (>=) à num2

Opération sur les fichiers :

-d \$nomFichier => Vérifie si le fichier existe.

-e \$nomFichier => Vérifie si le fichier est un répertoire

-f \$nomFichier => Vérifie si le fichier est un fichier

-L \$nomFichier => Vérifie si le fichier est un lien symbolique

-r \$nomFichier => Vérifie si le fichier est lisible (r).

-w \$nomFichier => Vérifie si le fichier est modifiable (w).

-x \$nomFichier => Vérifie si le fichier est exécutable (x).

\$fichier1 -nt \$fichier2 => Vérifie si fichier1 est plus récent que fichier2

\$fichier1 -ot \$fichier2 => Vérifie si fichier1 est plus vieux que fichier2

Paramètre

./fichier param1

if[\$1]

then

...

Le \$1 signifie le premier paramètre.

Les boucles :

While :

while[condition]

do

Code à exécuter

done

For :

for variable in « valeur 1 » « valeur 2 » ... « valeur n »

do

Code à exécuter

done

Exemple :

for fichier in `ls`

do

echo « Fichier trouvé : \$fichier »

done

On peut ajouter le nombre de répétition avec « seq 1 10 »

Les fonctions :

Création d'une fonction

maFonction()

{

Bloc d'instruction

}

#Appel d'une fonction

maFonction

Script batsh :

Début du fichier on écrit @echo off

Les variables :

Les strings

set nom_var = valeur

echo %nom_var%

#Les nombres

SET /A nomVar1 = 5

SET /A nomVar2 = 5

SET /A nomVar3 = %nomVar1% + %nomVar2%

echo %nomVar3%