

Linux Utilisation

Module 03 – Découverte du shell et premières commandes



1

Découverte du Shell et premières commandes



Objectifs

- Comprendre le fonctionnement d'une fenêtre du shell Bash
- Saisir quelques commandes d'informations
- Savoir changer son mot de passe
- Manipuler des variables
- Trouver de l'aide

Connaître le shell et les
commandes Linux.



2

Découverte du shell Bash



3

1 : Prompt

2 : Zone de saisie de commande par l'utilisateur

```
user30@LinuxUtil:~$
```

3 : Résultats

4 : Nouveau prompt



4

- Sous Linux et Unix, le compte **root** est le compte du **super administrateur**
- L'invite de commande root utilise le caractère **#**
- Un administrateur utilisera son compte habituel pour les opérations courantes puis effectuera ponctuellement une **élévation de privilèges** selon les besoins.

Nous n'utiliserons pas root dans ce cours.



Pour des raisons de sécurité il est **fortement déconseillé** de se connecter directement en root sur des systèmes Unix ou Linux

- Sur les distributions Debian Linux cela n'est nativement pas permis



```
Using username "root".
root@192.168.1.74's password:
Access denied
```

- Il faudra se connecter avec un compte utilisateur standard puis faire une **élévation de privilèges** en tant que root



Premières commandes



7

Premières commandes Conventions typographiques

- La syntaxe des commandes est représentée ainsi :
`Commande` `[options]` `<Argument>`
- Les exemples de commande et leur résultat sont représentés ainsi :

Rappel : L'espace est un séparateur d'expressions entre la commande, les options et les arguments.




```
user30@deb:~$ ls -l /home/user30
total 8
-rw-r--r-- 1 user30 user30  36 juin  26 12:08 aFaire.txt
-rw----- 1 user30 user30 586 avril 16 17:26 Edition
user30@deb:~$
```



8

logname

La commande logname affiche le nom de l'utilisateur ayant ouvert la session



```
user30@deb~:$ logname
user30
```

id [-ugGn] <utilisateur>

Afficher les informations de groupes, UID, GID




```
user30@deb~:$ id
uid=1060(user30) gid=1060(user30) groupes=1060(user30),1002(eni)
user30@deb~:$ id -u user05
uid=1035
```



who [-Hqu]

Afficher les utilisateurs connectés au système.



```
user30@deb~:$ who
user30    tty1          2020-06-26 10:26
user30    pts/0            2020-06-26 14:53 (192.168.1.35)
user29    pts/1            2020-06-26 15:13 (192.168.1.39)
```

who am i donne des informations sur l'utilisateur de la session courant



```
user30@deb~:$ who am i
user30    pts/0            2020-06-26 14:53 (192.168.1.35)
```



`passwd [options]`

Modifier le mot de passe utilisateur.

```
user30@deb~:$ passwd
```

Changement du mot de passe pour user30.

Current password:

Nouveau mot de passe :

Retapez le nouveau mot de passe :

passwd: password updated successfully

*D'autres options de passwd
font l'objet d'un autre cours
sur l'administration Linux*

-S Affiche des informations sur le mot de passe actuel

```
user30@deb~:$ passwd -S
```

```
user30 P 06/26/2020 0 99999 7 -1
```



Commandes d'information de l'utilisateur



Un peu plus de commandes



13

`date [options] <+format>`

date affiche des informations de date et d'heure du système.



```
user30@deb~:$ date  
vendredi 26 juin 2020, 15:52:06 (UTC+0200)
```

En contexte administrateur (root) `date` permet également de modifier date et heure du système d'exploitation.



14

- Formater la sortie de `date`

```
user30@deb~:$ date "+%nBonjour, nous sommes le %A
%d %B %Y.%nIl est %Hh%Mm"
```



```
Bonjour, nous sommes le lundi 22 juin 2020.
Il est 16h51m
```



La présence de quotes facilite l'utilisation du caractère espace



- Quelques options de format de `date`.

opt	Résultat	opt	Résultat
%a	Nom du jour abrégé (lun.)	%y	Année sur deux chiffres
%A	Nom du jour (lundi)	%Y	Année sur quatre chiffres
%b	Nom du mois abrégé (jan.)	%H	Heure de 0 à 23
%B	Nom du mois (janvier)	%I	Heure de 0 à 12
%d	Numéro du jour (0 – 31)	%M	Minutes
%D	Date au format anglo-saxon (mm/dd/yy)	%S	Secondes
%j	Numéro du jour dans l'année	%T	Heure sous forme HH:MM:SS
%w	N° du jour dans la semaine (0 = dimanche)	%n	Insertion d'un saut de ligne
%m	Numéro du mois	%t	Insertion d'une tabulation



`cal <mois> <année>`

Afficher le calendrier du mois et de l'année indiqué.



```
user30@deb:~$ cal
      Juin 2020
di lu ma me je ve sa
   1  2  3  4  5  6
  7  8  9 10 11 12 13
 14 15 16 17 18 19 20
 21 22 23 24 25 26 27
 28 29 30
```



`tty`

Afficher le nom du terminal utilisé par la session actuelle



```
user30@deb:~$ tty
/dev/tty1
```

En console



```
user30@deb:~$ tty
/dev/pts/2
```

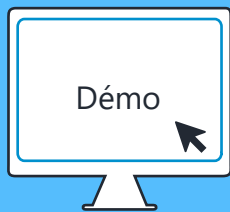
En connexion distante ou graphique



`pwd`

Afficher à quel endroit de l'arborescence on se trouve.

```
user30@deb:~$ pwd  
/home/user30
```



Quand et où



Découverte des variables



21

- Chaque **variable** porte un nom
- Ce nom permet de récupérer, modifier, le contenu de cette mémoire
- La donnée n'existe que dans la **mémoire vive** du système
- Pour lire une variable on accole un caractère **\$** avant le nom de la variable



```
var="Ceci est le contenu de ma variable"  
echo $var  
Ceci est le contenu de ma variable
```

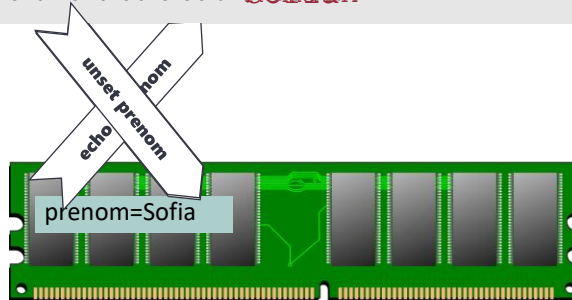


22

Etape 2



Libérer la mémoire une fois la variable inutile



23

env Visualisation des variables de l'utilisateur



```
user30@LinuxUtil:~$ env
SHELL=/bin/bash
PWD=/home/user30
LOGNAME=user30
HOME=/home/user30
LANG=fr_FR.UTF-8
```


set Visualisation de toutes les variables d'environnement et fonctions

- Cette commande renvoie de très nombreuses informations.
- Elle sera plus utile plus loin dans ce cours lorsque l'on aura appris à filtrer les résultats.



24

- Variables définis par le système
- Généralement en majuscules
- Utilisables par l'utilisateur



```
user30@deb:/tmp$ echo $LOGNAME
user30
user30@deb:/tmp$ cd $HOME
user30@deb:~$ pwd
/home/user30
```

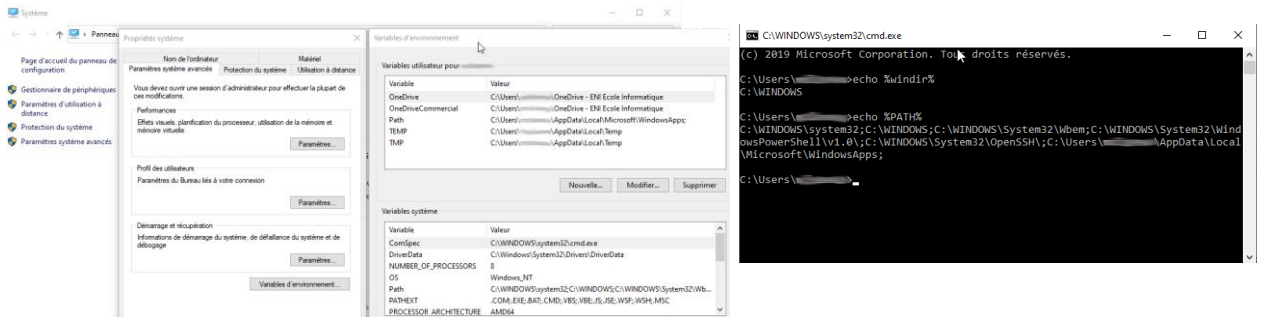


- Quelques variables d'environnement utiles

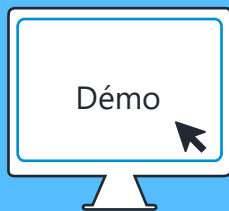
HOME	répertoire personnel de l'utilisateur courant
~	idem
HOSTNAME	nom d'hôte de la machine
LANG	Précise les options de régionalisations actives
LOGNAME	nom de l'utilisateur du shell actuel
PATH	chemin de recherche des exécutables (séparés par des ":")
PS1	formatage du prompt
PWD	Répertoire actuel
SHELL	Shell de connexion
USER	Nom de l'utilisateur actuel



- On peut faire une analogie avec Windows



27



Utilisation de variables

28

Se simplifier l'utilisation de la ligne de commande avec Bash



29

- Le shell Bash utilise un historique des commandes. Cette fonctionnalité apporte une grande efficacité à l'utilisateur.
- De manière basique l'on utilise les flèches haut et bas du clavier pour remonter ou redescendre dans ces commandes précédentes.
- **history** permet d'afficher le contenu de l'historique de commandes.



```
user30@deb:/$ history
1 ls -l
2 cat /etc/debian_version
3 touch /home/newfic01
4 ls -l
5 exit
```




30

- Il est possible d'augmenter encore son efficacité avec les raccourcis suivants
 - `!!` Rappeler la dernière commande exécutée
 - `!com` Rappeler la dernière commande commençant par "com"
 - `!?fich` Rappeler la dernière commande contenant la chaîne "fich"
 - `!287` Rappeler la commande numérotée 287 dans l'historique
- L'historique de commande est mémorisé dans le fichier `.bash_history` du répertoire personnel de l'utilisateur




- Bash utilise des variables d'environnement pour gérer l'historique
- On peut les visualiser afin d'obtenir des informations sur ces paramètres.

 `HISTFILE=/home/user30/.bash_history`
`HISTSIZE=1000`




En utilisant la touche [tab] du clavier le shell complétera de lui-même notre saisie. Il est possible de compléter

- des chemin d'accès



```
user30@deb:/$ cd /e [tab]  
user30@deb:/$ cd /etc
```


- des noms de commandes



```
user30@deb:~$ fir [tab]  
user30@deb:~$ firefox
```




- En des noms de variables



```
user30@deb:/$ echo $LOG [tab]  
user30@deb:/$ echo $LOGNAME
```

- Si plusieurs choix sont disponibles, un double appui sur la touche [tab] va permettre de lister les résultats possibles



```
user30@deb:/$ ls /etc/h [tab][tab]  
hdparm.conf host.conf hostname hosts hosts.allow hosts.deny
```



- Il existe plusieurs manière d'obtenir de l'aide à propos de notre utilisation de Linux.
- Internet
 - Réflexe aujourd'hui, internet possède de nombreuses ressources.
 - Sites web de communautés d'utilisateurs, tutoriels, vidéos, documentation technique des éditeurs, forums... les manières d'obtenir des informations foisonnent.
 - Attention toutefois il faut parfois savoir trier. Certaines informations peuvent-être obsolètes, voire fausse.



35

- Aide contextuelle des commandes
 - Aide plus ou moins succincte codée au sein même de la commande.
 - Permet de trouver rapidement une information sur l'utilisation générale de la commande ou sur les principales options.
 - La plupart des commandes utilisent l'option longue `--help` pour son appel.




```
user30@deb:/$ pwd --help
pwd: pwd [-LP]
Affiche le nom du répertoire de travail courant.
Options :
-L affiche la valeur de $PWD s'il nomme le répertoire de
travail courant
```



36

- **man** est la principale commande d'aide dans le monde Unix/Linux



```
man passwd
PASSWD(1)                                Commandes utilisateur                                PASSWD(1)

NOM
    passwd - Modifier le mot de passe d'un utilisateur


SYNOPSIS
    passwd [options] [LOGIN]

DESCRIPTION
    La commande passwd modifie les mots de passe des comptes d'utilisateurs. Un
    utilisateur normal ne peut changer que son propre mot de passe, alors que le
    superutilisateur peut changer le mot de passe associé à n'importe quel compte.
```

Pour obtenir l'aide de man : **man man**



- Le man possède des sections répondant à différents type d'informations.



```
man -S 5 passwd
PASSWD(5)                                Formats et conversions de fich                                PASSWD(5)

NOM
    passwd - fichier des mots de passe

DESCRIPTION
    /etc/passwd contient différentes informations sur les comptes utilisateurs.
    Ces informations consistent en sept champs séparés par des deux-points
```

- Les sections du man sont les suivantes :

- | | | |
|--------------------------|----------------------------|---------------------------|
| 1. Commandes utilisateur | 4. Fichiers spéciaux | 7. Divers |
| 2. Appels système | 5. Fichiers de paramétrage | 8. Administration système |
| 3. Appels bibliothèque | 6. Jeux | 9. Interface du noyau |





/mot

- Quelques commandes internes à man
 - `/mot` Rechercher « mot » dans la page
 - `n` Passer à l'occurrence suivante
 - `N` Passer à l'occurrence précédente
 - `h` Afficher l'aide interne de man
 - `q` Quitter man
- Utiliser le man en anglais
- `LANG=C man passwd`

```
PASSWD(1)          Commandes utilisateur          PASSWD(1)

NOM
    passwd - Modifier le mot de passe d'un utilisateur

SYNOPSIS
    passwd [options] [LOGIN]

DESCRIPTION
    La commande passwd modifie les mots de passe des
    comptes d'utilisateurs.
```

```
PASSWD(1)          User Commands          PASSWD(1)

NAME
    passwd - change user password

SYNOPSIS
    passwd [options] [LOGIN]

DESCRIPTION
    The passwd command changes passwords for user accounts.
```

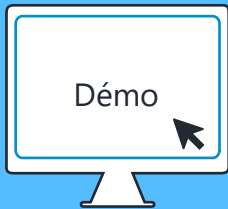


- Bash possède également la commande `help`. Celle-ci apporte de l'aide uniquement sur les commandes internes de ce shell.



```
user30@deb:/$ help
GNU bash, version 5.0.3(1)-release (x86_64-pc-linux-gnu)
Ces commandes de shell sont définies de manière interne.
Saisissez « help » pour voir cette liste.
Tapez « help nom » pour en savoir plus sur la fonction qui
s'appelle « nom ».
Utilisez « info bash » pour en savoir plus sur le shell en
général.
Utilisez « man -k » ou « info » pour en savoir plus sur les
commandes qui ne font pas partie de cette liste.
```





-
- Aide, historique et auto-complétion
-



TP



- Vous savez vous connecter à un système Linux
- Vous savez comment se constitue la ligne de commandes du Shell Bash
- Vous savez saisir une commande avec ses options et arguments sur la ligne de commande
- Vous savez utiliser quelques variables d'environnement

