



# TP 2 Linux

## Les commandes de base sous Shell



# Plan du cours

1. Présentation su Shell
2. Invite de commande
3. Présentation des commandes Shell
4. Les fichiers sous Shell
5. Manipulation des commandes de base sous Shell



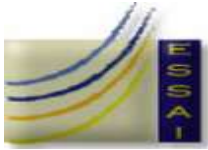
# Présentation du Shell

- Shell un nom anglais qui signifie «coquille».
- Le shell est chargé de faire l'intermédiaire entre le système d'exploitation et l'utilisateur grâce aux lignes de commandes saisies par ce dernier. Son rôle consiste ainsi à lire la ligne de commande, interpréter sa signification, exécuter la commande, puis retourner le résultat sur les sorties.
- Il existe plusieurs shells possédant chacun des spécifications qui lui sont propres (compétence, interactivité).
- Le choix de shell dépend du système installé ou de préférence de l'utilisateur.



# Présentation du Shell

- Voici une liste non exhaustive des différentes Shells:
  - **Bash(Bourne Again Shell):** est utilisé par défaut sur la plupart des distributions (nous allons l'utiliser dans le reste de nos TPS)
  - **Csh (C shell),** développé par l'université de Berkeley, utilisé par les systèmes BSD
  - **Sh,** le shell original développé par Steve Bourne



# Présentation du Shell

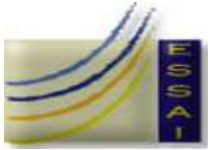
- Pour avoir une idée les différents Shells installés sur votre système, taper la commande

**cat /etc/shells**

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ cat /etc/shells  
# /etc/shells: valid login shells  
/bin/sh  
/bin/dash  
/bin/bash  
/bin/rbash  
administrateur@administrateur-VirtualBox:~$
```

les différents Shells installés sous mon Ubuntu

commande Shell permet d'afficher le contenu des fichiers



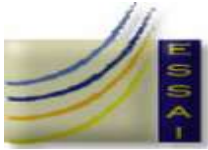
# Invite de commande

- Lorsque vous ouvrez le terminal, la première chose que vous voyez est l'invite de commande
- Avec Bash Shell, l'invite de commande par défaut prend la forme:

machine:/repertoire/courant\$

- «\$» indique qu'il s'agit d'un utilisateur normal
- «#» indique qu'il s'agit de l'administrateur, appelé «root»





# Passer en super-utilisateur

- Si vous voulez vous connecter entant que super utilisateur ou (root), taper la commande

**sudo su**

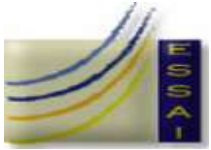
```
Termin... Fichier Édition Affichage Rechercher Terminal Aide
root@admistrateur-VirtualBox: /home/admistrateur
admistrateur@admistrateur-VirtualBox:~$ sudo su
[sudo] password for admistrateur:
root@admistrateur-VirtualBox: /home/admistrateur#
```

1. Taper cette commande et valider par entrée

2. Shell vous demande de fournir le mot de passe de l'utilisateur

3. maintenant vous êtes connectés entant que super utilisateur

dans ce cas \$ est remplacé par #



# Présentation des commandes Shell

- La forme des commandes Shell
  - Les commandes Shell sont de la forme générale suivante:  
**nom\_de\_commande [options] [arguments]**
  - Une commande peut n'avoir ni option ni argument : **ls**
  - Une commande peut être suivie par une ou plusieurs options:  
**ls -a -l ou ls -al** (on peut regrouper les options)
  - Si l'option est un mot, on doit la précéder par **--** :

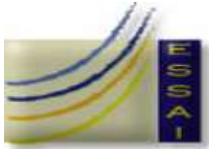
**ls --all**





# Présentation des commandes Shell

- Une commande peut avoir un ou plusieurs arguments: **ls /dev**
- Une commande peut être suivie d'une combinaison d'options et d'arguments: **ls -al /dev**
- Remarque: faites attention Shell respecte la casse donc **ls** n'est pas **LS**



# Présentation des commandes Shell

- Retrouvez les commandes

## Shell

- Si vous voulez chercher une commande, taper les premières lettres puis appuyer deux fois sur la touche tab. Une liste des commandes s'affiche





# Présentation des commandes Shell

- l'historique des commandes
  - appuyez sur la flèche directionnelle Haut ; vous verrez apparaître la dernière commande que vous avez tapée.  
Si vous appuyez de nouveau sur la flèche directionnelle Haut, vous verrez l'avant-dernière commande, puis l'avant-avant-dernière



Flèche directionnelle Haut

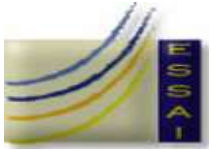


# Présentation des commandes Shell

- avoir l'aide sur une commande
- Utilisation de la commande man

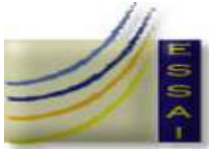
Exemple: **man ls**

- Voici quelques commandes pour défiler ou sortir des pages de man :



# Présentation des commandes Shell

PgUP	Page précédente
PgDwn	Page suivante
Entrer	Ligne suivante
space	Page suivante
b	Page précédente
q	Quitter l'aide



# Présentation des commandes Shell

- Utilisation de l'option `--help`

Exemple: **ls --help**

- cette option nous évite de fastidieuse recherche sur l'utilisation d'une commande et elle nous fournit une aide simplifiée qui ne contient que les options les plus utilisées pour une commande donnée.

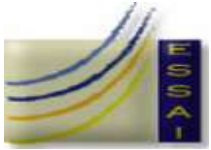


# Présentation des commandes Shell

- Utilisation de la commande info

**info <nom\_de\_commande>**

- Cette commande donne des informations plus précises et, parfois, agrémentées d'exemples
- Vous pouvez vous déplacer dans les pages avec les touches de direction. Passez aux pages suivantes avec la touche **n**, aux pages précédentes avec la touche **p** et enfin, vous pouvez quitter en utilisant la touche **q**.



# Présentation des commandes Shell

- Quelques raccourcis sur les commandes

## Shell

- Pour interrompre une commande, on peut utiliser la combinaison **ctrl+C**
- Pour se déconnecter du terminal, on peut utiliser encore la combinaison **ctrl +D**





# Les fichiers sous Shell

- Les différentes catégories de fichiers
  - fichiers normaux
    - \* texte : courrier, sources des programmes, scripts, configuration ...
    - \* exécutables : programmes en code binaire
  - fichiers répertoires
    - ce sont des fichiers conteneurs qui contiennent des références à d'autres fichiers.



# Les fichiers sous Shell

- fichiers spéciaux
  - situés dans /dev, ce sont les points d'accès préparés par le système aux périphériques.
- fichiers liens symboliques
  - Ce sont des fichiers qui ne contiennent qu'une référence (un pointeur) à un autre fichier. Cela permet d'utiliser un même fichier sous plusieurs noms sans avoir à le dupliquer sur le disque.



## Les fichiers sous Shell

- Pour savoir le type du fichier sous Shell, tapez la commande suivante: **file nom\_fichier**
- Voici un exemple d'utilisation de cette commande

```
Termin Fichier Édition Affichage Rechercher Terminal Aide
administrateur@administrateur-VirtualBox: ~
administrateur@administrateur-VirtualBox:~$ file /home
/home: directory
administrateur@administrateur-VirtualBox:~$
```



# Les fichiers sous Shell

- Nomenclatures des fichiers sous Linux
  - Les caractères autorisés
    - La plupart des caractères (chiffres, lettres, majuscules, minuscules, certains signes, caractères accentués) sont acceptés, y compris l'espace (très déconseillé).
    - Cependant quelques caractères sont à éviter : **& ; ( ) ~ <espace> \ | ` ? - (en début de nom)**



# Les fichiers sous Shell

- La notion d'extension
  - UNIX et Linux ne proposent aucune gestion des extensions.  
L'utilisateur est libre de se servir du caractère point autant qu'il désire. Ce signe n'ayant aucun sens particulier.
  - Ainsi le nom suivant est tout à fait correct: **cours.unix.2004.**
  - si le système Linux ou UNIX n'exploite pas la notion d'extension, en revanche certains logiciels l'utilisent. C'est le cas du compilateur C, qui attribue l'extension « .c » pour les programme source et « .o » aux programme objet.



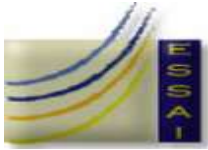
# Les fichiers sous Shell

- Les caractères jokers
  - Certains caractères jouent un rôle particulier pour l'interpréteur de commandes « shell ». ces caractères appelés jokers ou caractères de substitutions.
  - Ces caractères peuvent être placés n'importe où dans le nom du fichier



# Les fichiers sous Shell

- Ces caractères sont:
  - \*: remplace un nombre de caractères (y compris aucun)
  - ?: remplace un caractère et un seul
  - []: se comportent comme le point d'interrogation. Mais la substitution ne s'effectue qu'avec l'un des caractères présents entre les crochets
  - Exemple:



# Les fichiers sous Shell

[abc]	Le caractère de remplacement est la a, le b ou le c
[0..9]	Le caractère de remplacement est chiffre compris entre 0 et 9
[!abc]	Remplace par un caractère différent de a, b et c
[!0..9]	Remplace par un caractère autre que les chiffres entre 0 et 9





# Les chemins d'accès sous Shell

- Les chemins d'accès
  - Dans les chemins d'accès aux fichiers, les noms de répertoires et de fichiers sont séparés par un caractère spécial.
  - Ce caractère est traditionnellement, sous Linux, la barre oblique de division (nommée « slash » en anglais) : « / ».



# Les chemins d'accès sous Shell

- Les utilisateurs du DOS et de Windows prendront garde ici au fait que Microsoft a préféré la barre oblique inverse (nommée « backslash » en anglais) « \ »
- L'exemple suivant vous montre l'allure d'un chemin d'accès typique sous Linux :

**/home/dupond.jean/lettres/professionnelles/marketing/ventes2005.odt**



# Les chemins d'accès sous Shell

- Les chemins relatif et absolu
  - Chemin absolu d'un fichier est son emplacement pris à partir de la racine « / ». La séparation entre les niveau se fait par des « / »

## Exemple:

**`/home/kmaster/DOCS/LINUX/cours.sdw`**

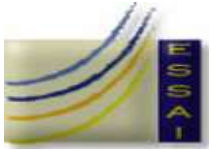
- Chemin relatif d'un fichier est son emplacement pris à partir du répertoire courant.

**Exemple : `DOCS/LINUX/cours.sdw`**



# Des commandes Shell d'ordre général

- date : affiche la date et l'heure système
- who : indique les utilisateurs connectés au système
- cal : affichage d'un calendrier
- Clear : effacer l'écran
- Exit: se déconnecter du terminal

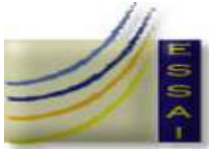


# Des commandes Shell d'ordre général

- history: affiche les commandes précédemment lancées par l'utilisateur courant.
- halt, qui permet d'arrêter le système (on doit être un super utilisateur)
- reboot, qui permet de redémarrer le système (on doit être un super utilisateur).
- Echo: affiche la ligne de texte passée en paramètre

syntaxe : echo "chaîne"

options    echo -n chaîne : n'ajoute pas de retour à la ligne automatique  
à la fin



# Des commandes Shell d'ordre général



- Les commandes relatives aux fichiers

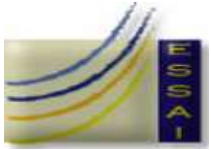
- **which**: connaître l'emplacement d'une commande

Exemple: \$**which pwd**

- **pwd**: permet d'afficher le chemin absolu du répertoire courant

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ pwd  
/home/administrateur  
administrateur@administrateur-VirtualBox:~$
```

Chemin d'accès absolu du répertoire courant

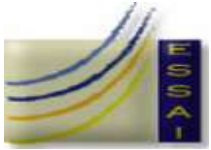


# Des commandes Shell liées aux fichiers et répertoires

- Créer un répertoire
  - Pour créer un répertoire, la commande à utiliser est la suivante : `mkdir <nom_de_repertoire>+`
  - où `<nom_de_repertoire>+` est une liste non vide de noms de répertoire séparés par un ou des espace(s).

Exemple : **\$ mkdir toto** (création d'un seul rep)

**\$ mkdir lulu B212** (création de 2 reps)



## Des commandes Shell liées aux fichiers et répertoires

- Avec l'option `-p` la commande `mkdir` peut créer une arborescence

Exemple: **`$ mkdir -p`**  
**`document/texte/perso`**





# Des commandes Shell liées aux fichiers et répertoires

- Afficher le contenu du répertoire
  - Pour lister les fichiers d'un répertoire, vous pouvez utiliser la commande `ls`. Sa syntaxe est la suivante :

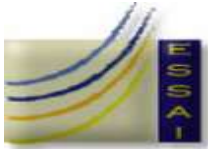
**`ls [<options>] [<fichiers>]`**

- Si le nom de `<fichiers>` n'est pas spécifié, la commande liste les fichiers du répertoire courant.



# Des commandes Shell liées aux fichiers et répertoires

- Quelques options de la commande ls:
  - -a ou --all : Liste tous les fichiers, y compris ceux cachés (avec un nom commençant par .) ;
  - -B : Ne liste pas les backups-fichiers (se terminant par ~) ;
  - -c : Trie par date. Agrémenté de l'option -l c'est plus clair ;
  - -C : Liste par colonne ;



# Des commandes Shell liées aux fichiers et répertoires

- Pour savoir les caractéristiques d'un fichier quelconque, taper : `ls -l nom_fichier`

```
Terminal
administrateur@administrateur-VirtualBox: ~
administrateur@administrateur-VirtualBox:~$ ls -l /home
total 4
drwxr-xr-x 21 admistrateur admistrateur 4096 janv. 17 07:33 admistrateur
administrateur@administrateur-VirtualBox:~$
```

type du fichier

Compteur du lien

Les droits d'accès

l'utilisateur et le groupe propriétaires

Taille fichier

Date dernière modif

nom du fichier

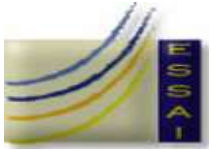


# Des commandes Shell liées aux fichiers et répertoires

- Changer de répertoires:

Exemple: **cd nom\_repertoire**

- Dans le cas où le nom de répertoire est absent, la commande cd vous dirige vers votre répertoire utilisateur



# Des commandes Shell liées aux fichiers et répertoires

- Astuces:

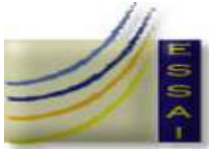
**cd ..** Permet de passer dans le répertoire parent

**cd ~** déplacement dans son répertoire personnel

**cd ~jeans** déplacement dans le répertoire personnel de jeans

**cd /** déplacement vers la racine

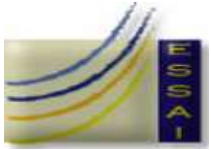
**cd -** Revient dans le répertoire précédent le dernier changement de répertoire



# Des commandes Shell liées aux fichiers et répertoires



- **Connaître l'espace disque occupé par le répertoire**
  - La commande `du` permet de connaître l'utilisation disque par le répertoire spécifié et ses sous répertoires .Exemple: `du /home`
  - `du` : une commande qui possède des paramètres, telles que:
    - `-h`: pour afficher des capacités compréhensibles par les humains.
    - `-c`: afficher le total d'espace occupé par le répertoire
    - `-a`: afficher l'espace occupé par chaque fichier et chaque sous répertoire existant sous le répertoire en question

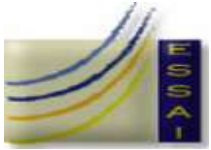


# Des commandes Shell liées aux fichiers et répertoires

- **Supprimer un répertoire**
- Pour supprimer un répertoire, la commande à utiliser est la suivante :  
**rmdir <nom\_de\_repertoire>+**
- où <nom\_de\_repertoire>+ est une liste non vide de noms de répertoire séparés par un ou des espace(s).
- Si le répertoire est non vide la commande affiche un message d'erreur.

Exemple :        :        \$ **rmdir MonRep**

Remarque: La commande **rm -r** permet de supprimer toute une arborescence.



# Des commandes Shell liées aux fichiers et répertoires

- **Création des fichiers**
  - pour créer un fichier, on a plusieurs possibilités. Parmi elles, on peut utiliser la commande touch

Exemple: **\$touch fichier**

- Remarque:
  - Si on veut créer plusieurs fichiers avec la même commande, il faut séparer ces noms par des espaces.

Exemple: **touch fichier1 fichier2 ... fichiern**





# Des commandes Shell liées aux fichiers et répertoires

- **Copier des fichiers**
- pour copier des fichiers, on utilise la commande `cp`

Exemple 1 : dupliquer un fichier dans le même répertoire

**`cp fichiersource fichiercopie`**

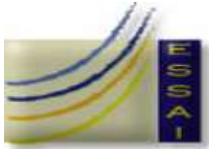
Exemple 2 : Copier un fichier dans un autre dossier

**`cp fichiersource dossier/`**

Exemple 3 : Copier un fichier dans un autre dossier avec un autre nom : **`cp`**

**`fichiersource dossier/fichierdestination`**

Exemple 4 : Copier des dossiers : **`cp -R dossier1 dossier2`**



# Des commandes Shell liées aux fichiers et répertoires

- Remarque: on peut utiliser les caractères jockers avec la commande cp

Exemple: **cp \*.jpeg mondossier/**

- **Déplacer/renommer des fichiers:**
  - la commande mv a deux utilités:
    - Le déplacement des fichiers
    - Renom mage des fichiers
  - Exemple1: Déplacer un fichier du répertoire source vers un répertoire destination: **mv fichiersource dossier/**



# Des commandes Shell liées aux fichiers et répertoires

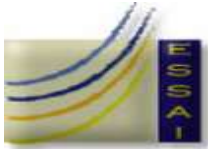
- Exemple 2 : Déplacer un répertoire vers un autre répertoire

**mv dossier dossierconteneur**

- Remarque: On peut utiliser des caractères jokers avec la commande mv.
- Exemple 3: renommage du fichier: **mv source destination**
- Exemple 4: déplacement et renommage des fichiers

**mv source dossier/destination**

Le graphique suivant, comment peut on déplacer et renommer en même temps un fichier ou un répertoire.



# Des commandes Shell liées aux fichiers et répertoires

```
Terminal
administrateur@administrateur-VirtualBox: ~
administrateur@administrateur-VirtualBox:~$ mv t1 rep2/t3
```

cette commande va déplacer le fichier t1 vers le répertoire rep2  
et elle va aussi changer son nom qui devient t3

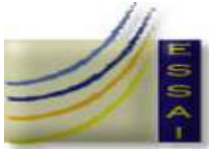


# Des commandes Shell liées aux fichiers et répertoires

- L'option `-i` : s'applique sur les commande `cp` et `mv` , elle sert à afficher un message de confirmation.
- **Afficher un fichier:**
  - Les commandes sont: `cat`, `more` ou `less`

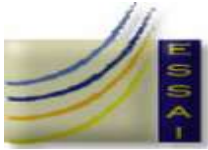
Exemple: **\$ cat t1.text**

- Une option utile de `cat` est `-n` qui permet de numéroter les lignes (ne pas oublier que `cat` permet de lire et non de modifier un fichier. Ainsi la numérotation de ligne apparaît à l'écran mais le fichier `t1.text` n'en est pas pour autant modifié).



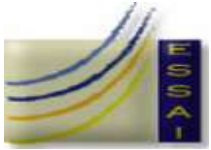
## Des commandes Shell liées aux fichiers et répertoires

- Vous pouvez utiliser la commande **more** pour visualiser un fichier. La commande more a l'avantage d'afficher le fichier **page par page**. Pour passer d'une page à l'autre, tapez sur la touche **ESPACE**.
- **less** a le même fonctionnement que more, mais elle possède plus d'options. Voici qu'elles unes



# Des commandes Shell liées aux fichiers et répertoires

Touche	Effet
Espace	Affiche la suite du fichier. La touche Espace fait défiler le fichier vers le bas d'un "écran" de console. C'est celle que j'utilise le plus souvent. Vous pouvez aussi utiliser la touche "Page Down".
Entrée	Affiche la ligne suivante. Cela permet donc de faire défiler le fichier vers le bas ligne par ligne. Vous pouvez aussi utiliser la touche "Flèche vers le bas".
d	Affiche les 11 lignes suivantes (soit une moitié d'écran). C'est un peu l'intermédiaire entre Espace (tout un écran) et Entrée (une seule ligne).
b	Retourne en arrière d'un écran. Vous pouvez aussi appuyer sur la touche "Page Up".
y	Retourne d'une ligne en arrière. Vous pouvez aussi appuyer sur la touche "Flèche vers le haut".
u	Retourne en arrière d'une moitié d'écran (11 lignes).
q	Arrête la lecture du fichier. Cela met fin à la commande less.



# Des commandes Shell liées aux fichiers et répertoires

- Supprimer un fichier:
- La commande `rm` (remove) supprime un ou plusieurs fichiers, et éventuellement une arborescence complète, suivant les options. La suppression est définitive.

**`rm [Options] fic1 [fic2...]`**

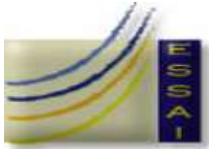
- Options :
  - `-d` ou `--directory` : Efface les répertoires, même si ils ne sont pas vides (en super-user seulement) ;
  - `-i` : la commande demandera une confirmation pour chacun des fichiers à supprimer.





# Des commandes Shell liées aux fichiers et répertoires

- **Afficher le début du fichier**
  - La commande head ("tête" en anglais) affiche seulement les premières lignes du fichier. Elle ne permet pas de se déplacer dans le fichier comme less, elle permet juste de récupérer les premières lignes.
- **Afficher la fin du fichier**
  - La commande tail affiche seulement les dernières lignes du fichier. Elle ne permet pas de se déplacer



# Des commandes Shell liées aux fichiers et répertoires

- Remarque: ces deux commandes possèdent une option `-n` qui sert à préciser le nombre de lignes à afficher au début ou à la fin du fichier.

**Exemple 1: afficher 3 lignes au début du fichier**

```
mateo21@mateo21-desktop:/var/log$ head -n 3 syslog
```

**Exemple 2: afficher 3 lignes à la fin du fichier**

```
mateo21@mateo21-desktop:/var/log$ tail -n 3 syslog
```



# Merci pour votre attention

Contact: [sihemmansour@yahoo.fr](mailto:sihemmansour@yahoo.fr)