

Linux Utilisation

Module 02 – Découverte du système d'exploitation Linux



1

Découverte du système d'exploitation Linux



Objectifs

- Décrire un système d'exploitation
- Découvrir la notion de Shell
- Se familiariser avec la syntaxe des commandes et les termes techniques liés
- Apprendre à utiliser les facilités offertes par Bash

*Savoir d'où vient Linux
Savoir ce qu'est un shell
et s'y connecter*



2

Présentation d'un système d'exploitation

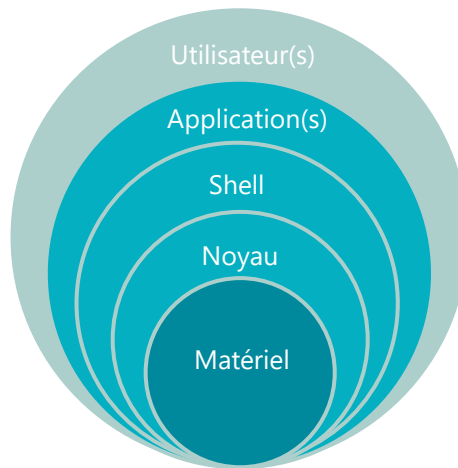


3

- Un système d'exploitation est un programme (ou ensemble de programmes) chargé de gérer les ressources matérielles d'un ordinateur et de les mettre à disposition des utilisateurs.
- Quelques exemples :
 - Système propriétaire
 - OS400 (IBM)
 - Système ouvert mono poste, mono tâche (micro)
 - DOS
 - Système ouvert multi-poste, multi-tâche, multi-plateforme
 - UNIX, GNU/Linux, Windows, OS-X



4



- Le **noyau Unix/Linux** est le programme principal du système.
- Il a pour rôle de :
 - Gérer les ressources (périphériques, mémoire ...)
 - Ordonnancer les différentes tâches (processus)
 - Mettre ces ressources à la disposition des utilisateurs
- Dans un sens plus large, on entend souvent par **noyau Linux**, le **noyau** lui-même ainsi que l'ensemble des **programmes essentiels** au système.
- On utilise également le mot anglais **kernel**.



Découverte du Shell



7

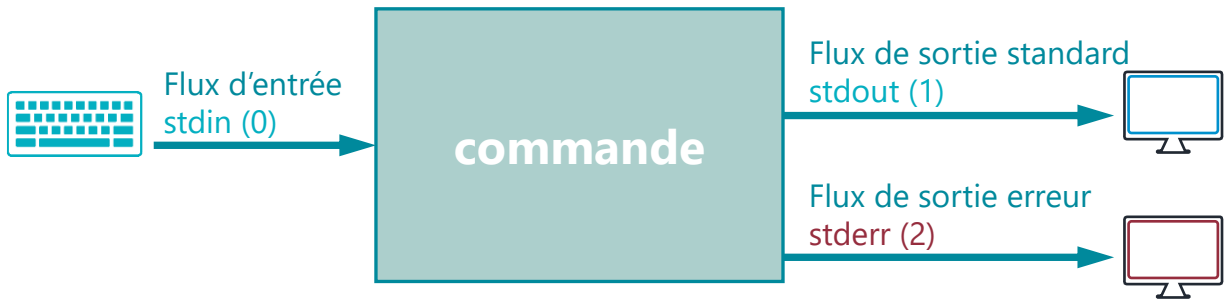
- Le **shell** est l'interface par laquelle l'utilisateur va interagir avec le système d'exploitation et les programmes disponibles.
- Le shell est un **interpréteur de commandes**.
- On parle également de CLI (Command Line Interface) en anglais, ligne de commande en français.



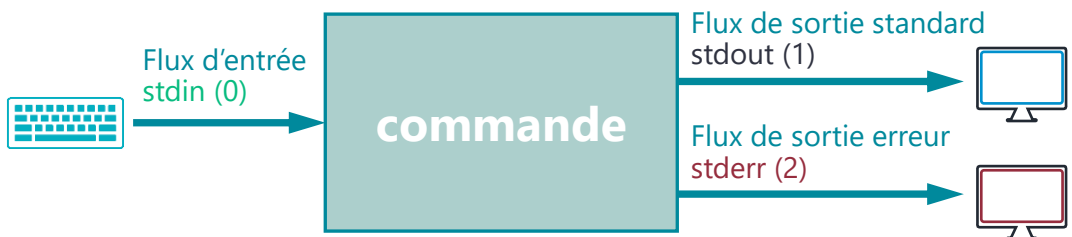
```
user30@deb:~$ echo "Bonjour le monde"  
Bonjour le monde
```




8



9



 user30@deb:~\$ ls Bonjour <- stdin
cat: Bonjour: Aucun fichier ou dossier de ce type <- stderr



10


- Bourne shell : **sh**
 - écrit par Steve Bourne
 - Apparaît dans Unix v7
- C Shell : **csch / tcsh**
 - Écrit par Bill Joy
 - Inspiré du langage C
- KornShell : **ksh**
 - Écrit par David Korn
 - Compatible Bourne Shell, il reprend des éléments du C shell



- Bo(u)rn(e) Again Shell : **bash**
- Shell principal des distributions GNU/Linux
- Réécriture de Bourne par Brian Fox au sein du projet GNU
- Implémentation libre de bourne shell
- Quelques améliorations de Bash par rapport à Bourne
 - Historique de commandes
 - Autocomplétion
 - Alias de commande



- L'utilisation de la ligne de commande impose le respect de certaines règles, sans lesquelles il n'est pas possible d'utiliser correctement le système.
- Un shell Linux est sensible à la casse




```
user30@deb:~$ echo "Bonjour le monde"
Bonjour le monde
user30@deb:~$ Echo "Bonjour le monde"
-bash: Echo : commande introuvable
```

- On commence une ligne de commande par le nom de la commande souhaitée.



- La **commande** doit toujours être située en premier dans l'instruction après le prompt.



```
user30@LinuxUtil:~$ ls -l /home/user30
```

- La **commande** peut être suivie par des **options**, puis des **arguments**, le tout séparé par des espaces.



`who`

Commande simple sans argument

`ls /etc`

Commande avec un argument

`head -n 5 /etc/pam.conf`

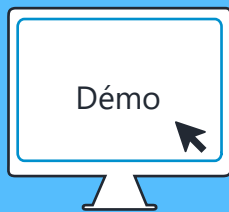
Commande avec une option et un argument

`cp /tmp/fic1 $HOME`

Commande avec deux arguments

`tar xzvf archive.tar.gz`

Commande avec quatre options (sans signe + ou -) et un argument d'option



Quelques exemples de commande



Se connecter à un système Linux



17

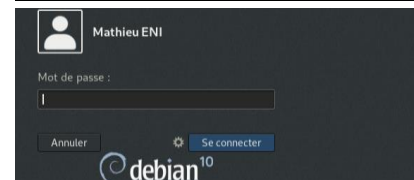
Se connecter à un système Linux
Les éléments requis

- Pour se connecter à un système Linux, un utilisateur doit disposer d'un **identifiant** auquel est associé un **mot de passe**.

- Connexion à la console

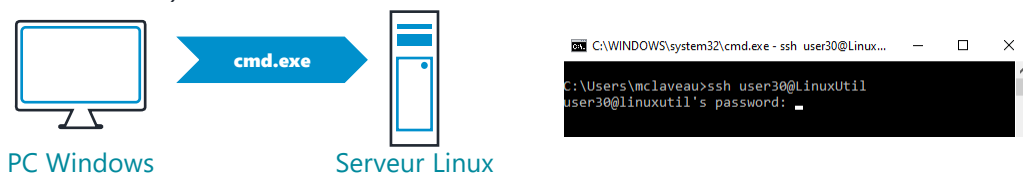
```
Debian GNU/Linux 10 LinuxUtil tty1
LinuxUtil login: user30
Password:
Last login: Fri Jun 26 10:25:41 CEST 2020 on tty1
Linux LinuxUtil 4.19.0-8-amd64 #1 SMP Debian 4.19.
Bienvenue sur Linux Utilisation
user30@LinuxUtil:~$
```

- Connexion à une interface graphique

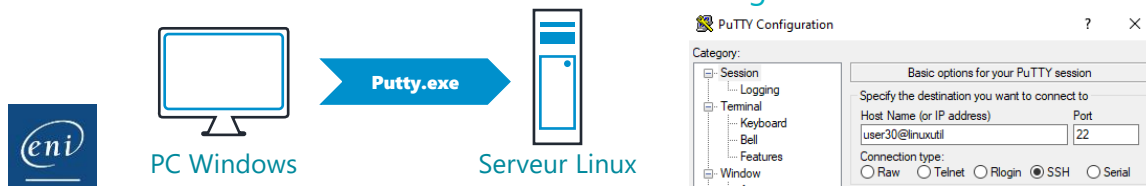


18


- Connexion distante en **ssh** à travers une fenêtre cmd (Windows récent).





- Connexion distante en **ssh** avec un **logiciel** dédié.




- **Putty** : Le plus connu des logiciels de connexion en **ssh**
- Open Source (Licence MIT)
- Modules de copie de fichiers **pscp** et **psftp**

 <https://https://www.chiark.greenend.org.uk/~sgtatham/putty/>

- Kitty : « Fork » de Putty  <https://kitty9bis.com/>
- Lancement automatique de commandes
- mRemoteNG : Plusieurs connexions actives dans plusieurs onglets
- Multi-protocole (Linux, Windows, Citrix, VNC...)

 <https://mremoteng.org/>

- MobaXterm : Logiciel propriétaire proche de mRemoteNG
- Multi-protocole (Linux, Windows, VNC...)

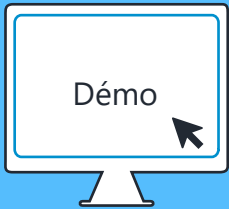
 <https://mobaxterm.mobatek.net/>



- Dernière étape essentielle : savoir se déconnecter. Plusieurs méthodes sont possibles et variables selon le shell utilisé.
- `exit` : commande la plus utilisée. Compatible avec la quasi-totalité des shells, à commencer par Bash.
- `logout` : autre commande disponible pour Bash (mais pas Bourne)
- `^D` : [ctrl] + D , raccourci clavier permettant de fermer une session




Se connecter à un système Linux




Démo

-
- Paramétrage de Putty et connexion à la VM Atelier
-




28

Se connecter à un système Linux



TP

-
- Connexion à la VM Atelier
-



29

- Vous savez de quoi est composé un système d'exploitation
- Vous savez ce qu'est un shell
- Vous savez vous connecter et vous déconnecter d'un système Unix/Linux

