

TP	COMMANDÉ LA VACHE QUI PARLE !	
	SYSTEME D'EXPLOITATION - LINUX	
	Nom, Prénom :	Date :

## PROBLEMATIQUE

Malgré la présence d'une interface graphique conviviale, Linux permet aussi d'utiliser des lignes de commandes en mode texte dans un Terminal.

Cette activité va vous permettre de vous familiariser avec les commandes de bases du système Linux via une application faisant parler une vache.

## OBJECTIFS

Être capable d'utiliser les commandes de gestion du système de fichier Linux (lister les moyens d'obtenir une aide sur une commande (man commande, l'option –help ou – h ou internet).

Être capable de créer sa propre commande avec du script Shell.

## CONCEPTS

### *Le terminal*

Malgré la présence d'une interface graphique conviviale, Linux permet aussi d'utiliser des lignes de commandes en mode texte pour interagir avec le système d'exploitation, appelées le Terminal. La gestion des commandes est gérée par le shell.

Le terminal permet d'aller beaucoup plus vite que l'interface graphique qui se résume en une succession, parfois fastidieuse, de clics. De plus, les interfaces graphiques ne permettent toujours pas, même à l'heure actuelle, de réaliser toutes les tâches faisables avec un shell.

### *Le shell*

Le shell est l'environnement utilisateur en mode texte sous Linux. C'est le programme qui se charge de lire et d'exécuter les commandes que l'utilisateur saisit. Classiquement, le shell est utilisé de manière interactive, c'est-à-dire que l'utilisateur dialogue avec le système par l'intermédiaire du shell. Il saisit les commandes, et le shell les exécute et affiche les résultats. Le shell le plus couramment utilisé sous Linux est bash.

En tout cas, c'est le shell par défaut que la plupart des distributions utilisent. Il est donc conseillé de connaître un petit peu ce que ce shell est capable de réaliser, et comment. Le shell bash est une évolution du shell sh, utilisé par quasiment tous les systèmes Unix. Son nom provient de l'abréviation de l'anglais « Bourne Again SHell », ce qui signifie qu'il s'agit effectivement d'une nouvelle variante du shell sh.

Quoi qu'il en soit, le shell est bien plus qu'un interpréteur de commande. Il s'agit réellement d'un environnement de programmation, permettant de définir des variables, des fonctions, des instructions complexes et des programmes complets, que l'on appelle des scripts shell.

## MATERIEL

Ordinateur avec accès Web : <https://www.webminal.org>

Cliquez sur « Login », connectez-vous et cliquer sur « Terminal » et ressaisir les identifiants :

**Login :** DUESNSI

**Password :** Nekochan

## AIDE

### REGARDER LE DOSSIER OU VOUS ETES :

#### LA COMMANDE BASH « PWD » – PRINT WORKING DIRECTORY

Lorsqu'on démarre le Terminal, on se situe automatiquement dans un dossier précis de notre ordinateur. Normalement, ce dossier devrait être le dossier "home" (i.e le dossier portant votre nom d'utilisateur) par défaut. Les différentes commandes qu'on va taper vont pouvoir produire des effets différents en fonction de l'endroit où on se situe. Il est donc essentiel de savoir où on se trouve.

```
[DUESNSI@webminal.org ~]$pwd  
/home/DUESNSI
```

### COMMENT SE DEPLACER DANS L'ARBORESCENCE :

#### LA COMMANDE « CD » – CHANGE DIRECTORY

Pour changer de répertoire, c'est-à-dire pour se déplacer dans votre ordinateur, vous allez utiliser la commande `cd` pour "change directory".

Vous devez savoir qu'on utilise le caractère slash « / » pour représenter un sous-dossier et le caractère « .. » pour se déplacer vers un dossier parent.

```
[DUESNSI@webminal.org ~]cd ../../
```

### QUE FAIT LA COMMANDE LS ?

Cette commande affiche la liste des fichiers et dossiers présents dans le répertoire :

```
[DUESNSI@webminal.org ~]$whatis ls ou man -f ls  
ls (1)           - list directory contents
```

### COMMENT AFFICHER DE L'AIDE SUR UNE COMMANDE ?

#### AFFICHER LE MANUEL D'UNE COMMANDE :

```
[DUESNSI@webminal.org ~]$man « nom de la commande »
```

#### AFFICHER L'AIDE D'UNE COMMANDE :

```
[DUESNSI@webminal.org ~]$« nom de la commande » --help
```

### QUESTION :

Est-ce que « `$ls -als` » est équivalent à « `$ls -lsa` » ou « `$ls -a -l -s` » ?

Vrai  Faux

TP	COMMANDER LA VACHE QUI PARLE !	
	SYSTEME D'EXPLOITATION - LINUX	
	Nom, Prénom :	Date :

## ACTIVITE 1 – DECOUVERTE / INDIVIDUELLE

**Problématique :** A ce stade de l'étude, vous ne savez pas quelle est la commande permettant de faire parler la vache. Le but est de vérifier si elle est présente sur votre système.

**Objectif :** Comment s'exécute une commande ? ses arguments ? options ? et paramètres ? usage de « pwd », « ls », « cd » et « man ».

Lorsqu'on ouvre le Terminal, une fenêtre noire ou blanche s'ouvre. Cette fenêtre contient une invite de commande, c'est-à-dire une ligne nous indiquant que le shell attend qu'on lui passe des commandes :

```
[DUESNSI@webminal.org ~] $
```

Par défaut, cette ligne contient le nom de votre machine suivi du caractère tilde (~) suivi du signe dollar (\$).

Le tilde est une abréviation pour indiquer qu'on se situe actuellement dans le dossier "home", c'est-à-dire dans le répertoire de base lié à notre nom d'utilisateur.

- Quelle commande permet de vérifier où vous êtes dans l'arborescence des fichiers ?

Le signe dollar indique qu'on est connecté en tant qu'utilisateurs classiques avec des privilèges normaux. Lorsque vous êtes connecté en tant que super utilisateur, le dernier signe de votre invite est # plutôt que \$.

Une commande à la particularité de pouvoir être appelée où que vous soyez dans l'arborescence. Pour cela, le fichier correspondant à la commande doit être placé dans un des dossiers de la variable environnement \$PATH. Pour visualiser cette variable taper dans le terminal :

```
[DUESNSI@webminal.org ~] $echo $PATH
```

- Combien de chemins différents renvoie cette variable ?

- Quelle commande proposez-vous pour visualiser le contenu dans « /bin » ?

- Avec l'aide de la commande « man ls », expliquer ce que renvoi la commande suivante ?

```
[DUESNSI@webminal.org ~] $ls -ls /bin/cow*
```

Signification des options :

Signification des paramètres :

---

## ACTIVITE 2 - FAIRE PARLER LA VACHE : PAR EQUIPE DE TROIS-REINVESTISSEMENT DE L'ACT.1

**Problématique :** le but ici est de mieux maîtriser la commande « la vache qui parle ».

**Objectif :** Comment s'exécute une commande ? ses arguments ? options ? et paramètres ? usage de « ls », « cd » et « man ».

- 
1. Quelle commande utiliser pour obtenir la réponse suivante :

```
[DUESNSI@webminal.org ~]$
```

```
<Hello !...>
-----
\ ^ ^
\ (oo) \
( ) \_____
| |---w |
| |
```

2. Quelle commande utiliser pour obtenir la réponse suivante :

```
[DUESNSI@webminal.org ~]$
```

Utiliser le manuel afin d'utiliser les bonnes options.

```
(Urghhh ... )
-----
\ ^ ^
\ (xx) \
( ) \_____
U | |---w |
| |
```

3. Expliquez la commande suivante :

```
[DUESNSI@webminal.org ~]$cowthink -d "Urghhh ..."
```

TP	COMMANDER LA VACHE QUI PARLE !	
	SYSTEME D'EXPLOITATION - LINUX	
	Nom, Prénom :	Date :

### ACTIVITE 3 – CHANGER DE PEAU ! CREER UNE COMMANDE

La commande suivante donne la liste des skins disponibles :

```
[DUESNSI@webminal.org ~]$cowsay -l
```

Par exemple :

```
[DUESNSI@webminal.org ~]$cowsay -f koala "What?"
```

Créer votre propre commande du type « votre skin »say tel que : « koalasay »

### AIDE

- ✓ Sous Webminal, utilisez Bash devant votre commande pour l'exécuter.
- ✓ Utilisez nano pour éditer
- ✓ La première ligne comporte :
  - `#!/bin/bash`
    - Cela indique au shell quel program prendre pour exécuter votre script.
- ✓ Récupérer les arguments avec \$
  - `$n` récupère le nième argument fournit...

---

#### ACTIVITE 4 – VOIR TOUTES LES PEAUX ! CREER UNE COMMANDE-REINVESTISSEMENT ACT.3

Vous allez créer un script pour afficher l'ensemble des skins disponibles.

**Rappel :** la commande « cowsay -l » donne la liste des skins disponibles.

Créer votre propre commande « Skinsay » (ou « Bimsay ») permettant de faire défiler vos skins en acceptant un texte.

---

#### AIDE

- ✓ Utilisez nano pour éditer
- ✓ La première ligne comporte :
  - `#!/bin/bash`
    - Cela indique au shell quel program prendre pour exécuter votre script.
- ✓ Récupérer les arguments avec \$
  - \$n récupère le nième argument fournit...
- ✓ Voir structure d'un fichier « .cow » dans « /usr/share/cowsay »
- ✓ Structure d'un IF : IF / then / else / elif/then / FI
- ✓ Structure d'une boucle FOR : FOR / DO / DONE
- ✓ Chmod +x « votre fichier » pour le rendre exécutables
- ✓ Utiliser la commande mv + « paramètres » pour copier vers un dossier de \$PATH.
- ✓ Sous Webminal, utilisez Bash devant votre commande pour l'exécuter.