

Administration système

M4101C

2ème année - S4, cours - 1/3
2017-2018

Marcel.Bosc@iutv.univ-paris13.fr

table des matières

- présentation du métier
- gestion des utilisateurs
- machine distantes
- gestion des logiciels
- les services

présentation du métier



le métier



serveurs



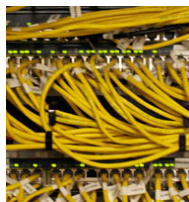
parcs de machines



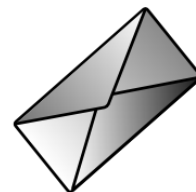
grand
nombre d'utilisateurs



sécurité



réseaux



e-mail

L'administration système est un domaine varié, offrant de nombreuses opportunités d'emploi.

On peut, par exemple, être amené à s'occuper d'un grand nombre de machines utilisateur, en mettant en place les systèmes d'automatisation nécessaire pour les installer, les maintenir à jour et les dépanner.

D'autres administrateurs systèmes peuvent, par exemple, se spécialiser dans la maintenance d'un nombre plus réduit de serveurs, critiques pour un service web commercial.

Dans tous les cas, la sécurité est une composante importante du métier.

le métier



Wikipedia : Administrateur systèmes

L'administrateur système est souvent spécialisé dans un système d'exploitation ou un dans un domaine particulier. Les compétences sur plusieurs systèmes sont difficiles à acquérir (et maintenir) mais appréciées.

Il est important de savoir gérer et prioriser les demandes d'utilisateur pour éviter la surcharge.

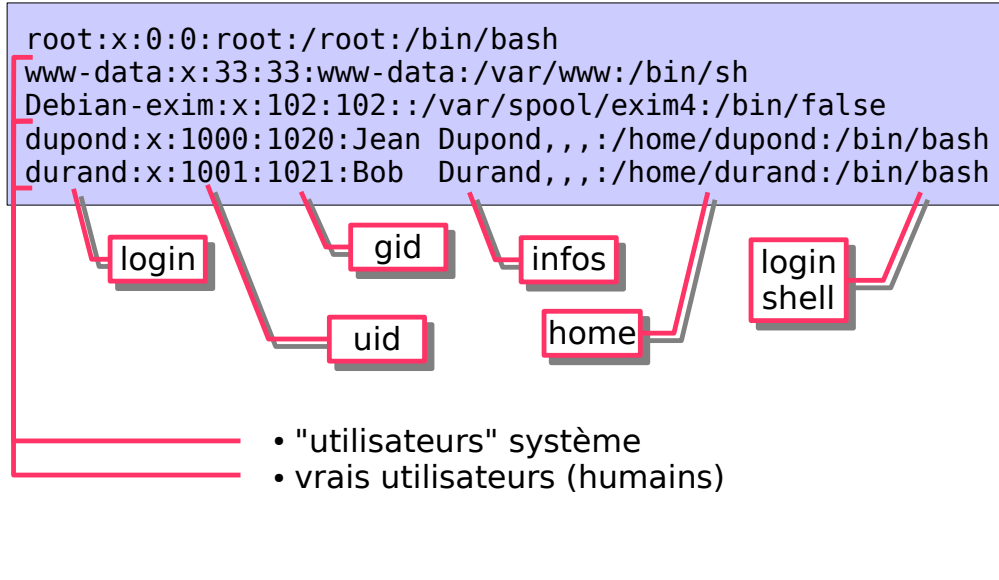
Tous les métiers de l'informatique demandent un important de travail de veille pour suivre les technologies en rapide évolution. Un informaticien qui ne se tient pas à jour se retrouvera à l'écart. Il est important de le faire comprendre à l'employeur.

1ère partie

gestion des utilisateurs



le fichier /etc/passwd



Dans un système UNIX les utilisateurs sont définis dans le fichier /etc/passwd

Contrairement à ce que son nom n'indique, ce fichier ne contient PAS de mot de passe. Il est lisible par tout le monde.

Il est décomposé en lignes. Chaque ligne correspond à un utilisateur. Chaque ligne est elle même

décomposée en plusieurs champs séparés par des ":".

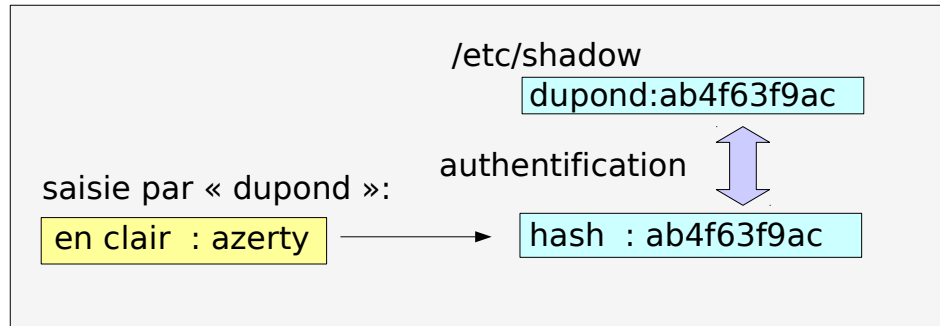
uid= user id (numéro de l'utilisateur)

gid= numéro du groupe principal (mais on peut être dans plusieurs groupes ... voir /etc/group).

La plupart des utilisateurs ne sont pas des "vrais" utilisateurs (humains), mais des utilisateurs nécessaire au fonctionnement de divers services du système.

Authentification : mot de passe

/etc/passwd lisible par tous!

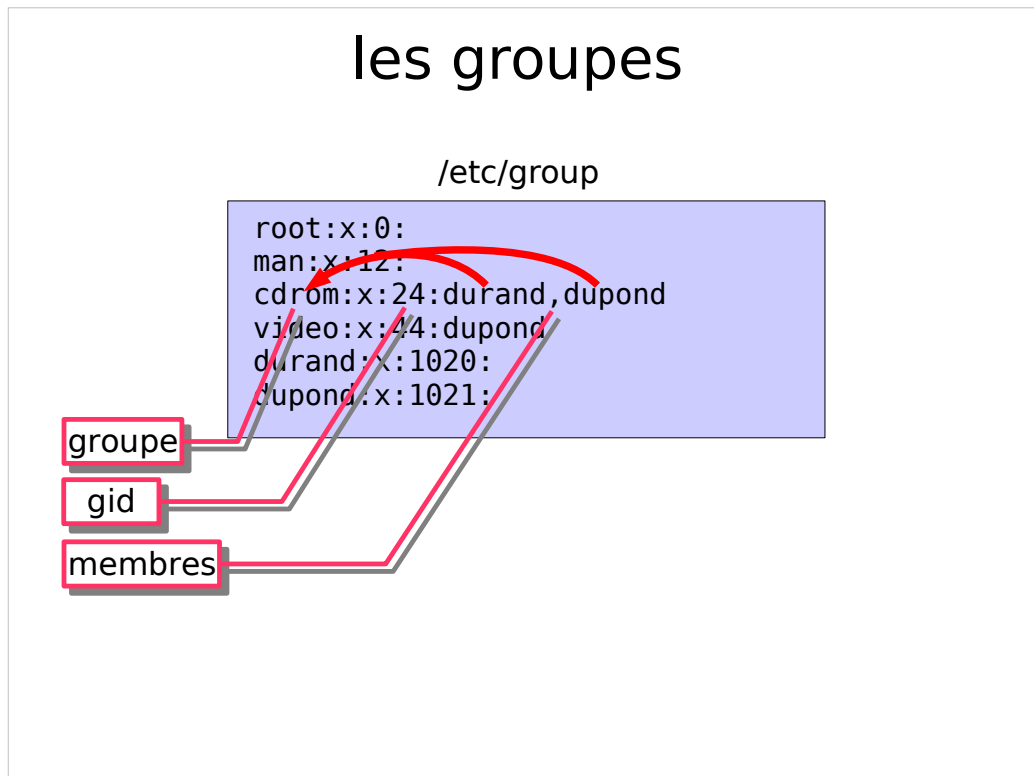


pas de mot de passe
enregistré en clair



Les mots de passe des utilisateurs sont hachés ("brouillés" avec SHA). Ils sont stockés dans le fichier `/etc/shadow`. Ce hachage est irréversible, on ne peut pas retrouver la version en clair à partir de la version hachée. Si un pirate obtient accès à `/etc/shadow`, il ne pourra pas simplement retrouver les mots de passe. Quand un utilisateur veut se connecter au système, le mot de passe qu'il a rentré est haché et la version hachée est comparée à la valeur dans `/etc/shadow`. Si les deux sont identiques, le système autorise la connexion.

les groupes



Les utilisateurs peuvent faire partie de différents groupes. Les groupes sont définis dans `/etc/group`

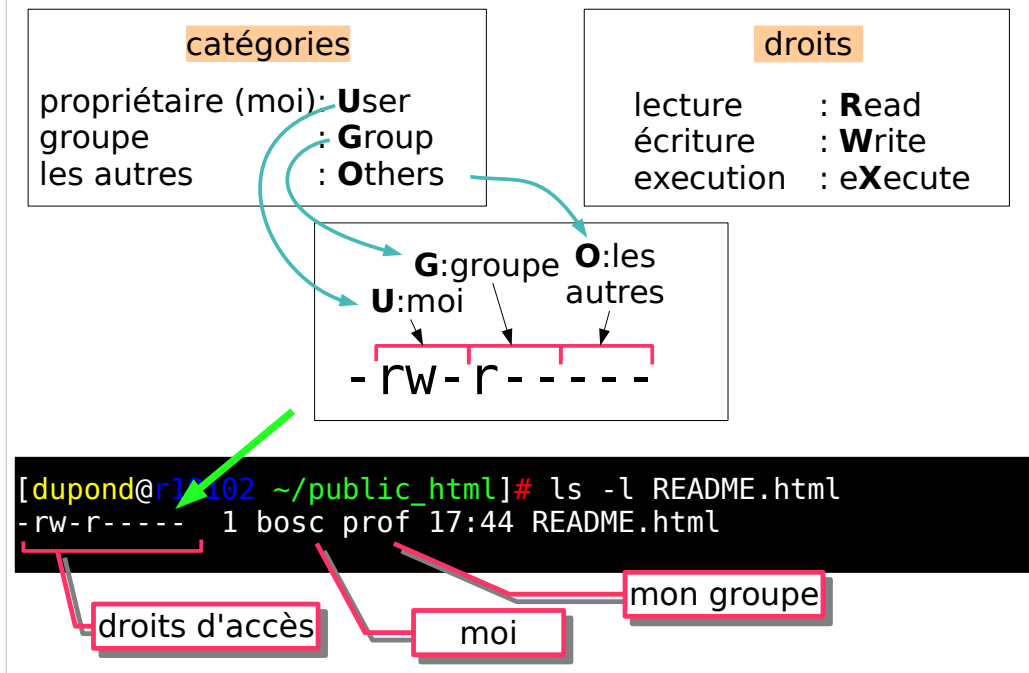
Chaque ligne définit un groupe.

Elle commence par le nom du groupe et finit par la liste des utilisateurs faisant partie du groupe.

Dans cet exemple: `durand` et `dupond` font partie du groupe "`cdrom`".

Attention: le système crée un groupe qui a le même nom que chaque utilisateur. Sur la dernière ligne, le mot "`dupond`" correspond au groupe "`dupond`" et pas l'utilisateur "`dupond`".

droits d'accès aux fichiers



Chaque fichier a un propriétaire (ici: bosc) et un groupe (ici: prof).

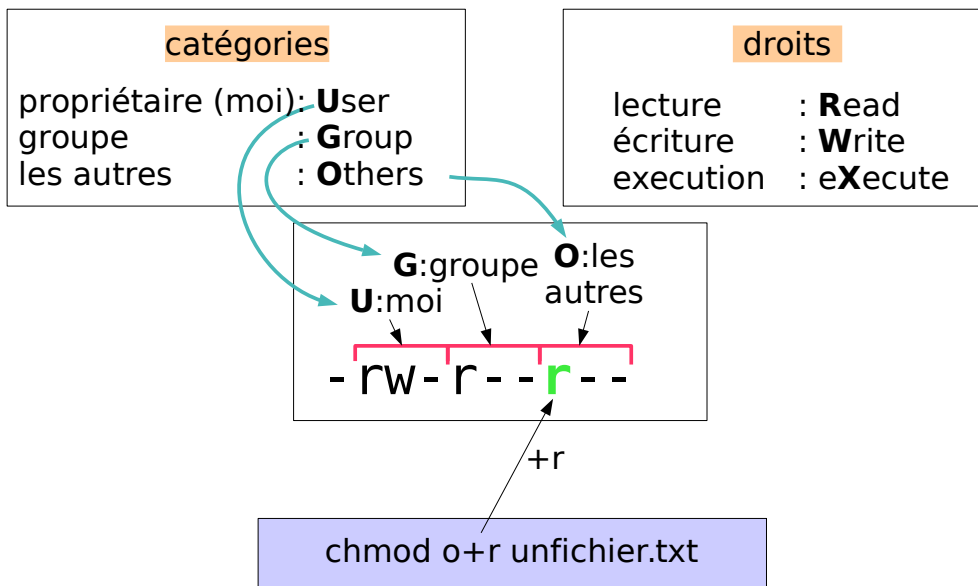
Pour un fichier, on raisonne sur 3 catégories :
User, Group, et Others. Dans cet exemple:
User=bosc, Group=prof

Pour chacune de ces 3 catégories on définit 3 droits: r,w,x

La commande `ls -l` affiche en début de ligne les 3 droits des 3 catégories:

`-rwxrwxrwx`

changer les droits : chmod



On modifie les droits d'accès à un fichier avec la commande chmod comme ceci:

chmod categories+/-droits fichier-exemple.txt

On peut changer le propriétaire d'un fichier:

sudo chown durand fichier-exemple.txt

On peut changer le groupe d'un fichier:

sudo chgrp etudiant fichier-exemple.txt

Remarquez que ces 2 opérations ne peuvent être faites que par root.

serveurs et utilisateurs

les serveurs ont des utilisateurs
spécifiques à droits limités

processus
apache piraté → fichier
sensible

bien réfléchir aux droits d'accès
limiter les droits en écriture

Chaque processus est associé à un utilisateur. Le processus a tous les droits de cet utilisateur.

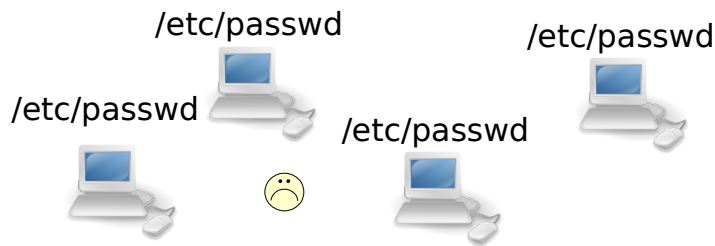
Les services (web, mail, ...) sont exposés à des tentatives d'intrusion. S'il y a une faille de sécurité, un pirate peut prendre le contrôle d'un processus d'un service.

La gravité de cette intrusion dépend alors des droits de l'utilisateur associé à ce processus.

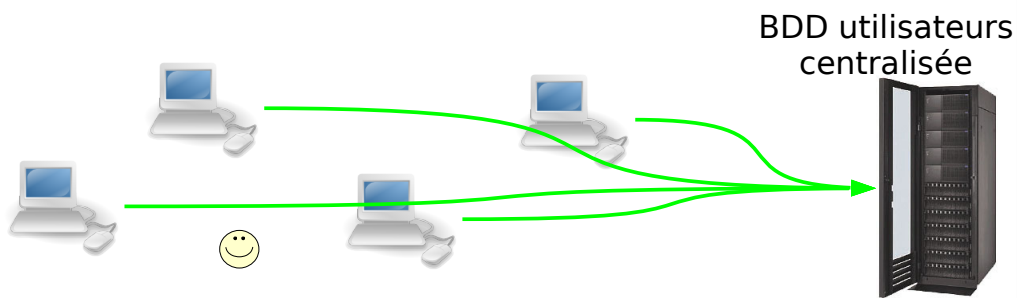
Si le service tourne en tant que "root", la machine est totalement compromise...

Sur Debian, le service web (apache) est associé à l'utilisateur www-data. Cet utilisateur a peu de droits. Faites attention à ne pas lui donner des droits d'écriture excessifs.

parcs de machines



LDAP: Lightweight Directory Access Protocol



La gestion des utilisateurs par /etc/passwd, /etc/shadow... n'est pas pratique pour un grand parc de machines.

Il faudrait créer et mettre à jours /etc/passwd, /etc/shadow... sur toutes les machines.

On utilise alors des systèmes comme LDAP permettant de gérer de manière centralisée des informations sur les utilisateurs (authentification, mais aussi d'autres informations: numéro de tél., numéro de bureau, ...)

machine distantes

machine distantes



- hébergement dans salles spécialisés
climatisation , sécurité, incendie, électricité, réseau
- location : serveurs dédiés (OVH, Online...)
- accès physique limité
- ➔ • travail à distance

Wikipedia : Data center, serveur dédié

Les serveurs doivent tourner sans interruption. C'est un enjeu important. Par exemple un site web de vente en ligne doit tourner en permanence. Toute interruption peut occasionner des pertes financières et une perte de crédibilité. Les serveurs sont donc rangés dans des armoires (racks, baies) qui se trouvent dans des salles spécialisées (datacenters). Les datacenters sont très sécurisés (accès restreint, protections anti-incendie, triple redondance électrique...) . L'accès physique à la machine est généralement impossible. On loue alors un serveur à un hébergeur (OVH, online). Toute la gestion doit donc se faire à distance... il faut savoir utiliser les outils logiciels appropriés.

connexion shell

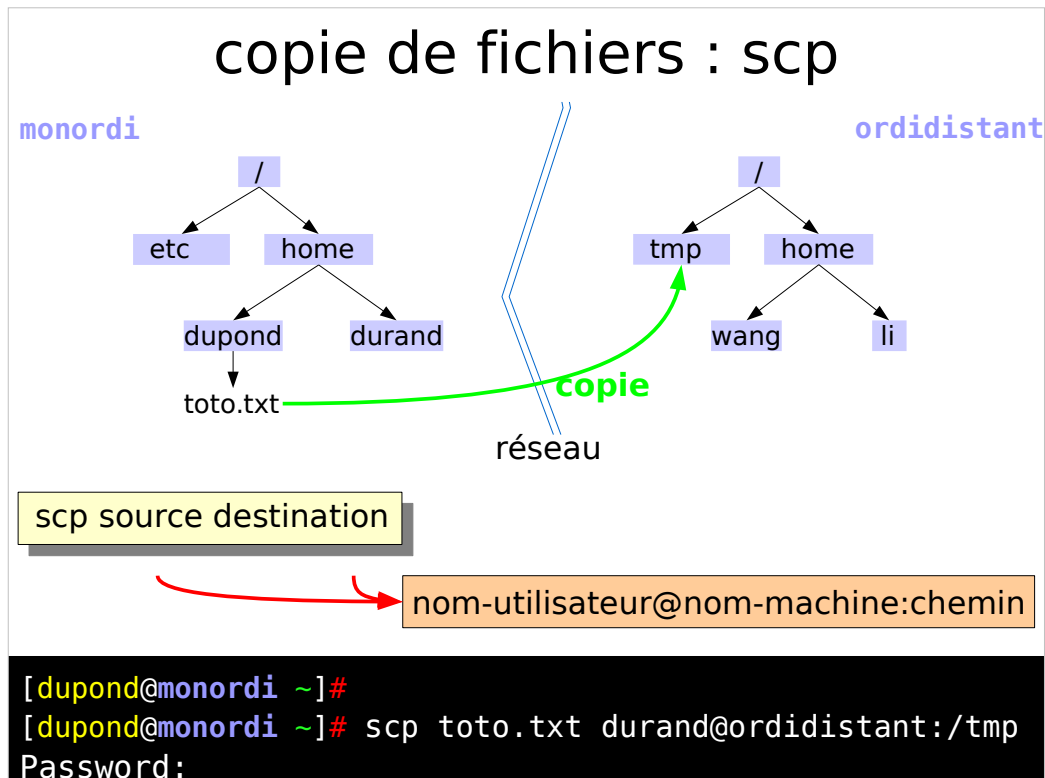
```
[dupond@monordi ~]#  
[dupond@monordi ~]# ssh ordidistant  
Password:  
[dupond@ordidistant ~]# sudo nano /etc/apache2/apache2.conf  
[dupond@ordidistant ~]# sudo systemctl restart apache2
```

- connexion à un terminal distant
- **ssh** : protocole crypté
- transfert de fichiers: **scp, rsync, ftp** (!)
- utilisation de fichiers distants: **sshfs**
- éditeurs mode texte : **nano, vi, emacs, ...**

Wikipedia : ssh

La gestion des serveurs Linux se fait généralement par ligne de commande. On peut se connecter à un serveur distant avec ssh et exécuter des commandes. Pour transférer des fichiers, préférez les protocoles cryptés (scp, rsync sftp) aux protocoles non-cryptés (ftp).

L'administration système consiste souvent à modifier des fichiers (par ex. de configuration). Des éditeurs "mode-texte" permettent d'éditer des fichiers en ligne de commande.



Pour copier des fichiers entre machines on peut utiliser scp. D'autres commandes comme rsync ont des syntaxes similaires.

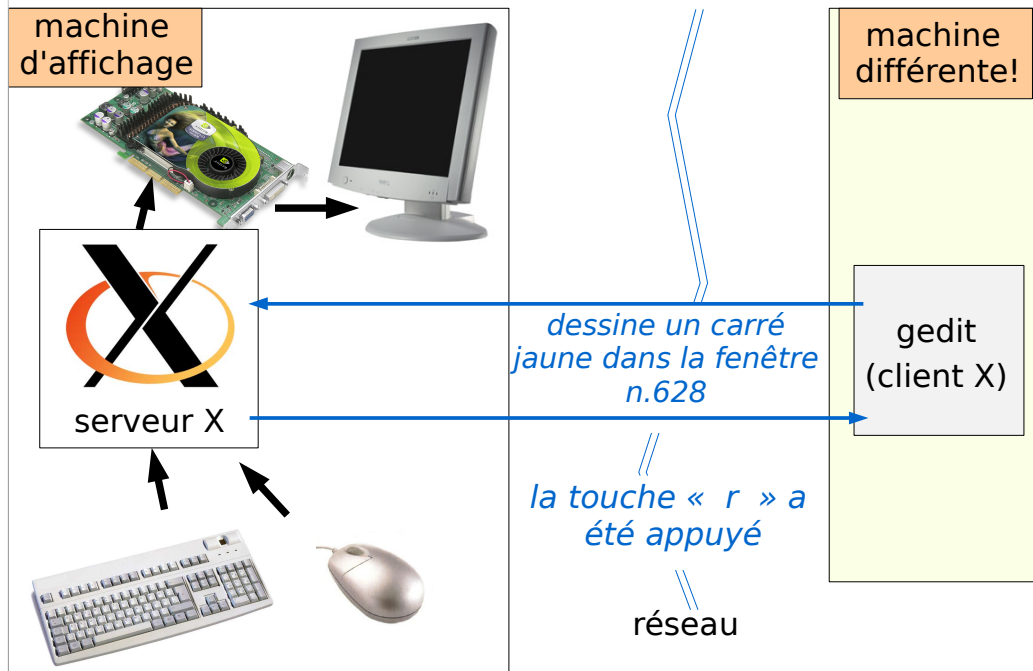
La syntaxe de la source et de la destination est :
nom-utilisateur@nom-machine:chemin

Si on omet "nom-utilisateur@" la commande utilisera l'utilisateur actuel.

Si on omet "nom-machine:" la commande utilisera la machine actuelle.

Si on omet "chemin", la commande utilisera le répertoire personnel (machine distante) ou le répertoire courant (machine actuelle)

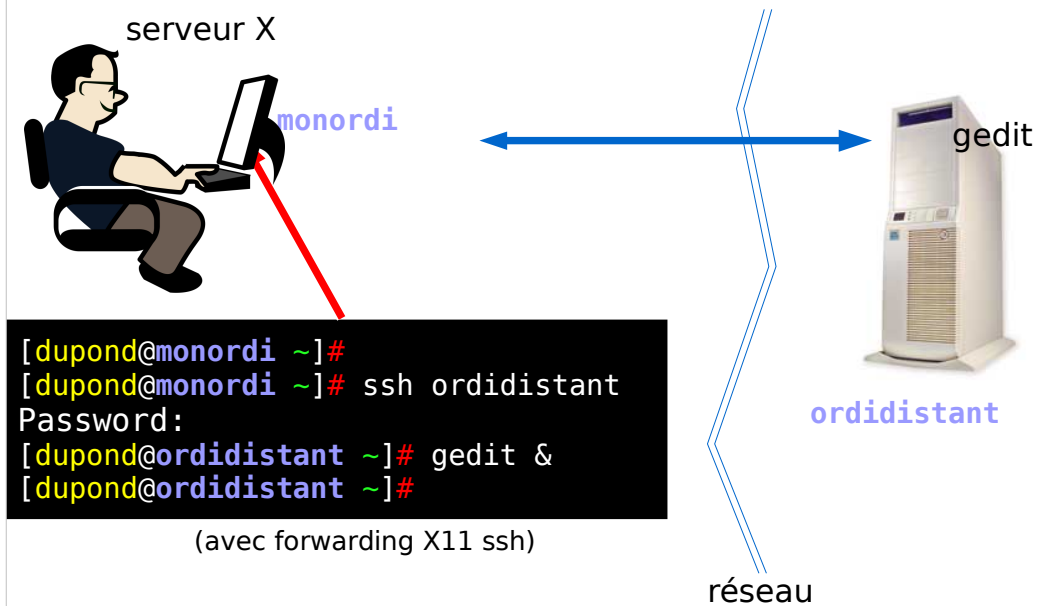
affichage graphique distant : X11



Il est possible aussi d'afficher des fenêtres à distance. Le système d'affichage X11 utilisé sur Linux permet d'afficher des fenêtres à distance.

Ce système fonctionne bien si les machines sont proches (ping bas).

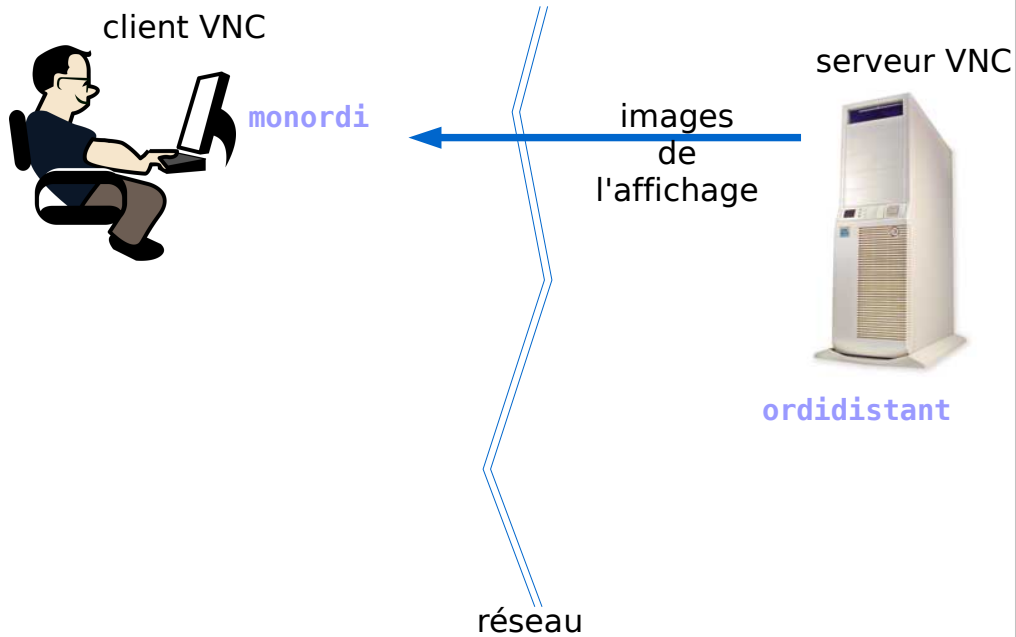
affichage graphique distant : X11



Wikipedia : X Window

On se connecte à une machine distante, on tape une commande qui s'exécute sur la machine distante, mais l'affichage de la fenêtre se fait sur la machine locale.

affichage graphique distant : VNC



Wikipedia : VNC

D'autres approches, comme VNC, permettent de déporter tout l'affichage de la machine à distance. Une "capture d'écran" est envoyé par le réseau.

gestion des logiciels

- librairies
- éléments et installation d'un logiciel
- paquets et leur gestion

librairies : rappel

rectangle.c

```
void dessiner_rectangle(...)  
{  
    ...  
}  
void dessiner_carre(...)  
{  
    ...  
}
```

librairie: collection de fonctions compilées

compilation

libdessin.a

cercle.c

```
void dessiner_cercle(...)  
{  
    ...  
}
```

compilation

Wikipedia : Bibliothèque logicielle

Les libraires sont des fichiers contenant une collection de fonctions compilées.

Supposons, qu'on écrive des fonctions pour dessiner des carrés, des rectangles et des cercles. On veut pouvoir utiliser ces fonctions dans de nombreux programmes.

On peut les compiler et les ranger dans un fichier libdessin.a

librairies statiques

mon-programme.c

```
int main(...)  
{  
    dessiner_cercle(...)  
}
```

libdessin.a

compilation

édition de liens

contient
"dessiner_cercle"

mon-programme

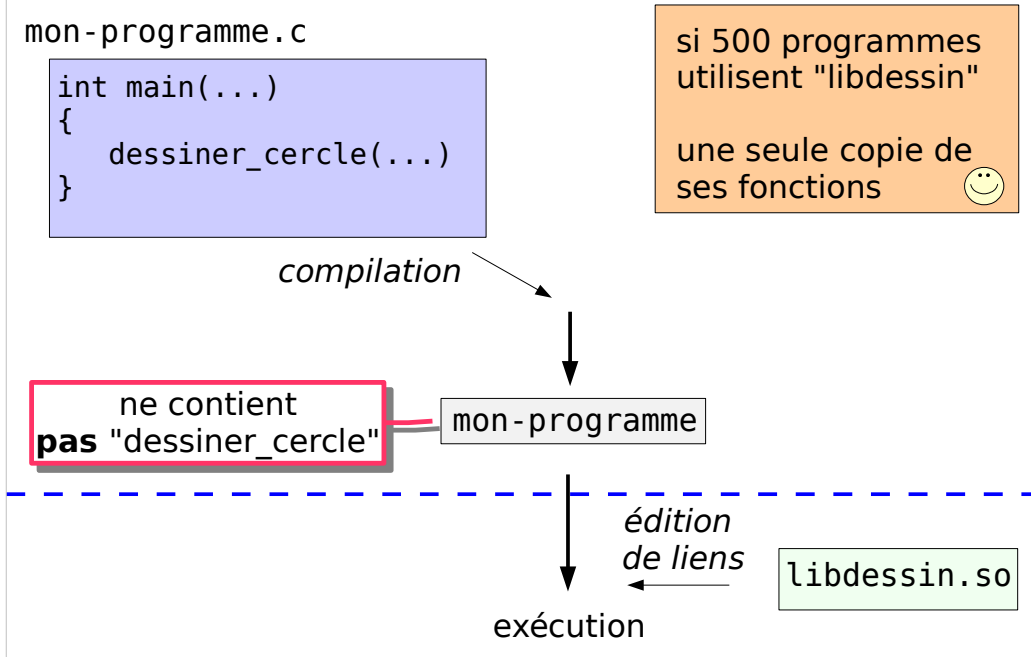
exécution

Wikipedia : Édition de liens

Une librairie statique est composée de fonctions compilées qui sont intégrées au programme qui les utilise, lors de l'édition de liens.

Dans ce cas, l'exécutable contient réellement les fonctions de libdessin.a. L'exécutable est autonome, il peut être exécuté sans libdessin.a.

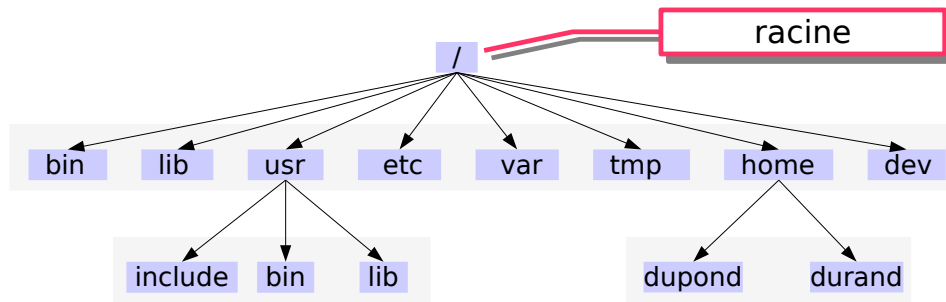
librairies dynamiques



Les librairies statiques posent des problèmes. Supposons, par exemple, que `libdessin.a` soit utilisée par 500 programmes. Ça veut dire qu'il existe 500 copies des fonctions `dessiner_cercle()` ! Ça prend beaucoup de place... Par ailleurs, si on corrige un bug dans `dessiner_cercle`, il est nécessaire de recompiler les 500 programmes... c'est ingérable.

On utilise donc des librairies "dynamiques" `.so` (`.DLL` sous windows). L'édition de liens ne se fait qu'à l'exécution. Le fichier exécutable n'est pas autonome, il **dépend** des libraires dynamiques (ici `libdessin.so`) pour pouvoir être exécuté.

répertoires UNIX importants



/bin et /usr/bin : programmes exécutables
/lib et /usr/lib : librairies
/etc : configuration du système
/usr/include : entête (header .h) programme C/C++
/tmp : fichiers temporaires
/home : répertoires personnels
/dev : périphériques
/var : données qui peuvent grossir
(logs, sites web, base de données...)

Wikipedia : FHS

On peut imaginer que /usr ne soit présent au démarrage. Donc tout ce qui est dans /usr ne doit pas être indispensable.
C'est ce qui explique la différence entre /bin et /usr/bin

éléments d'un logiciel

- programmes exécutables
- bibliothèques
- fichiers de données
- documentation
- ...

exemple: fichiers de « gpdf »

/usr/bin/gpdf

(aucune)

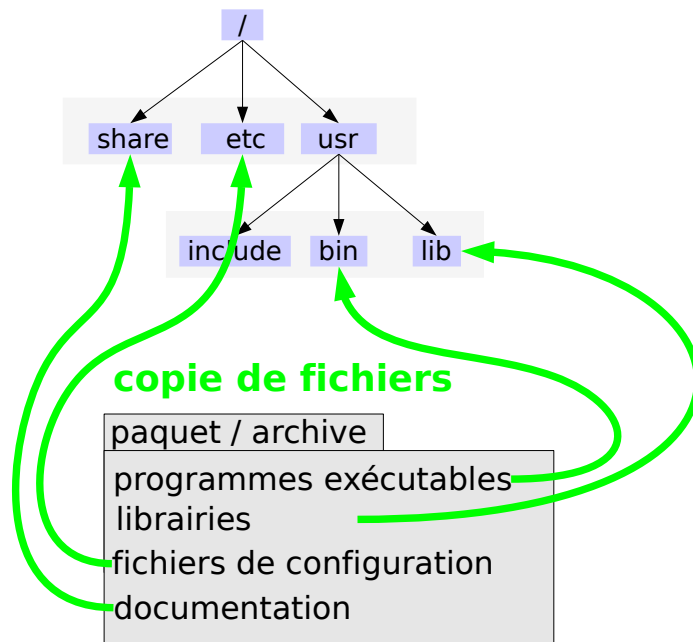
/usr/share/pixmaps/gnome-pdf.png

/usr/share/man/man1/gpdf.1.gz

Wikipedia : Logiciel (Contenu)

Un logiciel est constitué de nombreux fichiers.
On y retrouve des fichiers exécutables, des bibliothèques, des données, des images, de la documentation,...

installer un logiciel



L'installation d'un logiciel se fait généralement avec une commande qui gère tout automatiquement.

L'opération d'installation est, pour l'essentiel, une copie de fichiers à partir d'une archive vers les répertoires appropriés du système.

installer un logiciel

- à partir de "**code source** "
  compilation

- à partir de **paquets** / archives binaires

Il existe plusieurs manières d'installer un logiciel.

On peut, par exemple, l'installer à partir du code source. Dans ce cas, il faut configurer la compilation, compiler, et installer. C'est compliqué, et pose des problèmes importants de maintenance.

En général on installe des logiciels à l'aide du système de gestion de paquets de l'OS.

paquets : présentation

paquet: archive contenant des fichiers et des instructions pour les installer

comme une archive .tar
+ quelques informations

paquet binaires

formats



.deb

debian
ubuntu



fedora
mandriva

Wikipedia : Paquet logiciel, Gestionnaire de paquets

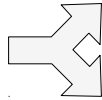
Les paquets sont des fichiers archives contenant tous les fichiers d'un logiciel. Nous parlerons ici uniquement des paquets binaires, contenant des exécutables déjà compilés. (Il existe aussi des paquets contenant du code source).

Il est rare de manipuler des paquets directement, on utilise en général un gestionnaire de paquets (par exemple, apt sous Debian).

paquets : administration

- installation
- gestion
- désinstallation
- mise à jour

informations
conservées



fichiers installés



quel paquet à servi à
installer ce fichier?

quels fichiers supprimer
pour désinstaller?

paquets installés



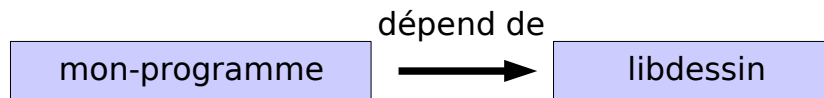
puis-je supprimer ce paquet,
sans casser de dépendances?

Le gestionnaire de paquets permet d'installer et de désinstaller des paquets. Il gère aussi les informations sur les paquets.

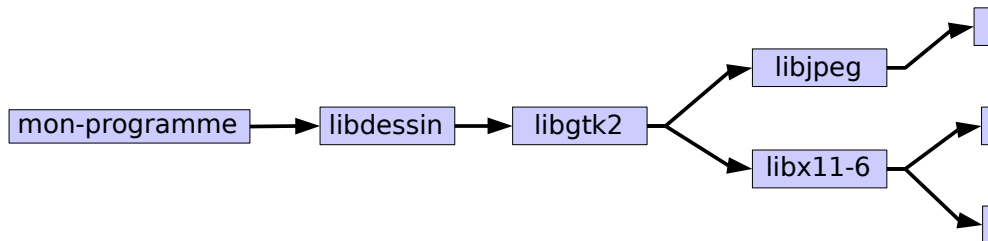
Si on trouve un fichier quelque-part et on ne sait pas à quoi il sert, on peut demander au gestionnaire de paquets: « Quel paquet a servi à installer ce fichier ? ».

Une fois que l'on connaît le nom du paquet, on peut demander, par exemple, une description du paquet.

dépendances entre paquets



pour installer **mon-programme**, il faut
d'abord installer **libdessin**



dépendances en chaîne!

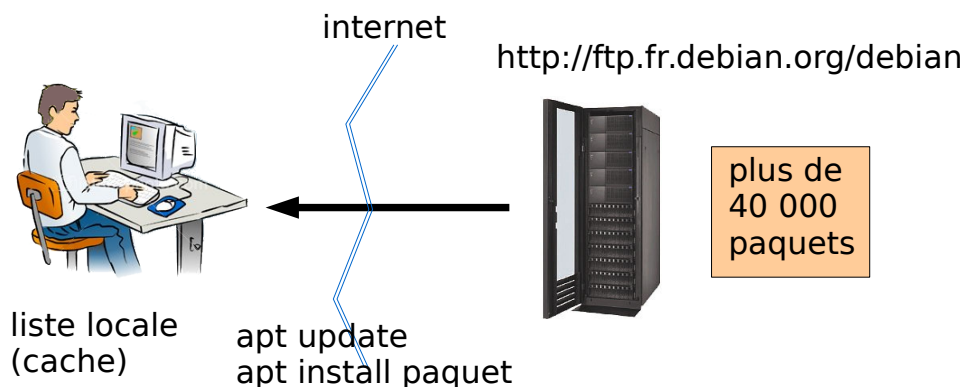
Certains paquets ont besoin d'autres paquets pour fonctionner. C'est qu'on appelle une « dépendance ». C'est le cas pour « mon-programme » qui a besoin de « libdessin ».

Il est fréquent d'avoir de nombreuses dépendances à la chaîne.

Quand on installe un logiciel, le gestionnaire de paquets va trouver toutes les dépendances et va toutes les installer.



ou trouver des paquets?



/etc/apt/sources.list

```
...  
deb http://ftp.fr.debian.org/debian/ jessie main  
...
```

Wikipedia : Advanced Packaging Tool

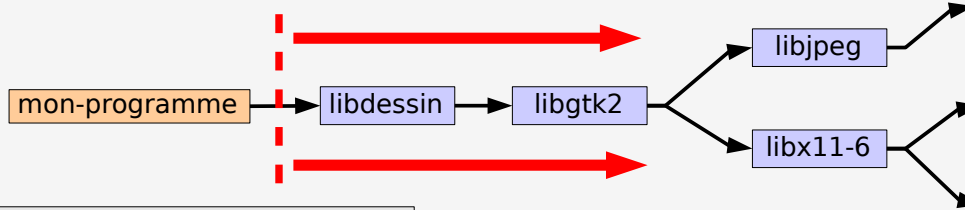
Debian est une distribution libre, presque tous les logiciels sont disponibles, gratuitement, sur un serveur. Ils sont trop nombreux et trop volumineux pour être stockés sur votre machine. Votre machine connaît la liste (cache) de tous les paquets disponibles.

Vous pouvez mettre à jour votre liste à partir du serveur avec « apt update ». Ceci ne met à jour que la liste, pas les logiciels eux mêmes.

L'adresse du serveur où se trouvent les paquets est configuré dans /etc/apt/sources.list

gestion de paquets

gestion des dépendances:



apt install mon-programme

mise à jour

versions: firefox-58.0

apt upgrade

interfaces graphiques
Debian: synaptic

debian: apt
mandriva: urpmi
fedora: yum

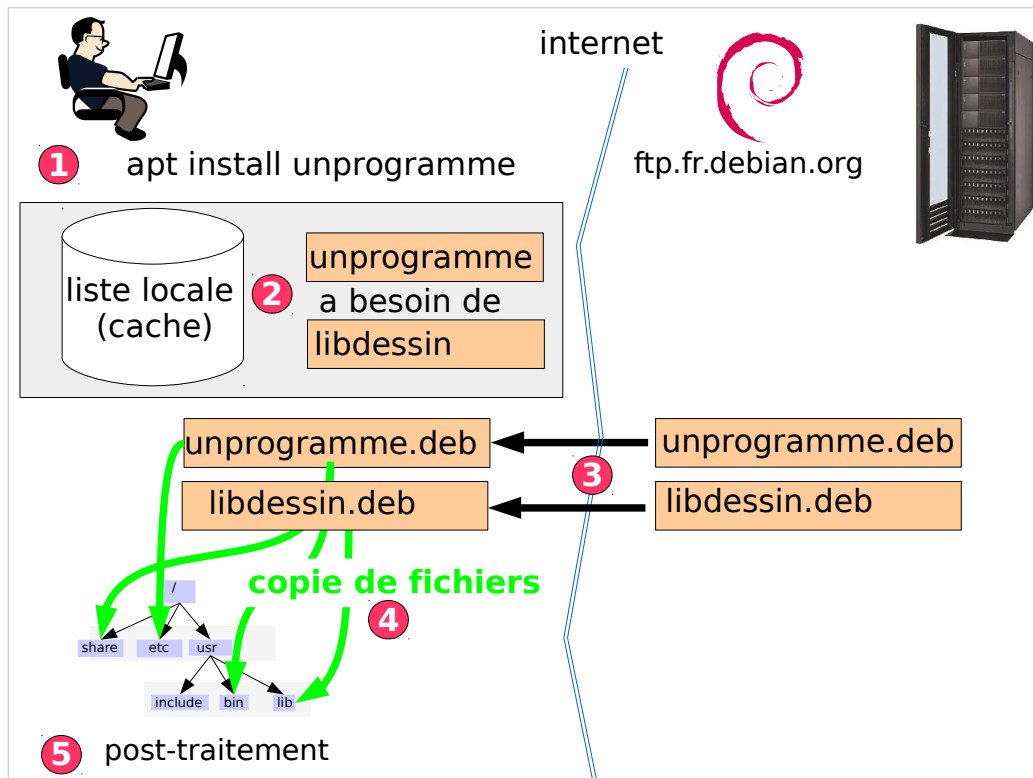
Pour installer un paquet:

apt install nom-paquet

Ceci va chercher aussi toutes les dépendances.

De nouvelles versions de logiciels apparaissent en permanence (corrections de bugs, failles de sécurité). Pour mettre à jour votre système:

apt upgrade



Récapitulons l'installation d'un paquet:

- 1) vous tapez : `apt install unprogramme`
- 2) Le gestionnaire de paquets regarde dans la liste locale de tous les paquets pour voir quelles sont les dépendances.
- 3) Le gestionnaire de paquets télécharge tous les paquets `.deb` à partir du serveur.
- 4) Le gestionnaire de paquets désarchive les paquets et copie les fichiers vers les répertoires du système.
- 5) Parfois, un post-traitement est fait (ex: lancer ou redémarrer un service)



commandes gestion paquets

- **apt update**
Mettre à jour votre *liste* de paquets
- **apt upgrade**
Mettre à jour tous les paquets déjà installés
- **apt search *mot-recherché***
Trouver le nom d'un paquet
- **apt show *nom-paquet***
Afficher des informations sur un paquet
- **apt install *nom-paquet***
Télécharger et installer le paquet et ses dépendances
- **apt list --installed**
Afficher une liste de tous les paquets déjà installés
- **dpkg -L *nom-paquet*** (L majuscule)
Afficher une liste des fichiers installés par ce paquet
- **dpkg -S *nom-fichier*** (S majuscule)
Trouver les paquets ayant servi à installer nom-fichier

les services

- présentation
- démons
- réseau
- fichiers de configuration
- fichiers logs

Wikipedia : Serveur informatique

exemples de services

serveur web ports 80 et 443

- apache
- nginx
- IIS



transfert de courrier
(SMTP) port 25

- exim
- postfix

serveur ftp port 21

- vsftp
- ncftpd
- ...

serveur ssh port 22

- OpenSSH

éléments d'un service

- démon
processus



- connexion réseau
port TCP



- fichiers de configuration



- fichiers de données



- fichiers log



démons

exemple : apache2 (serveur web)

démarrage / arrêt :

```
systemctl start    apache2  
systemctl stop     apache2  
systemctl restart  apache2
```

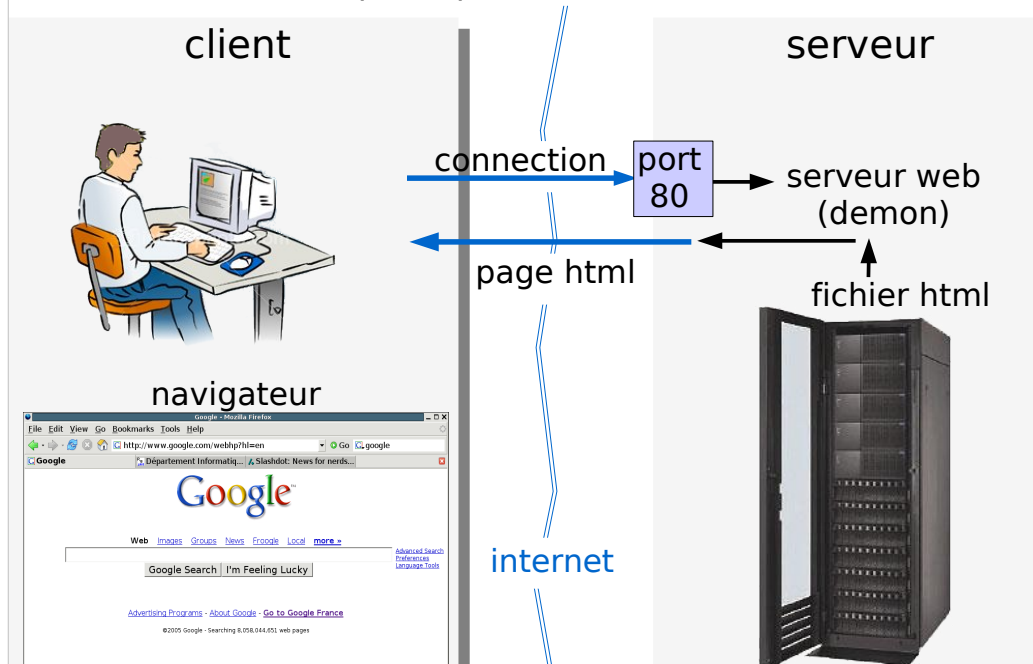
n'oubliez pas!!! — redémarrage après changement fichier config

démarrage du système : systemd

"apache2" est appelé "httpd" sur Red Hat

connection réseau

exemple : apache (serveur web)



fichiers de configuration

exemple : apache (serveur web)

répertoire: /etc

syntaxe généralement:

```
"nom valeur"  
"nom = valeur"  
"nom : valeur"
```

exemple: /etc/apache2/sites-available/000-default

```
...  
<VirtualHost 64.233.187.99>  
    ServerAdmin webmaster@toto.com  
    DocumentRoot /var/www  
    ServerName www.toto.com  
    ErrorLog /var/log/apache/public-error.log  
    CustomLog /var/log/apache/public-access.log ...  
</VirtualHost>  
...
```

Wikipedia : Fichier de configuration

fichiers log

fichiers où des serveurs écrivent
des messages traçant leur activité
ou indiquant des problèmes

réflexe : ça ne marche pas -> regarder les logs

répertoire: /var/log

/var/log/syslog
/var/log/auth.log
/var/log/apache2/access.log
/var/log/apache2/error.log

logrotate:

récent	• syslog
	• syslog.0
	• syslog.1.gz
	• syslog.2.gz
ancien ↓	• ...

journalctl

fichiers log

exemple : /var/log/syslog (log général)

```
Dec 28 03:45:59 monserveur sshd(pam_unix)[30148]: session opened for
user nicolas by (uid=0)

Dec 28 03:46:01 monserveur sshd(pam_unix)[30148]: session closed for
user nicolas

Dec 28 04:02:03 monserveur syslogd 1.4.1: restart.

Dec 28 12:02:50 monserveur named[32103]: lame server resolving
'monserveur.ovh.net' (in 'ovh.net?'): 213.186.50.98#53

Dec 28 12:41:20 monserveur xyz: maildircache: Cache create failure -
cannot change to bin

Dec 28 14:53:01 monserveur named[32103]: lame server resolving
'2.112.110.204.in-addr.arpa' (in '112.110.204.in-addr.arpa?'):
206.228.179.10#53
```

Doc: <http://minu.me/ab5f>

fichiers log

exemple : /var/log/apache/access.log (serveur web)

```
41.136.23.43 - - [26/Jan/2015:18:19:45 +0100] "GET /feed.xml HTTP/1.1"
200 3781 "-" "Digg Feed Fetcher 1.0 (Mozilla/5.0 (Macintosh; Intel Mac
OS X 10_7_1) AppleWebKit/534.48.3 (KHTML, like Gecko) Version/5.1
Safari/534.48.3)"

121.90.14.14 - - [26/Jan/2015:18:20:00 +0100] "GET /img/illus.png
HTTP/1.1" 301 608 "http://toto.org/" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:35.0) Gecko/20100101 Firefox/35.0"

17.10.116.23 - - [26/Jan/2015:18:20:02 +0100] "GET /article/784
HTTP/1.1" 200 16997 "https://www.google.fr/" "Mozilla/5.0 (Windows NT
6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.91
Safari/537.36"

17.10.116.23 - - [26/Jan/2015:18:20:03 +0100] "GET /css/accueil.css
HTTP/1.1" 200 5403 "http://web.example.org/article/784" "Mozilla/5.0
(Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/40.0.2214.91 Safari/537.36"

17.10.116.23 - - [26/Jan/2015:18:20:04 +0100] "GET /img/favicon.ico
HTTP/1.1" 200 576 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.91
Safari/537.36"
```

Ce document est distribué librement.

Sous licence GNU FDL :

<http://www.gnu.org/copyleft/fdl.html>

Les originaux sont disponibles au format LibreOffice

<http://www-info.iutv.univ-paris13.fr/~bosc>

Marcel.Bosc@iutv.univ-paris13.fr