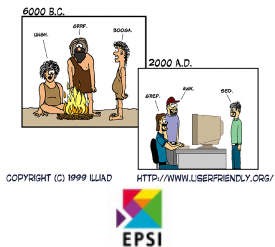


Environnement Linux



Sommaire

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

Paramétrage de l'environnement

- 1 Paramétrage de l'environnement
 - Invite de commande
 - Raccourcis utiles
 - Configuration
 - Caractères spéciaux
 - Redirections
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

Introduction

Le shell est la principale interface entre vous et le système d'exploitation.

```
tom@workine:~$
```

- ▶ Il vous fournit un *environnement de travail*.
- ▶ Il interprète et exécute les commandes que vous lui indiquez, soit directement, soit par l'intermédiaire de scripts
- ▶ C'est le 1^{er} programme à être lancé après une connexion réussie.

Invite de commande

L'invite de commande

```
tom@workine:~$
```

- ▶ Indique par sa présence que le système est en attente d'une entrée utilisateur.
- ▶ Fournie à l'utilisateur un certain nombre d'informations sur son environnement de travail

Syntaxe d'une ligne de commande

Les commandes GNU/Linux obéissent *généralement* au schéma suivant :

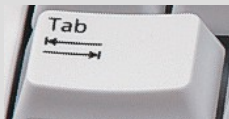
```
tom@workine:~$ ls -al --color=auto /var/log
```

Le shell vous facilite la vie

Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?

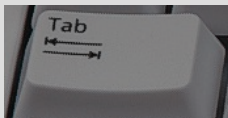
Le shell vous facilite la vie

Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



Le shell vous facilite la vie

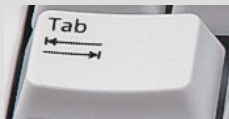
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowr<TAB>
```

Le shell vous facilite la vie

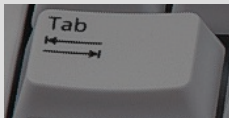
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter
```

Le shell vous facilite la vie

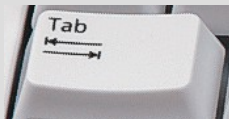
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter /v<TAB>
```

Le shell vous facilite la vie

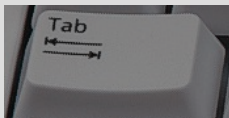
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter /var/
```

Le shell vous facilite la vie

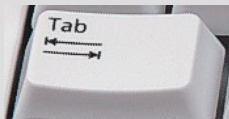
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter /var/lo<TAB>  
local log lock
```

Le shell vous facilite la vie

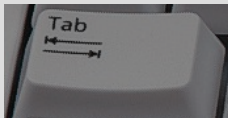
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter /var/log/
```

Le shell vous facilite la vie

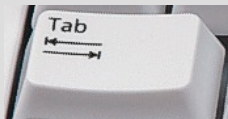
Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter /var/log/mess<TAB>
```

Le shell vous facilite la vie

Quelle est 1^{ère} touche la plus utilisée par un administrateur Linux ?



```
$ oowriter /var/log/messages
```


Le shell vous facilite la vie : Historique

Le shell garde un historique des commandes.

Les touches **haut et bas** permettent de naviguer dans l'historique

Le shell vous facilite la vie : Historique

Le shell garde un historique des commandes.

Les touches **haut et bas** permettent de naviguer dans l'historique

La séquence **Ctrl+r+recherche** permet de recherche dans l'historique

Le shell vous facilite la vie : Historique

Le shell garde un historique des commandes.

Les touches haut et bas permettent de naviguer dans l'historique

La séquence Ctrl+r+recherche permet de recherche dans l'historique

La séquence !! rappelle la dernière commande exécutée.

Le shell vous facilite la vie : Édition de la ligne de commande

En plus des touches fléchées, des raccourcis claviers permettent d'éditer facilement une ligne de commande

Control+a pour aller en début de ligne.

Control+e pour aller en fin de ligne.

Control+w pour effacer le mot *avant* le curseur.

Control+k efface tout à droite du curseur.

Control+u efface tout à gauche du curseur.

Control+c efface toute la ligne.

Control+l raccourci pour *clear* : efface l'écran.

Le shell vous facilite la vie : Utilisation de la précédente commande

- ▶ `^erreur^correction` rappelle la précédente commande en remplaçant la chaîne *erreur* par *correction*.
- ▶ `!$` représente le dernier argument de la précédente commande
- ▶ `!*` représente l'ensemble des arguments de la précédente commande
- ▶ `!!: n` représente le n^e argument de la précédente commande

Le shell vous facilite la vie : les alias

```
tom@cafeine$ alias ll='/bin/ls -a --color=auto -c1'  
tom@cafeine$ ll  
. lesshist  
.  
. recently-used.xbel  
. ibam  
. gimp-2.4  
. viminfo  
. gtk-bookmarks  
. mysql_history
```

Les alias permettent de créer des raccourcis

- ▶ Pour les commandes complexes

Le shell vous facilite la vie : les alias

```
tom@cafeine$ alias ll='/bin/ls -a --color=auto -c1'  
tom@cafeine$ ll  
. lesshist  
.  
. recently-used.xbel  
. ibam  
. gimp-2.4  
. viminfo  
. gtk-bookmarks  
. mysql_history
```

Les alias permettent de créer des raccourcis

- ▶ Pour les commandes complexes
- ▶ et souvent utilisées.

L'environnement

- ▶ Le shell représente votre environnement de travail
- ▶ Différentes méthodes permettent d'adapter cet environnement :
- ▶ Les alias
- ▶ Les *variables d'environnement*
 - ▶ La modification d'une VE entraîne un changement de comportement immédiat.
 - ▶ Elles sont en générales initialisées par l'intermédiaire des fichiers de configuration.
 - ▶ A moins d'être écrite dans le fichier de configuration adéquat, cette modification sera perdue à la prochaine fin de session.

Les variables d'environnement

```
PS1 = \u @ \h : \w \s
```

```
tom@workine:~$
```

PS1 permet de modifier l'apparence de l'invite de commande.

Les variables d'environnement

PATH détermine les répertoires dans lesquels le shell va chercher les commandes que vous lui passez.

Configuration du shell

- ▶ Lors de son lancement, le shell va lire un certain nombre de fichiers
- ▶ qui va lui permettre de mettre en place un environnement de travail adéquat.
 - ▶ Dans le cas d'un shell *de connexion*, il lira les fichiers `/etc/profile` et `~/.bash_profile`.
 - ▶ Dans le cas d'un shell non interactif, il lira les fichiers `~/.bashrc`

Caractère spéciaux

Les caractères spéciaux sont *interprétés* par le shell.

| tube anonyme

|| "OU logique"

& exécution en arrière-plan

&& "ET logique"

; sépare plusieurs expressions

< et > caractères de redirection

espace et tabulation

Si on souhaite utiliser ces caractères *textuellement*, on doit les *échapper* ou les *citer*.

Jokers

```
tom@cafeine$ mv *.exe Windows/
```

Les jokers permettent de représenter un ou plusieurs caractères :

* représente n'importe quel chaîne de caractères

? représente n'importe quel caractère

[a-d] n'importe quel caractère de l'intervalle spécifiée

[abc] soit a, soit b, soit c

Jokers

```
tom@cafeine$ mv *.mp? ~/Multimedia/
```

Les jokers permettent de représenter un ou plusieurs caractères :

* représente n'importe quel chaîne de caractères

? **représente n'importe quel caractère**

[a-d] n'importe quel caractère de l'intervalle spécifiée

[abc] soit a, soit b, soit c

Jokers

```
tom@cafeine$ ls -cl [a-d]*
archivage.tex
communication.tex
beamerthemeopendoor.sty
```

Les jokers permettent de représenter un ou plusieurs caractères :

* représente n'importe quel chaîne de caractères

? représente n'importe quel caractère

[a-d] n'importe quel caractère de l'intervalle spécifiée

[abc] soit a, soit b, soit c

Jokers

```
tom@cafeine$ ls -c1 [ab]*  
archivage.tex  
beamerthemeopendoor.sty  
tom@cafeine$ █
```

Les jokers permettent de représenter un ou plusieurs caractères :

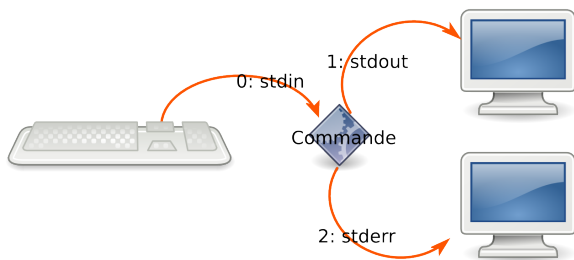
* représente n'importe quel chaîne de caractères

? représente n'importe quel caractère

[a-d] n'importe quel caractère de l'intervalle spécifiée

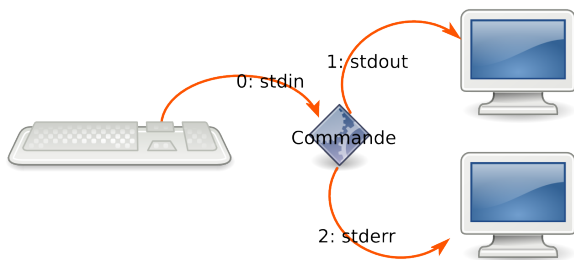
[abc] soit a, soit b, soit c

Entrées / Sorties et redirections



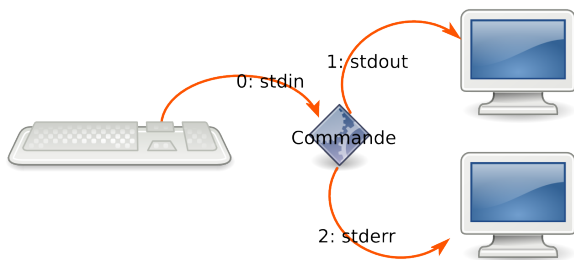
- ▶ La communication avec un programme se fait au moyen de *fichiers spéciaux*.

Entrées / Sorties et redirections



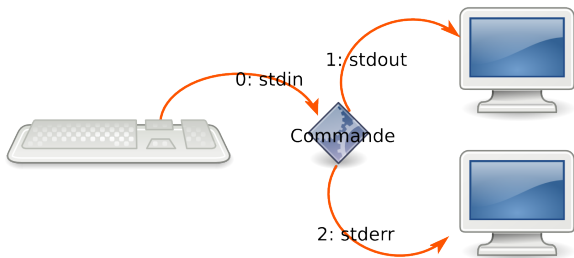
- ▶ La communication avec un programme se fait au moyen de *fichiers spéciaux*.
- ▶ Il est possible de *détourner* ces communications pour

Entrées / Sorties et redirections



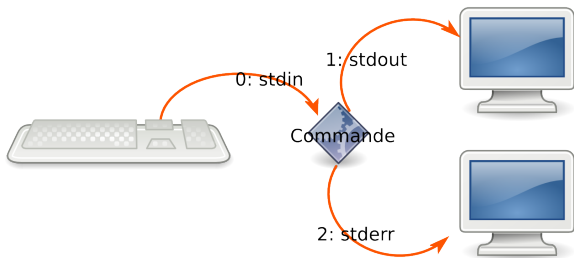
- ▶ La communication avec un programme se fait au moyen de *fichiers spéciaux*.
- ▶ Il est possible de *détourner* ces communications pour
 - ▶ récupérer le résultat d'une commande dans un fichier.

Entrées / Sorties et redirections



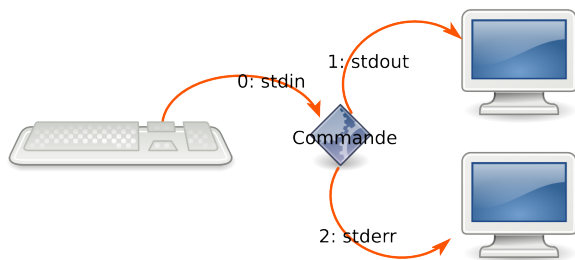
- ▶ La communication avec un programme se fait au moyen de *fichiers spéciaux*.
- ▶ Il est possible de *détourner* ces communications pour
 - ▶ récupérer le résultat d'une commande dans un fichier.
 - ▶ récupérer les erreurs issue de l'exécution d'une commande.

Entrées / Sorties et redirections



- ▶ La communication avec un programme se fait au moyen de *fichiers spéciaux*.
- ▶ Il est possible de *détourner* ces communications pour
 - ▶ récupérer le résultat d'une commande dans un fichier.
 - ▶ récupérer les erreurs issue de l'exécution d'une commande.
 - ▶ renvoyer le résultat d'une commande à une autre commande.

Entrées / Sorties et redirections



- ▶ La communication avec un programme se fait au moyen de *fichiers spéciaux*.
- ▶ Il est possible de *détourner* ces communications pour
 - ▶ récupérer le résultat d'une commande dans un fichier.
 - ▶ récupérer les erreurs issue de l'exécution d'une commande.
 - ▶ renvoyer le résultat d'une commande à une autre commande.
 - ▶ écrire des scripts non interactifs.

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

rediriger les 2 : `ls -al &> fichier`

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

rediriger les 2 : `ls -al &> fichier`

ou : `ls -al > fichier 2>&1`

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

rediriger les 2 : `ls -al &> fichier`

ou : `ls -al > fichier 2>&1`

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

rediriger les 2 : `ls -al &> fichier`

ou : `ls -al > fichier 2>&1`

- ▶ Attention, si le fichier existe, son contenu sera *écrasé*

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

rediriger les 2 : `ls -al &> fichier`

ou : `ls -al > fichier 2>&1`

- ▶ Attention, si le fichier existe, son contenu sera *écrasé*
- ▶ Sauf si on double le symbole de redirection `>>`

Redirection dans un fichier

rediriger stdout : `ls -al > fichier_resultat`

rediriger stderr : `ls -al 2> fichier_erreur`

rediriger les 2 : `ls -al &> fichier`

ou : `ls -al > fichier 2>&1`

- ▶ Attention, si le fichier existe, son contenu sera *écrasé*
- ▶ Sauf si on double le symbole de redirection `>>`
 - ▶ dans ce cas, les informations seront écrites à la fin du fichier.

Redirection depuis un fichier

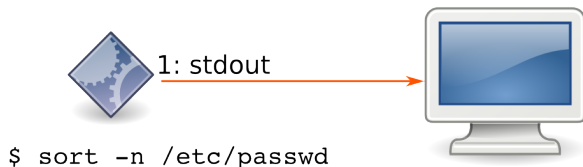
Elle consiste à remplacer l'entrée standard d'une commande par un fichier.

```
mysql < ~/base.sql
```

Les tubes

- ▶ Les tubes sont un mécanisme permettant de passer le résultat d'une commande à une autre.
- ▶ Ils sont représentés par le symbole | (alt gr + 6)

Les tubes



- ▶ Les tubes sont un mécanisme permettant de passer le résultat d'une commande à une autre.
- ▶ Ils sont représentés par le symbole | (alt gr + 6)

Les tubes



- ▶ Les tubes sont un mécanisme permettant de passer le résultat d'une commande à une autre.
- ▶ Ils sont représentés par le symbole | (alt gr + 6)

Redirections et tubes : la synthèse

- ▶ Si on souhaite enregistrer les résultats d'une commande dans un fichier

Redirections et tubes : la synthèse

- ▶ Si on souhaite enregistrer les résultats d'une commande dans un fichier
- ▶ en conservant l'affichage sur la console.

Redirections et tubes : la synthèse

- ▶ Si on souhaite enregistrer les résultats d'une commande dans un fichier
- ▶ en conservant l'affichage sur la console.
- ▶ on peut utiliser la commande *tee* :

Redirections et tubes : la synthèse

- ▶ Si on souhaite enregistrer les résultats d'une commande dans un fichier
- ▶ en conservant l'affichage sur la console.
- ▶ on peut utiliser la commande *tee* :

```
$ ls -Alr > fichier
```

tout dans *fichier*, rien sur la console

Redirections et tubes : la synthèse

- ▶ Si on souhaite enregistrer les résultats d'une commande dans un fichier
- ▶ en conservant l'affichage sur la console.
- ▶ on peut utiliser la commande *tee* :

```
$ ls -Alr | tee fichier
```

tout dans *fichier* **et** sur la console.

Vi

- 1 Paramétrage de l'environnement
- 2 Vi**
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

Introduction



- ▶ *vi* est un éditeur de texte
- ▶ disponible sur l'ensemble des systèmes de type unix / Linux.
- ▶ modulaire, il peut servir aussi bien comme éditeur de base, que comme plateforme complète de développement.
- ▶ Très puissant, sa maîtrise demande un peu d'habitude et de pratique.

Question

Quelle est la 2^e touche la plus utilisée d'un clavier d'utilisateur linux ?

Question

Quelle est la 2^e touche la plus utilisée d'un clavier d'utilisateur linux ?



Particularités



```
echo hello world
test.sh          3,1          Bas
```

vi propose 2 modes :

- ▶ Un mode *commande* permettant de manipuler le texte dans son ensemble.
- ▶ Un mode *insertion* permettant de taper le texte.

Un 3^e mode *visuel* permet de faire des sélections.

Particularités

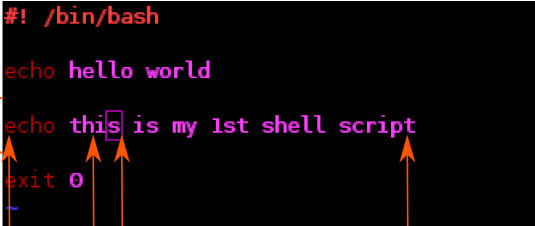
```
echo hello world
test.sh          3,1          Bas
-- INSERTION --
```

vi propose 2 modes :

- ▶ Un mode *commande* permettant de manipuler le texte dans son ensemble.
- ▶ Un mode *insertion* permettant de taper le texte.

Un 3^e mode *visuel* permet de faire des sélections.

Pour passer d'un mode à l'autre



The screenshot shows a terminal window with a black background and red text. The script content is: `#!/bin/bash`, `echo hello world`, `echo this is my 1st shell script`, and `exit 0`. A blue cursor is positioned at the end of the second line. Annotations include: an orange arrow labeled 'O' pointing to the start of the second line; an orange arrow labeled 'o' pointing to the start of the third line; a small orange box around the word 'this' on the third line; an orange arrow labeled 'i' pointing to the start of the word 'this'; an orange arrow labeled 'a' pointing to the end of the word 'this'; and an orange arrow labeled 'A' pointing to the end of the third line. Below the terminal, the letters 'l', 'i', 'a', and 'A' are aligned with their respective arrows.

```
#!/bin/bash
echo hello world
echo this is my 1st shell script
exit 0
```

on utilise

- ▶ les touches **i**, **I**, **o**, **O**, **a**, **A** pour passer du mode *commande* au mode insertion.

Pour passer d'un mode à l'autre

```
#!/bin/bash
echo hello world
echo this is my 1st shell script
exit 0
```

on utilise

- ▶ les touches **i**, **I**, **o**, **O**, **a**, **A** pour passer du mode *commande* au mode insertion.
- ▶ la touche **echap** pour passer en mode *commande*.

Pour passer d'un mode à l'autre

```

#!/bin/bash

echo hello world

echo this is my 1st shell script

exit 0
  
```

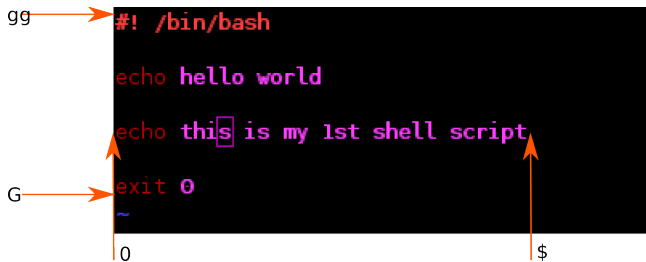
on utilise

- ▶ les touches **i**, **I**, **o**, **O**, **a**, **A** pour passer du mode *commande* au mode insertion.
- ▶ la touche **echap** pour passer en mode *commande*.
- ▶ Attention de ne pas mélanger les 2 modes !

Édition et navigation

La touche **i** permet de passer en mode *insertion* et d'éditer le texte avec les touches habituelles. Les touches fléchées, Delete, backspace, Tab, PGUp, PGDown, etc. fonctionnent de manière habituelle.

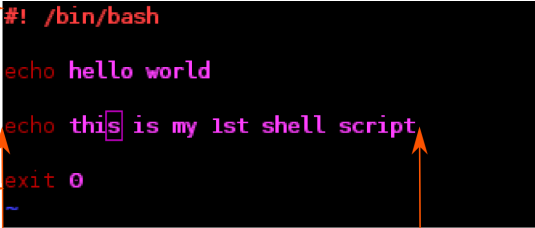
Navigation en mode commande



The image shows a terminal window with a black background and white text. The text is a shell script with four lines: `#!/bin/bash`, `echo hello world`, `echo this is my 1st shell script`, and `exit 0`. The cursor is positioned at the end of the third line. There are three orange arrows pointing to specific parts of the script: one from the label 'gg' to the start of the first line, one from the label 'G' to the start of the fourth line, and one from the label '\$' to the end of the third line. A small blue box highlights the word 'this' in the third line. Below the terminal window, there are two labels: '0' on the left and '\$' on the right, both with orange arrows pointing to the start and end of the third line respectively.

```
gg → #! /bin/bash  
echo hello world  
echo this is my 1st shell script  
G → exit 0  
~  
0                                $
```

Navigation en mode commande



The screenshot shows a terminal window with a black background and red and purple text. The script content is: `#!/bin/bash`, `echo hello world`, `echo this is my 1st shell script`, and `exit 0`. The cursor is positioned at the end of the third line. Three orange arrows point to specific locations: one to the start of the first line (labeled 'gg'), one to the start of the third line (labeled 'G'), and one to the end of the third line (labeled '\$'). Below the terminal, a line of text shows a tilde '~' at the start and a '\$' at the end, with a '0' under the tilde and a '\$' under the dollar sign.

```
gg → #! /bin/bash
    echo hello world
    echo this is my 1st shell script
G → exit 0
    ~
    0                                     $
```

- ▶ Les éléments de *navigation* sont relatifs à la position du curseur.
- ▶ Ils peuvent s'appliquer à des *commandes* afin d'en spécifier l'*étendue*.
- ▶ Ils peuvent aussi être *quantifiés*.
- ▶ Ainsi, la séquence de touche `<Esc>d3w` va supprimer les 3 mots suivant le curseur. `dG` va supprimer toutes les lignes depuis le curseur jusqu'à la fin du fichier.

Les commandes de suppression

d

Les commandes de suppression

ddelete - supprimer

Les commandes de suppression

ddelete - supprimer
c

Les commandes de suppression

ddelete - supprimer
correct - corriger

Suppression de texte

- x supprime le caractère sous le curseur
- dmouvement supprime *mouvement* (ex : d0 supprime tout du curseur jusqu'au début de la ligne).
- cmouvement est équivalent, mais passe en mode *insertion* après la suppression.

Suppression de texte

- x supprime le caractère sous le curseur
- dmouvement supprime *mouvement* (ex : d0 supprime tout du curseur jusqu'au début de la ligne).
- cmouvement est équivalent, mais passe en mode *insertion* après la suppression.
- dd supprime l'ensemble de la ligne courante.
- cc supprime toute la ligne courante et passe ensuite en mode *insertion*.

Suppression de texte

- x supprime le caractère sous le curseur
- dmouvement supprime *mouvement* (ex : d0 supprime tout du curseur jusqu'au début de la ligne).
- cmouvement est équivalent, mais passe en mode *insertion* après la suppression.
- dd supprime l'ensemble de la ligne courante.
- cc supprime toute la ligne courante et passe ensuite en mode *insertion*.
- D supprime tout jusqu'à la fin de la ligne.
- C idem, en passant ensuite en mode *insertion*

Couper-Copier-coller

Sous vi, la suppression est en fait une “coupure”. On peut aussi copier de manière explicite :

ymovement *copie* mouvement.

Y ou **yy** *copie* la ligne entière.

Couper-Copier-coller

Sous vi, la suppression est en fait une “coupure”. On peut aussi copier de manière explicite :

ymovement *copie* mouvement.

Y ou **yy** *copie* la ligne entière.

Et pour coller :

p permet de *coller* l'élément précédemment coupé à *droite* du curseur.

P *colle* l'élément à *gauche* du curseur.

Annuler

vi propose un mécanisme d'annulation :

- u permet d'annuler la dernière opération
- . permet de répéter la dernière opération

Control+r revient dans l'historique des opérations (inverse de *u*).

Quitter vi

Pour finir :

:w écrit les modifications apportées au fichier.

:q quitte vi

:q! quitte vi en abandonnant les modifications.

:wq combine les 2 1^{res} commandes.

Configuration

La configuration de *vim* est centralisée :

- ▶ dans `/etc/vim/vimrc` pour la configuration globale.
- ▶ dans `~/.vimrc` pour la configuration personnelle
- ▶ Le 2^efichier étant interprété *après* le 1^{er}.

Les principales directives du fichier personnel

```
filetype plugin on
syntax on
set showmode
set splitbelow
set splitright
set expandtab
set shiftwidth=3
set tabstop=3
set softtabstop=3
set smarttab
set smartindent
set foldmethod=indent
set nohlsearch
set showmatch
set binary noel
set backspace=indent,eol,start
set laststatus=2
set nocompatible
set visualbell
set ruler
set autochdir
set background=dark
set cursorline
set linebreak

set modeline
set modelines=5
"_F2_permet_de_passer_en_mode_(no)paste
" avec affichage dans la barre de statut
nnoremap <F2> :set invpaste paste?<CR>
inoremap <F2> <C-O><F2>
```


Correction Orthographique

```

echo "Ceci est un tesst"

i=$RANDOM
( $i -lt 15000 ) && echo $i
exit 0

```

- ▶ Vim permet de détecter les erreurs d'orthographe en ignorant les mots-clé du langage utilisé.
- ▶ Il suffit de rajouter les instructions suivantes dans le fichier de configuration :

```

nnoemap <F3> :set invspell spell?<CR>
set spelllang=fr
augroup filetypedetect
  au BufNewFile,BufRead *.sh setlocal spell spelllang=fr
  au BufNewFile,BufRead *.c setlocal spell spelllang=fr
  au BufNewFile,BufRead *.tex setlocal spell spelllang=fr
augroup END

```

- ▶ L'activation / désactivation de la correction se fait par la touche <F3>.
- ▶ La commande z= affiche une liste de proposition de correction.

Correction Orthographique

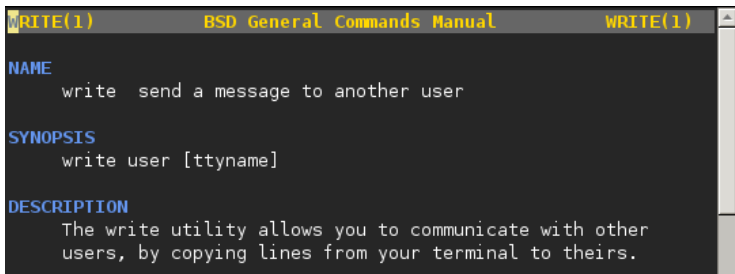
```
Remplacer "tesst" par :
1 "test"
2 "tests"
3 "t'est"
4 "est"
5 "lest"
```

- ▶ Vim permet de détecter les erreurs d'orthographe en ignorant les mots-clé du langage utilisé.
- ▶ Il suffit de rajouter les instructions suivantes dans le fichier de configuration :

```
nnooremap <F3> :set invspell spell?<CR>
set spelllang=fr
augroup filetypedetect
  au BufNewFile,BufRead *.sh setlocal spell spelllang=fr
  au BufNewFile,BufRead *.c setlocal spell spelllang=fr
  au BufNewFile,BufRead *.tex setlocal spell spelllang=fr
augroup END
```

- ▶ L'activation / désactivation de la correction se fait par la touche <F3>.
- ▶ La commande **z=** affiche une liste de proposition de correction.

Page de man



```
WRITE(1) BSD General Commands Manual WRITE(1)

NAME
  write  send a message to another user

SYNOPSIS
  write user [ttyname]

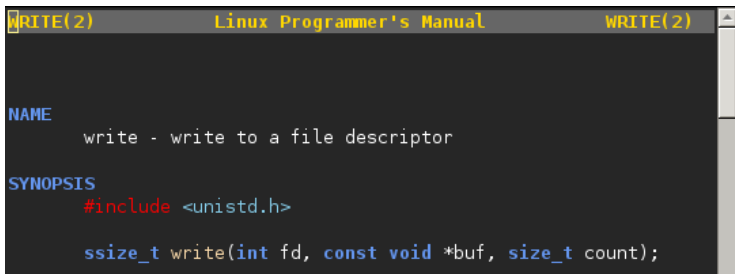
DESCRIPTION
  The write utility allows you to communicate with other
  users, by copying lines from your terminal to theirs.
```

On peut configurer vim pour afficher la page de manuel d'un mot. Il suffit d'ajouter les instructions suivantes dans le fichier de configuration :

```
runtime ftplugin/man.vim
nnoremap K :Man <word><CR>
nnoremap k :Man 2 <word><CR>
```

- ▶ la touche "K" affiche la page de man du mot sous le curseur.
- ▶ la touche "k" affiche la page de man de l'appel-système sous le curseur.

Page de man



```

WRITE(2)                               Linux Programmer's Manual                               WRITE(2)

NAME
  write - write to a file descriptor

SYNOPSIS
  #include <unistd.h>

  ssize_t write(int fd, const void *buf, size_t count);
  
```

On peut configurer vim pour afficher la page de manuel d'un mot. Il suffit d'ajouter les instructions suivantes dans le fichier de configuration :

```

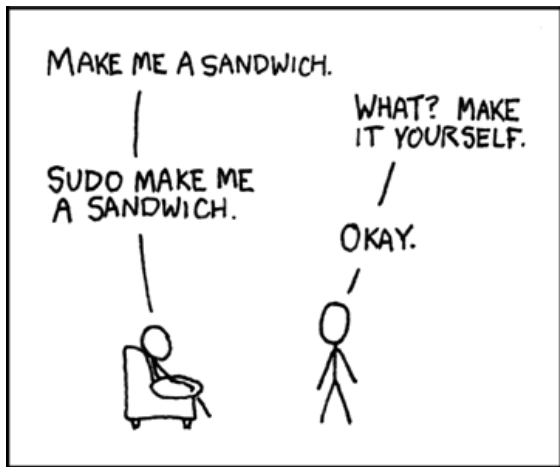
runtime ftplugin /man.vim
nnoremap K :Man <word><CR>
nnoremap k :Man 2 <word><CR>
  
```

- ▶ la touche "K" affiche la page de man du mot sous le curseur.
- ▶ la touche "k" affiche la page de man de l'appel-système sous le curseur.

Sudo

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo**
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

Environnement Linux



Introduction

Comment...

... autoriser un utilisateur à redémarrer le serveur apache sans lui communiquer le mot de passe "root" ?

Le bit *suid*

les permissions *suid* et *sgid* répondent en partie à cette problématique.

Le bit *suid*

les permissions *suid* et *sgid* répondent en partie à cette problématique. mais :

Le bit suid

les permissions *suid* et *sgid* répondent en partie à cette problématique. mais :

- ▶ les shell récents désactivent ces permissions avant d'exécuter un script.
- ▶ les bits suid et sgid sont une source d'insécurité importante.

Sudo

sudo permet d'autoriser des utilisateurs à exécuter certaines commandes.

- ▶ *sudo* demande à l'utilisateur de s'authentifier.
- ▶ Il interroge le fichier */etc/sudoers* pour vérifier les autorisations.
- ▶ Il exécute la commande en mode privilégié.
- ▶ Il envoie à *syslog* un message contenant l'utilisateur, le résultat d'authentification et la commande à exécuter.
- ▶ L'utilisateur peut continuer d'utiliser *sudo* pendant 15 minutes sans avoir à s'authentifier de nouveau

Le fichier /etc/sudoers 1/2

Ce fichier définit *qui* a le droit de faire *quoi*, ainsi qu'un certain nombre *d'alias*. Les alias obéissent à la syntaxe **type** NOM membre1, membre2, membre3, avec *type* :

host_Alias permet de définir un groupe de machines

Runas_Alias permet de définir les utilisateurs sous l'identité desquels les commandes seront exécutées.

User_Alias permet de regrouper des utilisateurs. Il est possible d'inclure des noms de groupes en les préfixant par le symbole %.

Cmnd_Alias permet de définir une liste de commandes. Celles-ci doivent être référencées en utilisant leur *chemin absolu*

Le fichier `/etc/sudoers` 2/2

Le fichier associe les utilisateurs, les commandes qu'ils ont le droit d'exécuter, et les identités sous lesquelles ils peuvent exécuter ces commandes.

Exemples : `tom = localhost (operator) /bin/lis`, `(root) /etc/init.d/apache2 reload`

Gérer le système de fichiers

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers**
 - Introduction
 - Répertoires
 - Attributs et types de fichiers
 - Commandes associées
 - Find
- 5 Utilisateurs
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

Gérer le système de fichiers



Arborescence

- ▶ La structure d'un système de fichier Linux obéit au standard *FHS*.
- ▶ Les fichiers sont organisés en répertoires selon une structure *arborescente*.
- ▶ Le point de départ de cette arborescence est / (root, racine).

Les répertoires

- ▶ Les répertoires sont les unités d'organisation de base d'un système Linux.
- ▶ Leur structure respecte une norme (le *FHS*) :
 - /bin** contient les binaires essentiels au fonctionnement du système.
 - /boot** contient les fichiers nécessaires au démarrage du système.
 - /dev** fournit une représentation "fichier" des périphériques du système.
 - /etc** rassemble les fichiers de configuration du système et de ses composants et logiciels.
 - /home** centralise les répertoires personnels des utilisateurs du système.

Les répertoires

`/usr` contient les programmes applicatifs installés sur le système (binaires, bibliothèques, documentation. . .)

`/sbin` `/et` `/usr/sbin` contiennent les commandes réservées à l'administration.

`/root` est le répertoire personnel de l'administrateur.

`/proc` et `/sys` sont une représentation "fichier" de l'état du système.

`/tmp` rassemble les données temporaires

`/var` contient les données des programmes

Les répertoires

Dans chaque répertoire se trouvent 2 répertoires spéciaux :

- . fait référence au répertoire *courant*.
- .. fait référence au répertoire *parent*.

Taille d'un répertoire

```
du [option] [fichier|repertoire]
```

```
du -h -c ~/Documents/Epsi
8,0K    /home/tom/Documents/Epsi/1 i / test
1,7M    /home/tom/Documents/Epsi/1 i
1,9M    /home/tom/Documents/Epsi/1A
3,6M    total
```

Affiche la taille du fichier ou du répertoire passé en paramètre. Principales options :

- h taille formatée pour une lecture facile.
- s n'affiche que la taille totale

Taille du système de fichier

```
df [option] [systeme de fichier]
```

```
df -h
Sys. de fichiers  Taille  Uti. Disp.  Uti% Monte sur
/dev/sda          12G    5,2G  6,1G  46% /
...
turbine :/srv     368G   268G   82G  77% /srv
```

df affiche l'espace disponible / utilisé des différents systèmes de fichiers.
L'option *-h* formate les tailles en Mo, Go, etc.

Noms de fichiers

Quelques différences entre système Linux et système Microsoft :

- ▶ Les noms de fichiers sont *SenSibleS à lA cASsE*.
- ▶ L'extension n'est pas obligatoire et est d'ordre purement "cosmétique".
- ▶ Pour cacher un fichier, il faut préfixer son nom par un "."

Lire les attributs d'un fichier

Un fichier unix est divisé en 2 parties :

- ▶ Les attributs ou *métadonnées*.
- ▶ Les données

Les principaux attributs sont visibles grâce à la commande `ls`

La commande ls

```
tom@workine:~$ ls -l ~
drwx----- 2 tom maison 4,0K jui 23 08:33 PDF
drwxr-xr-x  2 tom maison 4,0K jan  9  2008 Templates
-rw-r--r--  1 tom maison 8,0K jui 23 09:47 console.png
drwxr-xr-x  2 tom maison 4,0K jui 18 11:29 public_html
...
```


Les types de fichier

Le type de fichier est indiqué par le tout premier symbole du résultat de la commande ls.

```
tom@workine:~$ ls -l ~
drwx----- 2 tom maison 4,0K jui 23 08:33 PDF
drwxr-xr-x  2 tom maison 4,0K jan  9 2008 Templates
-rw-r--r--  1 tom maison 8,0K jui 23 09:47 console.png
drwxr-xr-x  2 tom maison 4,0K jui 18 11:29 public_html
...
```

Type:

- fichier régulier ou lien dur
- l lien symbolique
- d répertoire
- c périphérique en mode caractère
- b périphérique en mode bloc
- s socket
- p canal nommé

Les attributs de date

- ▶ date
 - ▶ de modification (*mtime*).
 - ▶ d'accès (*atime*).
 - ▶ de modification des attributs (*ctime*)).

Copier, supprimer et renommer un fichier

cp option source destination : copie de fichier. L'option *-a* *préserve les attributs*.

rm fichier : supprime *fichier* **définitivement** après demande de confirmation.

mv source destination : déplace / renomme le fichier *source* en *destination*.

Visualiser un fichier

`cat fichier` permet d'afficher l'intégralité du contenu de *fichier*.

`less fichier` permet :

- ▶ de naviguer dans le fichier à l'aide des touches fléchées ; gg ; G
- ▶ de faire une recherche : */terme recherché*.
- ▶ d'ouvrir le fichier dans un éditeur (touche v).

La commande file

```
tom@workine:~$ file support.pdf
support.pdf: PDF document, version 1.4
```

file option fichier : permet d'afficher différentes informations relatives aux *données* du fichier

La commande find

find permet de rechercher des fichiers en fonctions de leurs *attributs*.

`find chemin_depart expression`

`chemin_depart` : dossier à partir duquel la recherche est faite.

`expression` : est composée d'*options*, d'*actions* et de *tests*

Find : tests

Les *tests* permettent de limiter l'étendue d'une recherche aux fichiers répondant à des critères particuliers :

```
tom@cafeine$ find ~ -name "*.pdf" 
```

- type d,f,l,... recherche sur le type de fichier.
- name nom recherche sur le nom du fichier.
- perm permissions recherche sur les permissions.
- size taille(bkM) recherche sur la taille.
- user nom : recherche sur le propriétaire

Find : éléments numériques pour les tests

lorsqu'un test prend une valeur *numérique*, celle-ci peut être spécifiée de la manière suivante :

```
tom@cafeine$ find . -cmin -60 -ls
```

+num : supérieur à *num*

-num : inférieur à *num*

num : égal à *num*

Find : actions

Les *actions* s'appliquent à chaque fichier correspondant au résultat d'une recherche.

```
tom@cafeine$ find . -not -user tom -okdir mv  
{ } /tmp/ \;  
< mv ... ./fichier2 > ? y  
tom@cafeine$ █
```

-print : action par défaut. Affichage du nom de fichier.

-ls : Affichage détaillé du fichier et de ses attributs.

-delete : Suppression du fichier.

-execdir command { } : exécution de *commande* pour chaque fichier du résultat.

-okdir command { } : idem ci-dessus, en demandant confirmation.

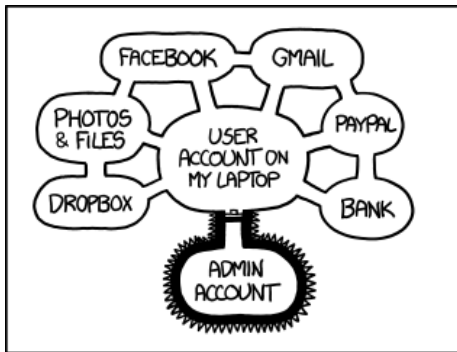
Find : erreurs courantes

find : les chemins doivent précéder l'expression : le *point de départ* de la recherche n'a pas été spécifié.

find : paramètre manquant pour exec : la commande `-exec` ne se termine pas par ;

Utilisateurs

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs**
 - Introduction
 - Groupes
 - Utilisateurs
 - Mots de passe
- 6 Permissions de fichiers
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage



IF SOMEONE STEALS MY LAPTOP WHILE I'M LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY MONEY, AND IMPERSONATE ME TO MY FRIENDS,
BUT AT LEAST THEY CAN'T INSTALL DRIVERS WITHOUT MY PERMISSION.

Introduction

- ▶ GNU/Linux propose un mécanisme sophistiqué de gestion des permissions sur les fichiers et répertoires.
- ▶ Chaque utilisateur du système se voit attribuer un identifiant unique (*uid*).
- ▶ Les programmes partagent l'*uid* de l'utilisateur qui les a lancés.
- ▶ Les droits d'accès sont déterminés en fonction de cet uid et des groupes auxquels l'utilisateur appartient.

Groupes

- ▶ Les groupes permettent de rassembler des comptes utilisateurs de manière logique, afin d'en simplifier l'administration.
- ▶ De la même manière que les comptes utilisateurs, ils sont représentés par un entier unique.
- ▶ Les informations sont stockées dans le fichier `/etc/group` :
 - 1 nom du groupe
 - 2 mot de passe (obsolète)
 - 3 GID identifiant numérique
 - 4 utilisateur1,utilisateur2,utilisateur3,...

Commande de gestion des groupes

- 1 ajout d'un groupe : `groupadd`

Commande de gestion des groupes

- 1 ajout d'un groupe : `groupadd`
- 2 suppression : `groupdel group`

Commande de gestion des groupes

- 1 ajout d'un groupe : `groupadd`
- 2 suppression : `groupdel group`
- 3 ajout d'un utilisateur à un groupe : `gpasswd -a login groupe`

Utilisateurs

- ▶ Unix est un système multi-utilisateurs.
- ▶ Chaque utilisateur est identifié par un numéro.
- ▶ root correspond à l'utilisateur administrateur. Son id est 0.
- ▶ Les comptes dont l'id est inférieur à 500 sont des comptes systèmes.
- ▶ A chaque compte utilisateur est associé un login, un répertoire personnel et un shell
- ▶ Ces informations sont renseignées dans le fichier `/etc/passwd`

Types de compte

Les types de comptes Unix sont les suivant :

root : compte administrateur, dont *l'uid* est 0.

comptes systèmes : comptes utilisateur associés à des *services*. Il n'est en général pas possible de se connecter à la machine en utilisant ces comptes, car aucun mot de passe ne leur est attribué.

compte utilisateur normal

compte désactivé : Le compte ne peut *plus* être utilisé pour se connecter.

Le fichier `/etc/passwd`

Fichier contenant les informations sur les comptes utilisateurs.

```
tom@cafeine$ cat /etc/passwd
tom:x:1234:1234:Thomas Constans,,,:/home/tom
:/bin/zsh
```

- ▶ 7 champs séparés par “ : ”
 - 1 login
 - 2 mot de passe chiffré (optionnel)
 - 3 id numérique
 - 4 id numérique de group
 - 5 Nom complet / commentaire
 - 6 répertoire personnel
 - 7 shell
- ▶ Doit être lisible par tout le monde -> le mot de passe n'est plus stocké dans ce fichier.

Création d'un compte utilisateur

Elle se fait à l'aide de la commande *useradd* : `useradd option login` avec comme option possible :

- d **repertoire** : Utiliser *repertoire* comme répertoire personnel de l'utilisateur.
 - m pour créer ce répertoire
- g **groupe_principal** : définition du groupe principal de l'utilisateur
- s **/bin/bash** shell de l'utilisateur

...

Création d'un utilisateur

La plupart des options de la commande *useradd* peut être fixée dans le fichier */etc/default/useradd*

```
GROUP=100  
HOME=/home  
INACTIVE=-1  
EXPIRE=  
SHELL=/bin/bash  
SKEL=/etc/skel  
CREATE_MAIL_SPOOL=yes
```

Suppression d'un compte

À l'aide de `userdel` option **login**

Suppression d'un compte

À l'aide de `userdel` option **login**

`-r` supprime le répertoire utilisateur

Informations d'authentification

Ces informations sont désormais déportées dans le fichier `/etc/shadow` :

```
gdm:*:14130:0:99999:7:::  
openldap:!:14138:0:99999:7:::  
toto:$1$5EpzrHQD$EqglMzfSkQDqtgLW1gHI80:14186:0:99999:7:::
```

- 1 login.
- 2 mot de passe chiffré.
- 3 nombre de jours depuis le dernier changement de mot de passe, à compter du 01/01/1970.
- 4 délai avant de pouvoir changer le mot de passe.
- 5 durée (en jours) de validité du mot de passe.
- 6 durée (en jours) avant la date de fin de validité, pendant lesquels l'utilisateur est averti.
- 7 nombre de jours, une fois passée la date de validité, au-delà duquel le compte sera désactivé.
- 8 nombre de jours, à compter du 01/01/1970, depuis que le compte est désactivé.

Informations d'authentification

Les données du fichier `/etc/shadow` sont modifiables par la commande `passwd`.

`passwd` permet de changer votre mot de passe.

`passwd utilisateur` permet de changer le mot de passe d'*utilisateur*.

`passwd --lock login` verrouille le compte de *login*.

`passwd --maxdays jours login` définit la durée de validité du mot de passe.

`passwd --status login` affiche l'état des données d'authentification de *login*.

Informations d'authentification

La commande `passwd` et la valeur par défaut de ses options peuvent être réglées via le fichier `/etc/login.defs`

MAIL_DIR	/var/spool/mail
PASS_MAX_DAYS	99999
PASS_MIN_DAYS	0
PASS_MIN_LEN	5
PASS_WARN_AGE	7
UID_MIN	1000
UID_MAX	60000
SYS_UID_MIN	201
SYS_UID_MAX	999
GID_MIN	1000
GID_MAX	60000

Utilisateurs : commandes associées

```
tom@cafeine:~$ id
uid=1234(tom) gid=1234(maison) groupes=4(adm),20(dialout),24(cdrom),29(audio),40(src),44(video),46(plugdev),105(netdev),107(fuse),109(scanner),1234(maison)
tom@cafeine:~$
```

id affiche votre identifiant numérique et les groupes auxquels vous appartenez.

groups affiche les groupes auxquels vous appartenez

w liste les utilisateurs actuellement connectés sur la machine, et ce qu'ils font.

last affiche les dernières connexions

getent passwd affiche la liste des comptes utilisateurs présents sur le système.

Utilisateurs : commandes associées

```
tom@cafeine:~$ groups
maison adm dialout cdrom audio src video plugdev netdev fuse scanner
tom@cafeine:~$
```

id affiche votre identifiant numérique et les groupes auxquels vous appartenez.

groups affiche les groupes auxquels vous appartenez

w liste les utilisateurs actuellement connectés sur la machine, et ce qu'ils font.

last affiche les dernières connexions

getent passwd affiche la liste des comptes utilisateurs présents sur le système.

Utilisateurs : commandes associées

```
tom@cafeine:~$ w
12:01:58 up 9 days, 19:27, 3 users, load average: 0.06, 0.21, 0.24
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
tom       tty7     :0            Sat21   0.00s  9:30m  0.06s /bin/bash /home/tom
tom       pts/5    :0.0         11:56   5:48m  0.22s  0.22s zsh
tom       pts/2    :0.0         11:56   0.00s  1.38s  0.00s w
tom@cafeine:~$
```

id affiche votre identifiant numérique et les groupes auxquels vous appartenez.

groups affiche les groupes auxquels vous appartenez

w liste les utilisateurs actuellement connectés sur la machine, et ce qu'ils font.

last affiche les dernières connexions

getent passwd affiche la liste des comptes utilisateurs présents sur le système.

Utilisateurs : commandes associées

```

tom      tty1      Tue Jul 29 19:06 - 19:07 (00:01)
tom      tty1      Tue Jul 29 19:06 - 19:06 (00:00)
reboot   system boot 2.6.25-2-powerpc Tue Jan 5 08:05 - 19:30 (-11517+-20;
tom      tty7      :0        Tue Jul 29 18:26 - crash (11517+21;57
reboot   system boot 2.6.25-2-powerpc Tue Jan 5 07:24 - 19:30 (-11517+-20;
tom      tty7      :0        Tue Jul 29 17:50 - crash (11517+21;53
reboot   system boot 2.6.25-2-powerpc Tue Jan 5 06:49 - 19:30 (-11517+-19;
tom      tty7      :0        Mon Jul 28 11:13 - crash (11519+03;54
reboot   system boot 2.6.25-2-powerpc Mon Jan 4 00:12 - 19:30 (-11516+-13;
tom      tty7      :0        Sun Jul 27 20:56 - down (00:35)
reboot   system boot 2.6.25-2-powerpc Sun Jan 3 09:55 - 21:31 (-11517+-20;
reboot   system boot 2.6.25-2-powerpc Sun Jan 3 01:50 - 21:31 (-11517+-12;
reboot   system boot 2.6.25.6      Sun Jan 3 01:46 - 01:48 (00:01)
tom      tty7      :0        Sun Jan 3 00:45 - down (-11517+-20;
reboot   system boot 2.6.25.6      Sun Jan 3 00:44 - 12:44 (-11517+-20;
reboot   system boot 2.6.25.6      Sat Jan 2 01:39 - 21:30 (-11522+-12;

```

id affiche votre identifiant numérique et les groupes auxquels vous appartenez.

groups affiche les groupes auxquels vous appartenez

w liste les utilisateurs actuellement connectés sur la machine, et ce qu'ils font.

last affiche les dernières connexions

getent passwd affiche la liste des comptes utilisateurs présents sur le système.

Utilisateurs : commandes associées

```
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailng List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
messagebus:x:101:103:/:/var/run/dbus:/bin/false
tom:x:1234:1234:Thomas Constans,,,:/home/tom:/bin/zsh
libuuid:x:102:104:/:/var/lib/libuuid:/bin/sh
statd:x:104:65534:/:/var/lib/nfs:/bin/false
```

id affiche votre identifiant numérique et les groupes auxquels vous appartenez.

groups affiche les groupes auxquels vous appartenez

w liste les utilisateurs actuellement connectés sur la machine, et ce qu'ils font.

last affiche les dernières connexions

getent passwd affiche la liste des comptes utilisateurs présents sur le système.

Permissions de fichiers

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers**
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

Permissions

Il y a 3 droits principaux, avec une valeur numérique et un symbole :

symbole	valeur	description

Permissions

Il y a 3 droits principaux, avec une valeur numérique et un symbole :

symbole	valeur	description
r	4	lecture

Permissions

Il y a 3 droits principaux, avec une valeur numérique et un symbole :

symbole	valeur	description
r	4	lecture
w	2	écriture

Permissions

Il y a 3 droits principaux, avec une valeur numérique et un symbole :

symbole	valeur	description
r	4	lecture
w	2	écriture
x	1	exécution / accès

Permissions sur les fichiers et les répertoires

Si elles sont les même, les permissions sur les fichiers et les répertoires n'ont pas le même effet :

permissions	fichier	répertoire
lecture	le contenu du fichier peut être lu	le contenu du répertoire peut être affiché
écriture	le fichier peut être modifié	les fichiers et répertoire présents dans le répertoire peuvent être supprimés .
exécution	le fichier peut être exécuté	on peut utiliser le répertoire comme répertoire courant.

Permissions

Les permissions s'appliquent à **3** entités différentes :

Propriétaire (u) : l'utilisateur à l'origine du fichier

Groupe Propriétaire (g) : groupe principal auquel appartient le propriétaire du fichier

Autres (o) : le reste des utilisateurs

Permissions

Les permissions s'appliquent à **3** entités différentes :

Propriétaire (u) : l'utilisateur à l'origine du fichier

Groupe Propriétaire (g) : groupe principal auquel appartient le propriétaire du fichier

Autres (o) : le reste des utilisateurs

```
tom@workine:~$ ls -l ~
drwx----- 2 tom maison 4,0K jui 23 08:33 PDF
drwxr-xr-x 2 tom maison 4,0K jan  9 2008 Templates
-rw-r--r-- 1 tom maison 8,0K jui 23 09:47 console.png
drwxr-xr-x 2 tom maison 4,0K jui 18 11:29 public_html
...
```


Commandes associées

Les commandes suivantes permettent de modifier les permissions d'un fichier et de modifier son propriétaire :

chmod mode fichier : permet de modifier les permissions. *mode* peut être spécifié en mode numérique ou symbolique.

```
chmod 644 permissions.pdf
```

ou

```
chmod u+rw,g+r,o-r permissions.pdf
```

Commandes associées

Les commandes suivantes permettent de modifier les permissions d'un fichier et de modifier son propriétaire :

chmod mode fichier : permet de modifier les permissions. *mode* peut être spécifié en mode numérique ou symbolique.

chown user fichier : rend *user* propriétaire du fichier.

```
chown tom permissions.pdf
```

Commandes associées

Les commandes suivantes permettent de modifier les permissions d'un fichier et de modifier son propriétaire :

chmod mode fichier : permet de modifier les permissions. *mode* peut être spécifié en mode numérique ou symbolique.

chown user fichier : rend *user* propriétaire du fichier.

chgrp groupe fichier : rend le groupe *group* propriétaire du fichier

```
chgrp stagiaires permissions.pdf
```

Déterminer les permissions à appliquer

L'utilisateur "*tom*" souhaite accorder les permissions suivantes sur le répertoire */home/tom/Exercices* :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Déterminer les permissions à appliquer

L'utilisateur "*tom*" souhaite accorder les permissions suivantes sur le répertoire */home/tom/Exercices* :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R (4)			
W (2)			
X (1)			

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R(4)	4		
W(2)	2		
X(1)	1		

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R(4)	4	4	
W(2)	2	0	
X(1)	1	1	

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ **Aucun accès pour les autres utilisateurs**

Valeur	User	Group	Other
R(4)	4	4	0
W(2)	2	0	0
X(1)	1	1	0

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R(4)	4	4	0
W(2)	2	0	0
X(1)	1	1	0

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R(4)	4	4	0
W(2)	2	0	0
X(1)	1	1	0
	7		

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R(4)	4	4	0
W(2)	2	0	0
X(1)	1	1	0
	7	5	

Déterminer les permissions à appliquer

L'utilisateur "tom" souhaite accorder les permissions suivantes sur le répertoire `/home/tom/Exercices` :

- ▶ Tous les accès pour le propriétaire (lui-même)
- ▶ Accès en lecture pour les membres de son groupe principal
- ▶ Aucun accès pour les autres utilisateurs

Valeur	User	Group	Other
R(4)	4	4	0
W(2)	2	0	0
X(1)	1	1	0
	7	5	0

Ce qui donne `chmod 750 /home/tom/Exercices`

Les permissions spéciales

Elles sont généralement à 0, mais peuvent être utiles dans certains cas :

La permission Suid : permet d'exécuter une commande avec l'identité de son propriétaire.

La permission sGid : permet d'exécuter une commande avec l'identité de son groupe propriétaire.

La permission sGid , appliquée sur un *répertoire*, rend le groupe propriétaire du répertoire *propriétaire* des fichiers et répertoires qui y seront créés.

La permission sTicky , appliquée sur un répertoire, empêche les utilisateurs de pouvoir supprimer les fichiers dont ils ne sont pas propriétaires.

Les permissions spéciales

Le calcul de ces permissions est similaire à celui des permissions standards.

Valeur		U ser	G roup	O ther
4				
2				
1				

Les permissions spéciales

Le calcul de ces permissions est similaire à celui des permissions standards.

Valeur	spec	U ser	G roup	O ther
4		r	r	-
2		w	-	-
1		x	x	-

Les permissions spéciales

Le calcul de ces permissions est similaire à celui des permissions standards.

Valeur	spec	U ser	G roup	O ther
4	Suid	r	r	-
2	sGid	w	-	-
1		x	x	-

Les permissions spéciales

Le calcul de ces permissions est similaire à celui des permissions standards.

Valeur	spec	U ser	G roup	O ther
4	Suid	r	r	-
2	sGid	w	-	-
1	sTicky	x	x	-

Les permissions spéciales

Le calcul de ces permissions est similaire à celui des permissions standards.

Valeur	spec	User	Group	Other
4	Suid	r	r	-
2	sGid	w	-	-
1	sTicky	x	x	-

Si on veut appliquer la permissions *sTicky* au répertoire */home/tom/Exercices* on aura :

Les permissions spéciales

Le calcul de ces permissions est similaire à celui des permissions standards.

Valeur	spec	User	Group	Other
4	Suid	r	r	-
2	sGid	w	-	-
1	sTicky	x	x	-

Si on veut appliquer la permissions *sTicky* au répertoire */home/tom/Exercices* on aura :

```
chmod 1750 /home/tom/Exercices
```

Les permissions par défaut

Les permissions affectées à un nouveau répertoire ou fichier dépendent de la valeur de *l'umask*.

- ▶ Pour un répertoire, les droits par défaut sont à 0777 - l'umask du créateur.
- ▶ Pour un fichier, ses droits sont à 0666 - l'umask du créateur.

Ainsi, avec un *umask* à 0027, un nouveau fichier aura pour permissions :

Les permissions par défaut

Les permissions affectées à un nouveau répertoire ou fichier dépendent de la valeur de *l'umask*.

- ▶ Pour un répertoire, les droits par défaut sont à 0777 - l'umask du créateur.
- ▶ Pour un fichier, ses droits sont à 0666 - l'umask du créateur.

Ainsi, avec un *umask* à 0027, un nouveau fichier aura pour permissions :
0666 permission de départ

Les permissions par défaut

Les permissions affectées à un nouveau répertoire ou fichier dépendent de la valeur de *l'umask*.

- ▶ Pour un répertoire, les droits par défaut sont à 0777 - l'umask du créateur.
- ▶ Pour un fichier, ses droits sont à 0666 - l'umask du créateur.

Ainsi, avec un *umask* à 0027, un nouveau fichier aura pour permissions :
0666 permission de départ
-0027 umask

Les permissions par défaut

Les permissions affectées à un nouveau répertoire ou fichier dépendent de la valeur de *l'umask*.

- ▶ Pour un répertoire, les droits par défaut sont à 0777 - l'umask du créateur.
- ▶ Pour un fichier, ses droits sont à 0666 - l'umask du créateur.

Ainsi, avec un *umask* à 0027, un nouveau fichier aura pour permissions :

0666 permission de départ

-0027 umask

=0640 permissions effectives

En résumé

Pour mettre en place les permissions souhaitées :

- ▶ On joue sur l'utilisateur et/ou le groupe propriétaire du fichier

En résumé

Pour mettre en place les permissions souhaitées :

- ▶ On joue sur l'utilisateur et/ou le groupe propriétaire du fichier (à l'aide des commandes `chown` et `chgrp`)
- ▶ On applique à ces propriétaires les permissions

En résumé

Pour mettre en place les permissions souhaitées :

- ▶ On joue sur l'utilisateur et/ou le groupe propriétaire du fichier (à l'aide des commandes `chown` et `chgrp`)
- ▶ On applique à ces propriétaires les permissions

En résumé

Pour mettre en place les permissions souhaitées :

- ▶ On joue sur l'utilisateur et/ou le groupe propriétaire du fichier (à l'aide des commandes `chown` et `chgrp`)
- ▶ On applique à ces propriétaires les permissions (à l'aide de `chmod`)

Gestion des applications

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers
- 7 Gestion des applications**
 - Introduction
 - Dépendances
 - Gestionnaire de paquets
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage

INSTALL.SH

```
#!/bin/bash
```

```
pip install "$1" &  
easy_install "$1" &  
brew install "$1" &  
npm install "$1" &  
yum install "$1" & dnf install "$1" &  
docker run "$1" &  
pkg install "$1" &  
apt-get install "$1" &  
sudo apt-get install "$1" &  
steamcmd +app_update "$1" validate &  
git clone https://github.com/"$1"/"$1" &  
cd "$1"; ./configure; make; make install &
```

Introduction

Pour rappel, une *distribution* GNU/Linux propose

- ▶ Un noyau Linux
- ▶ Un ensemble d'utilitaire et de services (ex, support, media d'installation, mises à jour, ...)
- ▶ Une collection de logiciels intégrée au système

Bibliothèques partagées et dépendances

Application a

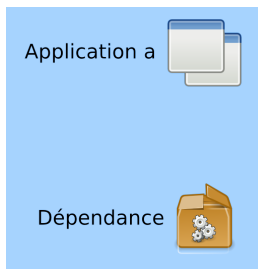


Dépendance



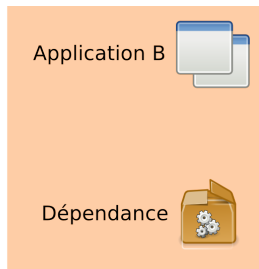
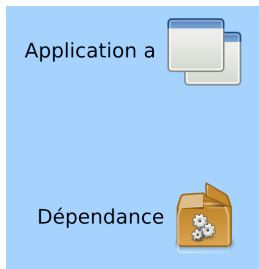
- ▶ 1 logiciel nécessite une fonction proposée par le paquet logiciel “dépendance”.
- ▶ La 1^{re} solution est de packager les 2 ensemble.
- ▶ Si un 2^e logiciel demande la même *dépendance*, cette solution devient suboptimale.
- ▶ Les logiciels GNU/Linux sont basés sur le principe de la *bibliothèque de fonctions partagées*.
- ▶ “*dépendance*” est packagée indépendamment. On obtient un système modulaire.

Bibliothèques partagées et dépendances



- ▶ 1 logiciel nécessite une fonction proposée par le paquet logiciel “dépendance”.
- ▶ La 1^{re} solution est de packager les 2 ensemble.
- ▶ Si un 2^e logiciel demande la même *dépendance*, cette solution devient suboptimale.
- ▶ Les logiciels GNU/Linux sont basés sur le principe de la *bibliothèque de fonctions partagées*.
- ▶ “*dépendance*” est packagée indépendamment. On obtient un système modulaire.

Bibliothèques partagées et dépendances



- ▶ 1 logiciel nécessite une fonction proposée par le paquet logiciel “dépendance”.
- ▶ La 1^{re} solution est de packager les 2 ensemble.
- ▶ Si un 2^e logiciel demande la même *dépendance*, cette solution devient suboptimale.
- ▶ Les logiciels GNU/Linux sont basés sur le principe de la *bibliothèque de fonctions partagées*.
- ▶ “*dépendance*” est packagée indépendamment. On obtient un système modulaire.

Bibliothèques partagées et dépendances

Application a



Application B

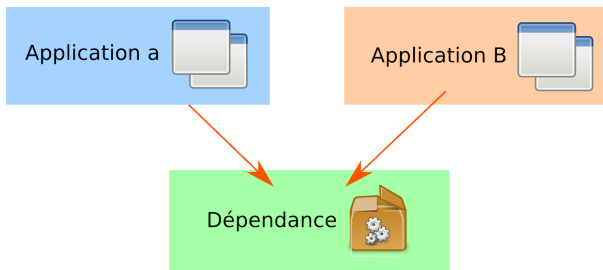


Dépendance



- ▶ 1 logiciel nécessite une fonction proposée par le paquet logiciel “dépendance”.
- ▶ La 1^{re} solution est de packager les 2 ensemble.
- ▶ Si un 2^e logiciel demande la même *dépendance*, cette solution devient suboptimale.
- ▶ Les logiciels GNU/Linux sont basés sur le principe de la *bibliothèque de fonctions partagées*.
- ▶ “*dépendance*” est packagée indépendamment. On obtient un système modulaire.

Bibliothèques partagées et dépendances



- ▶ 1 logiciel nécessite une fonction proposée par le paquet logiciel “dépendance”.
- ▶ La 1^{re} solution est de packager les 2 ensemble.
- ▶ Si un 2^e logiciel demande la même *dépendance*, cette solution devient suboptimale.
- ▶ Les logiciels GNU/Linux sont basés sur le principe de la *bibliothèque de fonctions partagées*.
- ▶ “*dépendance*” est packagée indépendamment. On obtient un système modulaire.

Question

Question :

Comment s'assurer qu'une application sera installée avec toutes ses dépendances

Les gestionnaires de paquets

Les “package manager” sont une des particularités de chaque distribution.
Leur rôle est :

- Installer** les logiciels
- Résoudre** les dépendances
- Désinstaller** les logiciels et leurs dépendances inutilisées.
- Maintenir** le système à jour.
- Rechercher** des logiciels

Les gestionnaires de paquets

Les “package manager” sont une des particularités de chaque distribution.
Leur rôle est :

Installer les logiciels

Résoudre les dépendances

Désinstaller les logiciels et leurs dépendances inutilisées.

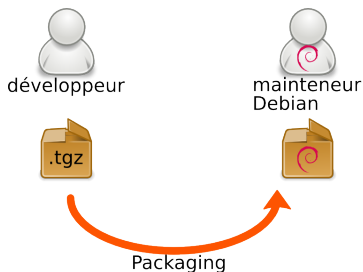
Maintenir le système à jour.

Rechercher des logiciels

Apt est le gestionnaire de paquet de GNU/Debian.

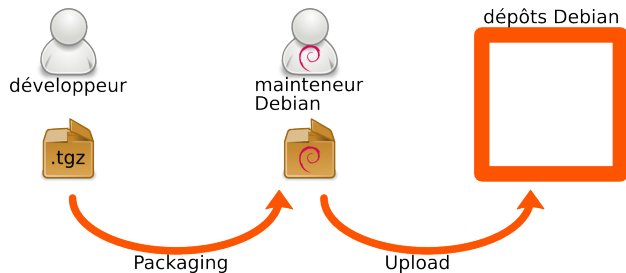
Yum est celui des distributions RedHat et CentOS

les paquets



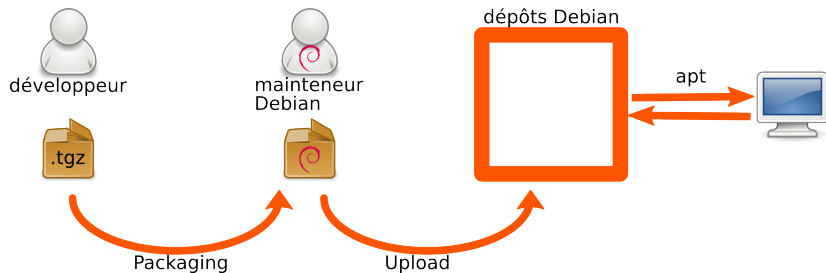
- ▶ Un logiciel (libre) est “*empaqueté*” au format “rpm”
- ▶ Il est ensuite inclu dans les dépôts CentOS/Debian
- ▶ Et devient donc disponible par l'intermédiaire de *yum*.

les paquets



- ▶ Un logiciel (libre) est “*empaqueté*” au format “rpm”
- ▶ Il est ensuite inclu dans les dépôts CentOS/Debian
- ▶ Et devient donc disponible par l'intermédiaire de *yum*.

les paquets



- ▶ Un logiciel (libre) est “*empaqueté*” au format “rpm”
- ▶ Il est ensuite inclu dans les dépôts CentOS/Debian
- ▶ Et devient donc disponible par l'intermédiaire de *yum*.

Le format .rpm

un paquet debian comprend :

Un en-tête , qui contient des informations sur le paquet

Une archive , qui sera extraite lors de l'installation.

Des scripts , qui seront exécutés lors de l'installation et la désinstallation

Configuration des dépôts

La connexion entre notre gestionnaire de paquet et les dépôts se fait via les fichiers de `/etc/yum.repos.d`. Configurer un nouveau dépôt revient à créer un nouveau fichier de définition dans ce répertoire. Cela peut se faire en installant un paquet. Voir par exemple le paquet `epel-release`.

Attention, les paquets d'un dépôt sont généralement signés. Si on configure un nouveau dépôt, il ne faut pas oublier la clé correspondante, sinon yum ne pourra pas vérifier l'intégrité des paquets lors de leur installation.

Interrogation de la base des paquets locaux

la commande *rpm* va principalement être utilisé pour interroger la base des paquets installés :

`rpm -qa` liste de tous les paquets installés.

`rpm -qi paquet` affiche des informations sur un paquet.

`rpm -ql paquet` affiche le contenu d'un paquet.

Gestion de paquets avec yum

yum install package : installe *package* et ses dépendances

yum search motif recherche un package dont la description ou le résumé contient "*motif*".

yum remove package : désinstalle *package* et ses dépendances

yum update : installation des dernières mises à jour disponibles.

yum provides /chemin : cherche les paquets contenant */chemin*

Gestion de l'historique

La sous-commande *history* de yum permet de revoir et au besoin d'annuler toutes les commandes yum.

yum history list - affiche l'historique des installations, déinstallation, mise à jour de paquets

yum history info 234 affiche des infos (principalement la liste des paquets concernés) par l'opération 234

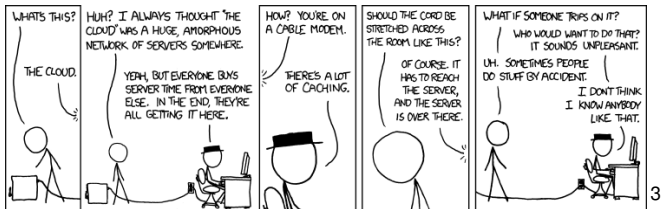
yum history undo 234 annule les actions réalisées par l'opération 234 on désinstalle ce qui aura été installé)

yum history rollback 234 on annule toutes les opérations de la plus récente jusqu'à (y compris) l'opération 234.

Réseau TCP/IP sous GNU/Linux

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux**
 - Configuration TCP/IP
 - Parefeu
- 9 Arrêt et démarrage

Environnement Linux



Nom de machine

Modifié par la commande *hostnamectl*.

hostnamectl affiche le nom actuel.

hostnamectl set-hostname nom modifie le nom de la machine.

Vérification de la configuration

La configuration des interfaces est visible par la commande `ip a`

```
tom@cafeine$ /sbin/ifconfig eth2
eth2      Link encap:Ethernet  HWaddr 00:14:51:db:cc:ee

          inet adr:192.168.10.102  Bcast:192.168.10.255
          Masque:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

```
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536
   qdisc noqueue state UNKNOWN group
   default qlen 1000
   link/loopback 00:00:00:00:00:00 brd
     00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft
       forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft
```

Configuration des interfaces

La gestion du réseau sur les distributions récentes est assurée par *NetworkManager*. On ne touche plus aux fichiers de configuration, on passe désormais par différentes interfaces :

- 1 l'interface graphique, si on est en environnement graphique
- 2 l'interface *nmtui* (Network Manager Text User Interface)
- 3 ou l'interface *nmcli*

Cette dernière commande va principalement servir à lister, activer ou désactiver une connexion :

- ▶ `nmcli c`
- ▶ `nmcli c up enp0s3`
- ▶ `nmcli c down enp0s3`

Configuration de la résolution locale

```
cat /etc/hosts
127.0.0.1    localhost localhost.
             localdomain localhost4 localhost4.
             localdomain4 cafeine foobar
#:::1       localhost localhost.
             localdomain localhost6 localhost6.
             localdomain6
192.168.10.96 cafeine
192.168.10.99 turbine
```

Configuration des serveurs de noms

Les adresses des serveurs de nom sont renseignées dans le fichier `/etc/resolv.conf` :

```
search opendoor.fr
nameserver 192.168.10.254
```

nameserver permet d'indiquer l'adresse du serveur DNS. il peut y en avoir plusieurs (3).

domain indique le nom de domaine local.

search liste des domaines qui seront accolés au critère de recherche, si celui-ci n'est pas un fqdn.

options permet de modifier le comportement du résolveur

Les directives *domain* et *search* sont mutuellement exclusives.

Configuration du routage

Le routage indique à une machine ses possibilités d'accès sur l'extérieur. On distingue 3 types :

minimal : lors de la configuration des interfaces, une route est automatiquement créée vers le réseau local.

statique : Les tables de routages sont maintenues manuellement.

dynamique : Des protocoles spécifiques maintiennent à jour les tables de routage.

Affichage de la table de routage

La commande `ip r(oute)` permet d'afficher la table de routage locale.

```
# ip r
default via 10.251.255.254 dev wlp3s0
    proto dhcp metric 20600
10.251.0.0/16 dev wlp3s0 proto kernel
    scope link src 10.251.150.229 metric
    600
100.0.0.0/8 dev vboxnet0 proto kernel
    scope link src 100.0.0.1
172.17.0.0/16 dev docker0 proto kernel
    scope link src 172.17.0
```

Table de routage

Les informations renvoyées par la commande `ip r` sont :

default ou réseau : passerelle par défaut ou réseau de destination.

Passerelle : adresse du routeur.

dev wlp3s0 : périphérique à utiliser

Metric : *coût* de la route.

Ajout et modification d'une route

La table de routage peut être modifiée lors de l'activation d'une interface, par l'intermédiaire de la directive *gateway* du fichier */etc/network/interfaces*.

La commande *ip r(oute) add* permet de modifier la table de routage :

`ip r add 10/8 dev eth0` ajout de la route vers le réseau local

`ip r add default via 10.0.0.254` ajout de la passerelle par défaut.

`ip r del default` suppression de la route par défaut.

Parefeu

S'il s'appuie toujours sur netfilter/iptables, le parefeu sur RedHat CentOS est désormais géré par *firewalld*.

Zones

Les zones firewalld

Arrêt et démarrage

- 1 Paramétrage de l'environnement
- 2 Vi
- 3 Sudo
- 4 Gérer le système de fichiers
- 5 Utilisateurs
- 6 Permissions de fichiers
- 7 Gestion des applications
- 8 Réseau TCP/IP sous GNU/Linux
- 9 Arrêt et démarrage**
 - Les grandes étapes du démarrage
 - Systemd
 - Gestion des services
 - Unit-files personnalisées

Les grandes étapes

Un système GNU/Linux passe par les étapes suivantes lors du démarrage :

- 1 Bios
- 2 Chargeur de démarrage
- 3 Noyau
- 4 Détection et activation du périphérique “*racine*”.
- 5 Lancement du processus *systemd*.
- 6 Exécution des unités *systemd*
- 7 Lancement de l'utilitaire de connexion.

Le chargeur de démarrage

Le rôle du chargeur de démarrage est de *détecter* et de *charger* le noyau.

- ▶ Il y a 2 gestionnaire de démarrage : 1 rudimentaire stocké dans le *MBR*, un autre stocké situé sur une partition du disque.
- ▶ Sous Linux, le chargeur de démarrage est *grub*
- ▶ Ils proposent un menu interactif permettant de choisir les différents noyaux présents sur le système.
- ▶ Une erreur de configuration peut entraîner un problème de démarrage.

Grub

Le **G**rand **U**nified **B**oot loader

- ▶ Son fichier de configuration est */boot/grub/menu.lst*
- ▶ Mini shell
- ▶ Pas de réinstallation suite à modification de configuration
- ▶ Plus souple d'utilisation

Le noyau

Le noyau est le premier processus (dont le pid est 0) du système.

- ▶ Il initialise les principaux périphériques en chargeant les pilotes associés.
- ▶ Le noyau détecte et rend accessible le *système de fichier racine*.
- ▶ Il lance un certain nombre de processus chargés de gérer certaines parties du système ([kswapd], [kjournald], ...).
- ▶ Enfin, il exécute le 1^{er} processus utilisateur : *init*.

Systemd

Systemd a remplacé, sur les distributions récentes *init*. Il s'occupe de la configuration de la machine (paramétrage réseau, nom de machine, montage des systèmes de fichiers, etc.) et du lancement des différents services. Cela se fait au moyen de fichiers d'unités, ou "*unit files*" qui ont remplacé les anciens script d'*init*

Les unit files

Les unit files sont de différent type (suivant l'extension) et offrent différentes fonctions.

les plus utilisées sont les units de type service.

La tendance actuelle est à la convergence de la configuration d'un système (montage, tâches planifiées, ...) via systemd.

Les unit-files par défaut des applications et services sont dans `/usr/lib/systemd/system/`.

Les unit-files personnalisées sont à mettre dans `/etc/systemd/system`.

À l'installation d'une nouvelle unit-file, il faut recharger systemd :
`systemctl daemon-reload`.

La gestion des services

Les services se gèrent de la manière suivante :

arrêt et redémarrage `systemctl start|stop|restart unit-file-name.service - ex`
`systemctl restart httpd`

Activation et désactivation `systemctl enable|disable unit-file-name.service - ex`
`systemctl enable httpd`

Unit-files personnalisées

Créer un fichier `/etc/systemd/system/foo.service` :

```
[Unit]
Description=script de démarrage foo
After=network.target remote-fs.target nss-
lookup.target

[Service]
Type=notify
WorkingDirectory=/opt/bin
ExecStart=/opt/bin/foo start

[Install]
WantedBy=multi-user.target
```

Ne pas hésiter à consulter les pages de man `systemd.service` et `systemd.unit`

Commandes associées

`systemctl` sans argument, liste toutes les unités chargées, y compris celles en **échec**

`systemd-analyze` permet d'identifier les unités qui ralentissent le démarrage du système

Licence



CC-BY-NC-SA

Ce support est mis à disposition selon le *Contrat Paternité - Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique 2.0 France* disponible en ligne ici ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.