







**Linux**  
**Le petit grand frère**  
**de UNIX**

DUT Informatique – Semestre 1  
Ressource R1.04  
Responsable : Jean-François ANNE

**JFA - 63**

02/10/2023

## Mise à jour du system Linux

Il faut mettre régulièrement son système linux à jour pour corriger des bugs, ajouter des logiciels, supprimer les logiciels obsolètes, ...

Pour cela on a une commande unique : **apt**

En effet **apt** permet d'installer des logiciels sur les distributions à base de Debian depuis les dépôts (repository).

Quelques rappels concernant les packages de distribution.

Un mainteneur s'occupe de maintenir un package qui permet d'installer un logiciel sur la distribution.

Il s'agit d'un fichier **.deb** que le mainteneur crée et maintient.

Ce fichier permet d'exécuter des scripts afin de placer les fichiers, de configuration et de l'application, dans Linux.

En clair donc d'installer les paquets et applications dans Linux.

**JFA - 64**

## Les dépôts

Les **dépôts** (repositories en anglais) sont des serveurs mis en ligne qui stockent tous les **.deb** d'une distribution.

Ainsi, **apt** se connecte à l'un de ses serveurs afin de télécharger le **.deb** et l'installer. On appelle cela les sources.

Pour ce fait, **apt** appelle la commande **dpkg**.

**JFA - 65**

Un dépôt peut stocker plus de 20 000 fichiers **.deb**.

Enfin **apt** peut aussi mettre à jour une distribution Linux.

Cela consiste donc à télécharger tous les **.deb** de la nouvelle version et les installer.

Les logiciels systèmes comme udev, libc ou encore le noyau linux sont aussi mis à jour par cette méthode.

Du côté des distributions Redhat, l'équivalent est **yum**.

Son fonctionnement est assez proche.

02/10/2023

## Les fichiers de configuration de apt

Afin de fonctionner **apt** utilise plusieurs fichiers et dossiers.

Voici une liste de ces derniers :

- ✓ **/etc/apt/sources.list** : stocke les sources avec l'adresse des dépôts
- ✓ **/etc/apt/sources.list.d/** : sources additionnels. Ainsi on peut ajouter des dépôts non officielles.
- ✓ **/etc/apt/apt.conf** : fichier de configuration APT
- ✓ **/etc/apt/apt.conf.d/** : fichiers de configuration additionnels.
- ✓ **/etc/apt/preferences.d/** : Les fichiers de préférences additionnels.
- ✓ **/var/cache/apt/archives/** : Stocke les packages (fichiers **.deb**)
- ✓ **/var/lib/apt/lists/** : stocke la liste et informations sur les packages du systèmes.

**JFA - 66**

le dossier **/var/cache/apt/archives** contient les deb déjà téléchargés.

Cela évite en cas de réinstallation un nouveau téléchargement.

Sinon la commande **apt clean** permet de vider ce dossier.

02/10/2023

<https://www.malekal.com/apt-installer-mise-a-jour-paquet-distribution-debian-ubuntu-mint/>

## Les commandes apt :

**Apt** regroupe différentes commandes selon les besoins.

Comme vous modifiez la configuration système, ces opérations nécessitent des droits **root**.

Ainsi sur Debian, il faut s'identifier en **root** ou utiliser **sudo** pour les distributions basées sur ce dernier.

❑ **sudo apt update**

JFA - 67

**apt update** permet de mettre à jour l'**indexation du dépôt** sur votre Linux.

En effet, lorsque votre indexation est trop ancienne, il n'est plus synchronisé avec celui en ligne. Ainsi, vous pouvez demander une version qui n'existe plus et obtenir une erreur.

Enfin on utilise **apt update** lorsque l'on modifie les sources afin de télécharger les nouveaux index.

Cette commande resynchronise votre indexation.

02/10/2023

<https://www.malekal.com/apt-installer-mise-a-jour-paquet-distribution-debian-ubuntu-mint/>

## Les commandes apt :

❑ **sudo apt upgrade**

Cette commande met à jour les paquets de la distribution Linux de votre machine.

Cela ne change pas la version de la distribution mais met à jour les paquets.

En effet, des mises à jour de sécurité sont publiées chaque jour.

On lance la mise à jour des paquets avec la commande suivante :

**sudo apt upgrade**

JFA - 68

La liste des mises à jour s'affiche.

Ici celle-ci peut être très longue si la dernière mise à jour remonte à très longtemps.

Puis validez par o (ou y)

puis les paquets se téléchargent.

La vitesse et le délai s'affichent en bas à droite de l'écran.

Enfin la phase d'installation s'effectue.

**apt** peut poser des questions sur des actions à effectuer durant la mise à jour.

02/10/2023

<https://www.malekal.com/apt-installer-mise-a-jour-paquet-distribution-debian-ubuntu-mint/>

## Les commandes apt :

❑ **sudo apt autoremove**

❑ **sudo apt autoclean**

Ces commandes permettent de supprimer les fichiers qui ne sont anciens ou plus nécessaires.

**autoclean** : Tout comme clean, autoclean nettoie le référentiel local des paquets récupérés. La différence est qu'il supprime uniquement les paquets qui ne peuvent plus être téléchargés et qui sont inutiles.

JFA - 69

**autoremove** : apt-get supprime les paquets installés dans le but de satisfaire les dépendances d'autres paquets et qui ne sont plus nécessaires.

❑ **sudo apt dist\_upgrade**

Enfin la commande **apt dist-upgrade** permet la mise à niveau de la distribution.

Vous passez donc à la version suivante de votre distribution.

Ensuite on lance **apt upgrade** ou encore **apt-full upgrade** pour finir la mise à jour,

02/10/2023

<https://www.malekal.com/apt-installer-mise-a-jour-paquet-distribution-debian-ubuntu-mint/>

## Qu'est-ce qu'un Shell Unix/Linux ?

Un Shell Unix est un interpréteur de commandes destiné aux systèmes d'exploitation Unix et de type Unix qui permet d'accéder aux fonctionnalités internes du système d'exploitation. Il se présente sous la forme d'une interface en ligne de commande accessible depuis la console ou un terminal. L'utilisateur lance des commandes sous forme d'une entrée texte exécutée ensuite par le shell. Dans les différents systèmes d'exploitation Microsoft Windows, le programme analogue est command.com, ou cmd.exe.

JFA - 70

Il existe deux environnements très différents sous Unix :

- l'environnement console
- l'environnement graphique

Et il existe plusieurs environnements console : les Shells.

02/10/2023

## Les Shells Unix/Linux

Tous les systèmes d'exploitation de type Unix disposent le plus souvent d'un shell. À l'origine, l'interpréteur de commandes par défaut était sh, qui donna naissance à de nombreuses variantes, dont csh, étendu en tcsh, ou ksh, ou encore rc... Mais aujourd'hui bash, s'inspirant de sh, ksh, et csh, est le shell le plus répandu, bien qu'il existe d'autres interpréteurs de commandes, comme zsh, ou ash.

Voici les noms de quelques-uns des principaux Shells :

JFA - 71

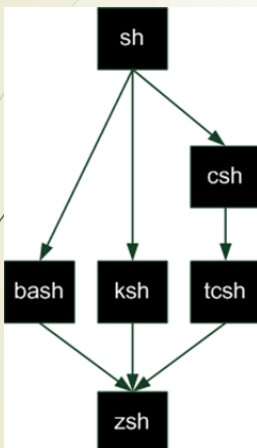
- **sh** : Bourne Shell. L'ancêtre de tous les Shells. Sa syntaxe des commandes est proche de celle des premiers UNIX
- **Bash** : Bourne Again Shell. Une amélioration du Bourne Shell augmenté de la plupart des fonctionnalités avancées du C shell, un script Bourne shell sera correctement interprété avec un Bash, disponible par défaut sous GNU/Linux et Mac OS X.
- **ksh** : Korn Shell. Un Shell puissant assez présent sur les Unix propriétaires, mais aussi disponible en version libre, compatible avec Bash.
- **csh** : C Shell. Un Shell utilisant une syntaxe proche du langage C.

02/10/2023

## Les Shells Unix/Linux

- **tcsh** : Tenex C Shell. C'est une extension et Amélioration du C shell d'origine.
- **zsh** : Z Shell. Shell assez récent reprenant les meilleures idées de Bash, ksh et tcsh.

JFA - 72



```

jfa@jfa-VirtualBox: ~
jfa@jfa-VirtualBox:~$ cd
jfa@jfa-VirtualBox:~$ pwd
/home/jfa
jfa@jfa-VirtualBox:~$ █
  
```

02/10/2023

Bash est le Shell par défaut. Cependant, sh reste toujours plus répandu que Bash.



## Login

Le premier concept important à garder en mémoire avant de travailler avec Linux est qu'il s'agit d'un système **multiutilisateur**. L'accès à la machine Linux doit donc être **contrôlé** par un compte utilisateur (login, mot de passe) créé par l'administrateur du système.

Chaque usager est connu dans le système par son nom d'utilisateur et son identificateur unique (un nombre entier) qui lui sont attribués par l'administrateur du système.

JFA - 73

Il est également affecté à un **groupe** (associé lui-même à un identificateur de groupe unique). Il existe un utilisateur particulier qui dispose de tous les droits : **root** ou super utilisateur. Le groupe sert généralement à grouper des utilisateurs qui ont un centre d'intérêt commun (projet, facturation) et à en gérer efficacement tous les membres.

La console a toujours un fond noir et un texte blanc(par défaut) . En revanche, les fonctionnalités offertes par l'invite de commandes peuvent varier en fonction du Shell que l'on utilise.

Beaucoup de solutions sont données en ligne de commande, non pas qu'Unix n'ait pas d'interface graphique, mais pour certaines tâches, **l'utilisation de la ligne de commande s'avère bien plus pratique et plus puissante que la souris.**

## Démarrer Linux :

### ☐ Connexion :

**Compte = nom de connexion ou nom d'utilisateur + mot de passe.**

**Login : Jean**

```
root@jfa-VirtualBox:/home/jfa#
```

**Password : \*\*\*\*\***

**Bienvenue sur ...**

```
jfa@jfa-VirtualBox:~$
```

**Prompt>**

JFA - 74

« \$ » (ou « % », « # ») est le prompt ou l'invite de l'interprète de commande utilisé (*shell*) .

**L'interprète attend que l'utilisateur tape une commande, exécute cette commande, réaffiche la chaîne d'invite, et attend une nouvelle commande ...**

**^D (CTRL D) en début de ligne => déconnexion (fin du shell) ; ce caractère simule une fin de fichier.**

**^<caractère> correspond à « contrôle caractère ».**

**Il existe les comptes utilisateurs et le compte super-utilisateur (root). L'invite de ce dernier est en général « # » ; ce compte permet d'accéder à tous les fichiers et d'exécuter toutes les manipulations systèmes (administration).**

## Démarrer Linux :

### ❑ Modification du mot de passe :

C'est la commande **passwd** :

```
prompt> passwd
Changing password for Jean
Old password : ****
New password : *****
Re-enter new passwd : *****
prompt>
```

JFA - 75

### ❑ A propos du mot de passe :

**Un bon mot de passe doit être difficile à trouver ! :**

**Il faut mélanger des chiffres, des lettres, des majuscules/minuscules, mettre des caractères spéciaux, et qu'il ait suffisamment de symboles (plus de 8) !**

**Sous linux si vous avez oublié votre mot de passe utilisateur, l'administrateur (root) peut vous le changer.**

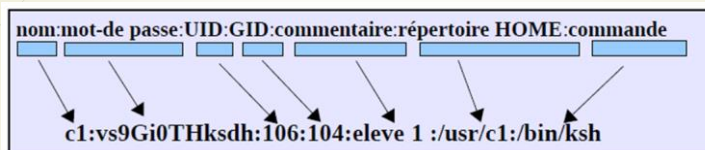
**Si c'est le mot de passe root que vous avez oublié, tout est perdu ! Il faut réinstaller le système !**

02/10/2023

## Fichiers stratégiques : /etc/passwd

❑ Ce fichier rassemble des informations sur tous les utilisateurs ayant un compte sur le système.

○ Une ligne par utilisateur et 7 champs par ligne, séparés par le caractère ':'



JFA - 76

➤ **Nom** : nom de l'utilisateur (connexion) ; 8 caractères au plus (minuscules)

➤ **Mot de passe** : soit rempli et crypté (sur 13 caractères) par le programme passwd. Soit contient « x » avec un mot de passe crypté et déporté dans un fichier accessible par l'administrateur

➤ **UID** : **[User Identification]** c'est le numéro d'identification de l'utilisateur (0 pour root).

➤ **GID** : **[Group Identification]** c'est le numéro d'identification du groupe auquel appartient l'utilisateur.

➤ **Commentaire** : Champ facultatif.

➤ **Répertoire HOME** : Répertoire d'accueil lors de la connexion de l'utilisateur.

➤ **Commande** : Commande lancée au moment de la connexion.

02/10/2023

## Fichiers stratégiques : Environnement shell

Lorsque l'utilisateur se connecte au système sur un terminal texte, plusieurs fichiers sont lus au lancement du shell pour définir l'environnement de travail.

### 1. /etc/profile

Le fichier /etc/profile est un script shell qui est exécuté en premier lors de la connexion à un terminal texte. Ce fichier contient les variables d'environnement de base de tous les processus, et seul l'administrateur système peut le modifier. En outre, ce fichier exécute des commandes dans l'environnement du shell de connexion.

Ce script n'est interprété qu'à la connexion de l'utilisateur.

JFA - 77

### 2. ~/.bash\_profile, ~/.bash\_login, ~/.profile

Après lecture du fichier /etc/profile, Bash recherche le fichier ~/.bash\_profile, ~/.bash\_login ou ~/.profile dans cet ordre et exécute les commandes contenues dans le premier de ces scripts trouvé et accessible en lecture.

Ce fichier a la même fonction que le fichier /etc/profile, à la différence près qu'il peut être modifié par l'utilisateur pour changer son propre environnement.

Comme le fichier précédent, ce script n'est interprété qu'à la connexion ; les modifications apportées ne sont prises en compte qu'après reconnexion de l'utilisateur.

### 3. ~/.bashrc

Le fichier ~/.profile n'est exécuté qu'à la connexion. Si l'utilisateur dispose d'un environnement...

02/10/2023

## Utilisation du terminal :

L'interface entre UNIX et les terminaux (*tty* abréviation de *teletype*) est prise en charge par un gestionnaire de ligne série (*driver tty*). Celui-ci est paramétrable :

- **erase** annule le dernier caractère frappé (**#**, **^H** ou **DEL**),
- **kill** annule tous les caractères de la ligne (**@** ou **^U**),
- **intr** envoie le signal « **interrupt** » ; interrompt la commande en cours (**DEL** ou **^C**),
- **quit** envoie le signal « **quit** » ; provoque la fin du processus courant (**^I** ou **^Q**),
- **eof** simule une fin de fichier à partir d'un terminal (**^D**),
- **tabs** indique si le terminal sait faire des tabulations (**stty tabs**),
- **speed** indique la vitesse du terminal (**stty speed**).
- **Plus de commandes :**

□ <https://ss64.com/bash/syntax-keyboard.html>

02/10/2023



## Exemples d'utilisation du terminal :

Pour voir les paramètres courants, tapez :

```
prompt> stty
speed 9600 baud; ...
```

Pour voir tous les paramètres, tapez :

```
prompt> stty -a
speed 9600 baud; ... line = 0; ...
```

JFA - 79

Pour modifier un paramètre, deux cas sont possibles :

- ❑ si le paramètre a une valeur :
 

```
stty nom_du_paramètre valeur
```
- ❑ si ce paramètre est un booléen :
 

```
stty nom_du_paramètre, ou
stty -nom_du_paramètre
```

02/10/2023

## Syntaxe générale d'une commande:

**commande [option ...] [argument ...] cd est une primitive du shell**

- Une commande peut être suivie d'options qui précisent la façon dont la commande doit travailler. .

*Exemple :*

```
prompt> wc -l fichier
8 fichier
prompt>
```

JFA - 80

L'option -l de la commande wc indique que l'on ne doit afficher que le nombre de lignes.

Les options ont le format suivant : -l [arg]

- o l est une lettre majuscule ou minuscule.
- o arg est facultatif ([ ]). Il permet de passer un argument à l'option

Il peut être séparé ou non de l'option par un blanc. Sauf remarque particulière, les options se situent OBLIGATOIREMENT avant les noms de fichiers.

02/10/2023

## Cas d'erreurs d'une commande:

Plusieurs cas d'erreurs peuvent se produire :

- La commande n'existe pas.

**ATTENTION** le Shell distingue les lettres majuscules des minuscules.

- Vous n'avez pas le droit d'exécuter cette commande.
- Les options de la commande sont erronées.

En général la commande affiche un message d'utilisation, précisant quels sont les options et arguments. Pour plus de détails taper

**prompt> man commande**

- Les arguments de la commande sont erronés

JFA - S1

02/10/2023

## I. Redirection d'entrées/sorties :

Lorsqu'une commande est lancée, ses entrées/sorties peuvent être redirigées sur un fichier.

Exemple 1 :

**prompt> date > date.out**

**prompt> cat date.out**

**Wed Jul 6 11:34:52 MET 1995**

**prompt>**

Dans ce cas les données produites par la commande **date** ne sont pas affichées sur l'écran, mais écrites dans le fichier **date.out** grâce à la redirection **>**.

Exemple 2 :

**prompt> mail c1 < reponse**

**prompt>**

La commande **mail** lit le texte de la lettre à envoyer depuis le fichier **réponse**, grâce à la redirection **<**, au lieu de lire les données à partir du terminal.

Exemple 3 :

**prompt> wc -l < fichier > nombre**

La commande **wc** compte le nombre de lignes du fichier « fichier » et envoie le résultat dans le fichier « nombre ».

JFA - S2

02/10/2023

## Commandes internes :

- Une commande interne est une commande intégrée au shell (built-in command), autrement dit un mot clé ou mot réservé du langage shell. Pour traiter une commande interne, le shell effectue lui-même un traitement spécifique, sans créer de sous-processus.
- Une commande interne peut donc modifier l'environnement du processus courant.
- Parmi les commandes déjà étudiées, `cd`, `echo`, `nice`, `nohup`, `pwd`, `test`, `trap`, `type`, `umask`, `wait` sont des commandes internes. Certaines comme `cd`, `trap` et `umask` doivent être internes, d'autres comme `echo`, `pwd` et `test` pourraient être externes, mais ont été intégrées au shell pour des raisons de rapidité d'exécution.
- Selon le shell utilisé, les commandes internes peuvent être différentes.

JFA - 83

Exemple :

prompt> type cd

cd est une primitive du shell

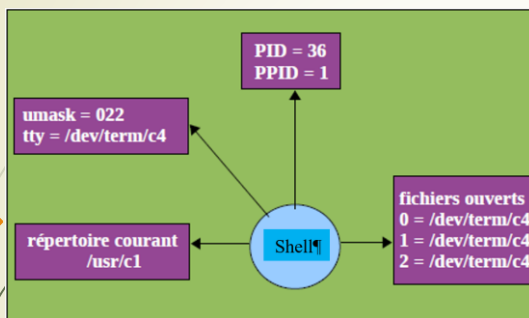
prompt>

la commande `type` indique que `cd` est une commande intégrée au shell. Certaines commandes internes possèdent une page au manuel spécifique. Pour les autres vous devez vous reporter aux pages de manuel concernant le shell lui-même.

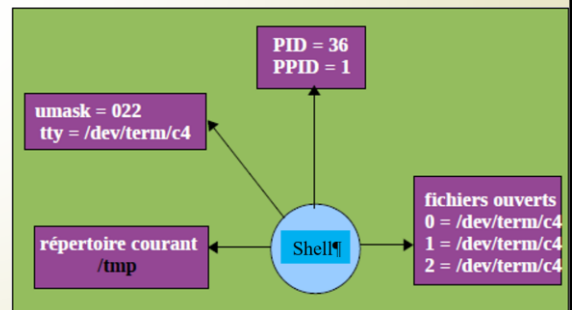
02/10/2023

## Commandes internes :

- Si on exécute une commande interne, il n'y a pas eu de création de sous-processus :



- **Avant** : après exécution de la commande `cd /tmp`, on obtient :



- **Après** : Il n'y a pas eu de création de sous-processus, mais le shell a modifié lui-même un des paramètres de son environnement.

JFA - 84

## Commandes externes :

- Une commande externe est un fichier exécutable. Le shell recherche ce fichier exécutable dans chacun des répertoires spécifiés par la variable **PATH**. Pour traiter une commande, le shell crée un sous-processus. Cette commande ne peut donc en aucun cas modifier l'environnement du processus. Parmi les commandes :

**who, date, ls, cp ...** sont des commandes externes.

- Exemple :

```
jfa@jfa-VirtualBox:~$ type cp
cp est /usr/bin/cp
jfa@jfa-VirtualBox:~$ ls -l /usr/bin/cp
-rwxr-xr-x 1 root root 141824 févr. 7 2022 /usr/bin/cp
jfa@jfa-VirtualBox:~$ file /usr/bin/cp
/usr/bin/cp: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=ed9fc17fdfe02c00607c19b011f27efa, for GNU/Linux 3.2.0, stripped
jfa@jfa-VirtualBox:~$
```

JFA - 85

- type indique le chemin absolu du fichier exécutable qui constitue la commande.
- ls -l confirme que le fichier existe et vous permet de voir les droits d'exécution.
- file vous confirme qu'il s'agit d'un fichier contenant du code machine exécutable. De plus, vous pouvez consulter le manuel en tapant « man cp ».
- Il existe une autre possibilité pour exécuter une commande externe : **exec**

02/10/2023

## Commandes externes :

- Il existe une autre possibilité pour exécuter une commande externe : **exec**

La commande exec sous Linux est utilisée pour exécuter une commande à partir du bash lui-même. Cette commande ne crée pas de nouveau processus, elle remplace simplement le bash par la commande à exécuter. Si la commande exec réussit, elle ne revient pas au processus appelant.

Exemple :

**\$ ps**

**PID TTY TIME CMD**

**5244 term/c4 0:00 ps**

**122 term/c4 0:05 ksh**

**\$ exec cp f1 f2**

JFA - 86

- Dans ce cas, le processus 122, qui exécutait la commande ksh dans un certain environnement, s'est mis à exécuter à la place du shell, la commande cp. Puis lorsque celle-ci s'est terminée, le processus s'est lui aussi terminé, et donc l'utilisateur s'est trouvé déconnecté !

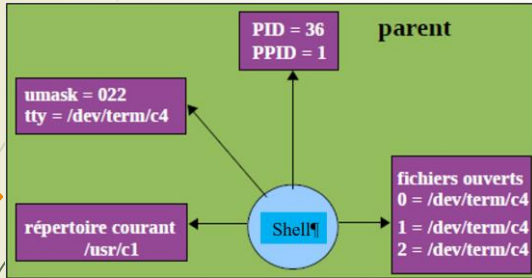
02/10/2023

## Commandes externes :

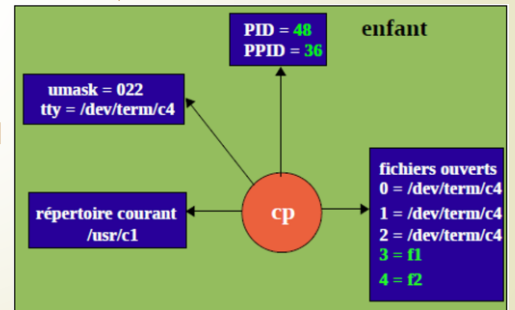
- Si on exécute une commande interne, il n'y a pas eu de création de sous-processus :

- **Avant** : pendant l'exécution de la commande "cp f1 f2": :

JFA - 87

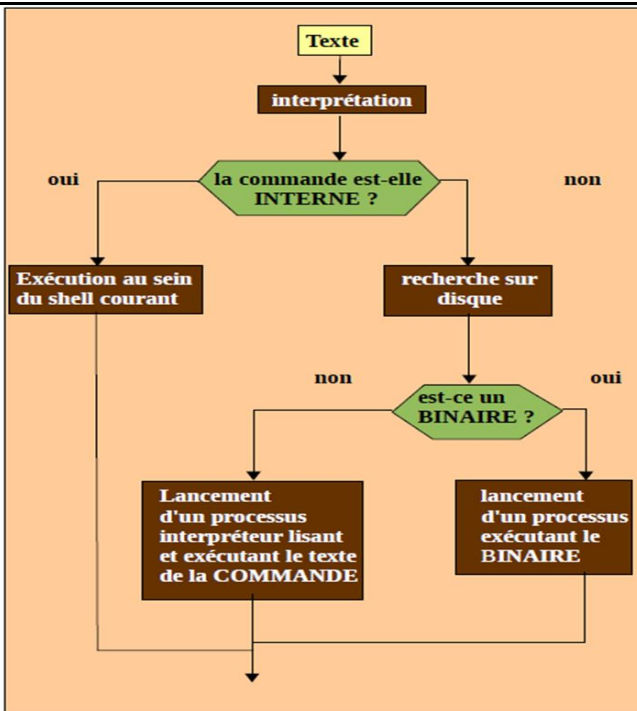


- **Après** : Il y a eu de création d'un sous-processus, et à la fin de la commande, on revient au processus père,



## Synthèse :

JFA - 88



02/10/2023



## Commandes générales du terminal :

Quelques commandes utiles dès le premier contact avec le système :

- **Uname** : informations relatives à la version du système,
- **Date** : date et heure connues par le système,
- **Tty** : nom de la ligne sur laquelle on est connecté,
- **Who** : liste des utilisateurs connectés,
- **Grep** : recherche les fichiers possédant des chaînes de caractères données,
- **Cal** : calendrier des mois de l'année demandée,
- **Pg** : affiche le contenu du fichier passé en argument par pages de 23 lignes,
- **Man** : pages du manuel correspondant à la commande demandée,
- **Pr** : formate le texte du fichier passé en argument pour l'impression,

JFA - 89

02/10/2023

## Commandes générales du terminal :

Quelques commandes utiles dès le premier contact avec le système :

- **Uname** : informations relatives à la version du système,

### ***Exemple d'utilisation***

```
jfa@jfa-VB:~$ uname -a
```

```
Linux jfa-VB 5.11.0-37-generic #41~20.04.2-Ubuntu SMP Fri Sep 24
09:06:38 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

```
jfa@jfa-VB:~$
```

L'option -a (all) affiche toutes les informations.

- **Date** : date et heure connues par le système,

### ***Exemple d'utilisation***

```
jfa@jfa-VB:~$ date
```

```
mar. 12 oct. 2021 12:37:16 CEST
```

```
jfa@jfa-VB:~$
```

**MET** est le nom du fuseau horaire, ici Mean European Time.

JFA - 90

02/10/2023

## Commandes utiles du terminal :

- **Tty** : nom de la ligne sur laquelle on est connecté,

### **Exemple d'utilisation**

```
jfa@jfa-VB:~$ tty
```

```
/dev/pts/0
```

```
jfa@jfa-VB:~$
```

`/dev/tty0` est le nom de la ligne série. En effet sous UNIX les périphériques sont désignés par des noms de fichiers.

- **Who** : liste des utilisateurs connectés,

### **Exemple d'utilisation**

```
jfa@jfa-VB:~$ who
```

```
jfa  :0      2021-10-12 12:29 (:0)
```

```
jfa@jfa-VB:~$
```

02/10/2023

JFA - 91

## Commandes utiles du terminal :

- **Grep** : recherche les fichiers possédant des chaînes de caractères données,

### **Exemple d'utilisation**

```
prompt> grep c1 /etc/passwd
```

```
c1:vs9Fi0TbjD6xg:208:2001:eleve 1:/usr/c1:/bin/ksh
```

```
c10:vs9Fi9bjD6xg:209:2001:eleve
```

```
10:/usr/c10:/bin/ksh
```

```
prompt>
```

Affiche toutes les lignes du fichier `/etc/passwd` contenant la chaîne `c1`.

- **Pg** : affiche le contenu du fichier passé en argument par pages de 23 lignes,

- **Pr** : formate le texte du fichier passé en argument pour l'impression,

02/10/2023

JFA - 92

## Commandes utiles du terminal :

➤ **Cal** : calendrier des mois de l'année demandée,

### ***Exemple d'utilisation***

```
jfa@jfa-VB:~$ cal
      Octobre 2021
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
31
jfa@jfa-VB:~$
```

JFA - 93

02/10/2023

## Commandes utiles du terminal :

➤ **Man** : pages du manuel correspondant à la commande demandée,

### ***Exemple d'utilisation***

```
prompt> man date
Date© ...
prompt>
```

Manuel de référence (man) sur la commande date. Pour avoir plus de détails sur l'utilisation de la commande man, faire « \$ man man ».

### ***Syntaxes :***

```
$ man [COMMAND NAME]
$ man [OPTION] [COMMAND NAME]
$ man [SECTION-NUM] [COMMAND NAME]
```

JFA - 94

02/10/2023

## Commandes utiles du terminal :

Le manuel de référence contient plusieurs sections :

1. Les commandes utilisateurs
2. Les appels systèmes
3. La bibliothèque C
4. Les fichiers spéciaux : les périphériques
5. Le format des fichiers
6. Les jeux
7. Divers
8. Les commandes d'administration et de maintenance
9. Le noyau

JFA - 95

02/10/2023

## Commandes utiles du terminal :

- ❑ Affichage de la page du manuel de la commande *mkfs*, section 8 :  
`prompt> man 8 mkfs`
- ❑ Affichage des pages de toutes les sections du manuel de la commande *passwd* :  
`prompt> man -a passwd`
- ❑ Affichage des pages correspondant à la commande *passwd* dans la section 5 du manuel :  
`prompt> man 5 passwd`
- ❑ Trouver toutes les rubriques contenant un mot clé donné :  
`prompt> man -k mot-cle-donne`
- ❑ Description succincte d'une commande :  
`prompt> man -f passwd`

JFA - 96

<https://www.geeksforgeeks.org/man-command-in-linux-with-examples/>

02/10/2023

## Commandes utiles du terminal :

Quelques commandes utiles dès le premier contact avec le système :

- **Lp** : gère les requêtes d'impression par file d'attente (spooler),
- **Echo** : affiche ses arguments,
- **Wc** : compte le nombre de lignes, de mots et de caractères contenus dans le fichier passé en argument,
- **Sort** : tri le contenu du fichier passé en argument,
- **Mail** : envoie du courrier à un utilisateur,
- **Write** : pour communiquer avec un autre utilisateur connecté sur le système,
- **Mesg** : interdit ou autorise la réception de messages sur son terminal.
- **Halt** : arrêt du système
- **Shutdown** : arrêt du système

JFA - 97

02/10/2023

## Commandes utiles du terminal :

- **Echo** : affiche ses arguments,

**Exemple d'utilisation :**

```
prompt> echo installation terminée
installation terminée
prompt> echo 'voulez-vous continuer [y]/n?\c'
voulez-vous continuer [y]/n? $
```

Affiche à l'écran la chaîne de caractères suivant la commande « echo », après d'éventuelles interprétations (variables, fonctions ...).

- **wc** : compte le nombre de lignes, de mots et de caractères contenus dans le fichier passé en argument,

**Exemple d'utilisation :**

```
prompt> wc fichier
8 48 208 fichier
prompt>
```

Affiche les caractéristiques du fichier de nom « fichier » en termes de contenu, de la façon suivante : il possède 8 lignes, 48 mots et 208 caractères.

JFA - 98

02/10/2023



## Commandes utiles du terminal :

- **Mail** : envoi du courrier à un utilisateur,

### **Exemple d'utilisation :**

```
prompt> mail c2
```

```
Subject: ...
```

```
^D
```

```
prompt>
```

Pour une communication asynchrone (mode messagerie) ; le destinataire n'est pas nécessairement connecté.

- **Shutdown** : arrêt du système,

### **Exemple d'utilisation :**

```
prompt> prompt> shutdown -h now
```

JFA - 99

02/10/2023

## Commandes utiles du terminal :

Pour une communication asynchrone (mode messagerie) ; le destinataire n'est pas nécessairement connecté.

Les personnes « c1 » et « c2 » peuvent communiquer simplement par la fonction « write » :

« c1 » tape :

```
prompt> write c2
```

« c2 » reçoit :

```
« beep » Message from c1 on ttyd1
```

« c1 » tape alors :

```
prompt> write c2
```

A partir de ce moment, tout ce que « c1 » frappe est envoyé à « c2 » dès le retour chariot, et réciproquement. Chacun tape ^D en fin de dialogue. Il s'agit de communications interactives.

JFA - 100

02/10/2023

## Commandes utiles du terminal :

**prompt> mesg is y**

**prompt>**

Permet de voir si l'on autorise (y) les messages à être envoyés sur le terminal courant. Pour les interdire faire :

**prompt> mesg n**

**prompt>**

**JFA - 101**

On peut taper plusieurs commandes sur la même ligne en utilisant le séparateur « ; » :

**prompt> date; who; tty**

02/10/2023

## Autres Commandes utiles du terminal :

- > **Sum** : Calcule et affiche une somme de contrôle pour un fichier (intégrité des fichiers).

**Exemple d'utilisation :**

**prompt> sum fichier**

**08860 1**

**prompt>**

**JFA - 102**

- > **Diff** : Affiche les lignes différentes devant être modifiées pour que les deux fichiers soient identiques.

**Exemple d'utilisation :**

**prompt> diff fichier fichier 1**

**6c6**

**< message**

**---**

**message 2**

**prompt>**

02/10/2023

## Autres Commandes utiles du terminal :

- **Touch** : Modifie la date de dernière modification du fichier, celle-ci devient égale à la date à laquelle la commande a été exécutée. **Si le fichier n'existe pas, il sera créé** (et de taille nulle) sauf si l'option c est utilisée.

### **Exemple d'utilisation :**

```
prompt> touch fichier
```

```
prompt> ls -l fichier
```

```
-rw-r--r-- 1 c1 cours 0 Oct 9 1991 fichier
```

```
prompt>
```

```
prompt> touch test_{a..j}.txt
```

```
prompt> ls -l test*.*
```

```
test_a.txt test_c.txt test_e.txt test_g.txt test_i.txt
```

```
test_b.txt test_d.txt test_f.txt test_h.txt test_j.txt
```

```
prompt>
```

02/10/2023

JFA -103

## Autres Commandes utiles du terminal :

- **Cmp** : Compare octet par octet les deux fichiers passés en paramètre. Cette commande renvoie « 0 » si les fichiers sont identiques, « 1 » sinon.

```
cmp fichier1 fichier2
```

### **Exemple d'utilisation :**

```
prompt> cmp file1.txt file2.txt
```

```
file1.txt file2.txt differ: byte 9, line 2
```

**/\*indique que la première différence trouvée entre les deux fichiers est le 9ème byte de la ligne 2\*/**

```
prompt> cmp file2.txt file3.txt
```

```
prompt>
```

**/\*indique que les fichiers sont identiques\*/**

02/10/2023

JFA -104

## Autres Commandes utiles du terminal :

- **Compress** : Permet une opération de compression visant à diminuer l'espace occupé par les différents fichiers -référéncés. Chaque fichier est remplacé par un nouveau fichier dont la référence est obtenue en suffixant la référence d'origine avec l'extension .Z. Les caractéristiques du fichier sont conservées. Le fichier d'origine est supprimé.

```
compress [options] liste_fichiers
```

### **Exemple d'utilisation :**

```
prompt> compress -v exemple.xls
```

```
exemple.xls : -- replaced with exemple.xls.Z Compression: 24.57%
```

```
prompt>
```

```
/*le fichier exemple.xls est compressé et remplacé par le fichier exemple.xls.Z*/
```

```
Prompt> compress -rv abc
```

```
prompt>
```

```
/*compresse tous les fichier contenus dans abc et ses sous répertoires de manière récursive*/
```

02/10/2023

<https://www.geeksforgeeks.org/compress-command-in-linux-with-examples/>

JFA -105

## Autres Commandes utiles du terminal :

- **uncompress** : Permet la décompression et la reconstruction d'une série de fichiers à partir de leurs formes compressées avec la commande compress. .

```
uncompress [options] liste_fichiers
```

### **Exemple d'utilisation :**

```
prompt> uncompress -v exemple.xls.Z
```

```
exemple.xls : 24.6% -- replaced with exemple.xls
```

```
prompt>
```

```
/*le fichier exemple.xls.Z est décompressé et remplacé par le fichier exemple.xls*/
```

02/10/2023

<https://www.computerhope.com/unix/uuncompr.htm>

JFA -106

## Autres Commandes utiles du terminal :

- **zcat** : Affiche sur la sortie standard le fichier d'origine, correspondant à un fichier compressé au moyen de la commande compress.

```
zcat [options] liste_fichiers.Z
```

### ***Exemple d'utilisation :***

```
prompt> zcat exemple.Z
```

```
Ceci est un fichier exemple
```

```
prompt>
```

```
/* Permet d'afficher de manière lisible le contenu d'un fichier compressé */
```

JFA - 107

<https://www.computerhope.com/unix/uzcat.htm>

02/10/2023

## Autres Commandes utiles du terminal :

- **tar** : tar (tape archiver) permet de gérer des archives de fichiers (sur bande, disque ou fichier). La clé définit les actions de la commande. Elle est constituée d'une suite de caractères définissant la fonction (crtux) et des qualificatifs de cette commande (Abfhmov) . .

```
tar [options] liste_fichiers.tar
```

### ***Exemple d'utilisation :***

```
prompt> tar cvf archive_cible.tar /etc
```

```
/* Place tous les fichiers du répertoire /etc dans le fichier "archive_cible.tar". */
```

```
prompt> tar xvf archive_cible.tar
```

```
/* Extraction de "archive_cible.tar" dans le répertoire courant. */
```

```
prompt> tar tvf archive_cible.tar
```

```
/* Liste les fichiers contenus dans "archive_cible.tar". */
```

JFA - 108

<https://www.computerhope.com/unix/uzcat.htm>

02/10/2023



## Options de Tar :

- **Les différentes fonctions de la commande sont :**
  - **-c** : création d'une nouvelle archive. Sur bande, l'écriture de l'archive a lieu en début de bande et non à la suite du dernier fichier ;
  - **-r** : fonction de remplacement permettant d'écrire en fin d'archive les fichiers de références données ;
  - **-t** : liste des références de fichiers dans l'archive sans restitution ;
  - **-u** : les fichiers sont ajoutés en fin d'archive s'ils n'y figurent pas encore ou si la date de modification de la dernière version archivée est antérieure à la version du fichier sur le disque ;
  - **-x** : fonction d'extraction de l'archive. Si la référence examinée est une référence de répertoire, son contenu est extrait de manière récursive. Si aucune référence de fichier n'est donnée, tous les fichiers de l'archive sont extraits.
- **Un certain nombre d'indications supplémentaires permettent de qualifier la fonction réalisée :**
  - **-A** : les messages d'avertissement sont supprimés ;
  - **-f** : l'argument suivant est interprété comme une référence de fichier correspondant au nom de l'archive (au lieu d'une référence par défaut qui est en général celle d'un fichier spécial associé à un dérouleur de bande). Si cet argument est -, la commande lit sur l'entrée standard ou écrit sur la sortie standard ;
  - **-h** : les liens symboliques sont suivis (par défaut, ils ne le sont pas) ;
  - **-v** : option «verbeuse».

JFA -109

02/10/2023