



# Le Système Unix - Linux réseau

***Habib SMEI***

***Mail*** : *Habib.smei@isetsf.rnu.tn,*  
*habib@habibsmei.com*

***Web*** : *http://www.habibsmei.com*

# Plan

- Introduction
- Notions élémentaires de réseau
- Commandes
- Fichiers de configuration
- Services :
- NFS, SSH, SAMBA, APACHE, FTP, ...

**Réseau et généralité (en français):**

<http://www.urec.cnrs.fr/cours/>

**RFC (Request For Comments)**

**Document qui généralement définit un standard de communication ou une application sur l'Internet.**

**IETF : Internet Engineering Task Force**

<http://www.ietf.org/rfc.html>

# Introduction

**Un réseau est un ensemble de dispositifs matériels et logiciels permettant à 2 machines ou plus de communiquer. Pour mettre en évidence les concepts importants, on peut utiliser l'analogie avec la vie quotidienne:**

**Imaginez que vous (M. Bechir) ayez un client (M. ALi) au téléphone. Celui-ci vous soumet un problème délicat : vous attirez l'attention de votre collègue (M. Salah) et lui griffonnez quelques mots sur un bout de papier. M. Salah réfléchit un instant et vous griffonne sa réponse.**

**MM. Bechir, ALi et Salah représentent les machines connectées. Ces personnes sont reliées par 2 réseaux ("Liaison téléphonique" et "Liaison visuelle") auxquels ils sont reliés par 2 types d'interfaces ("Combiné téléphonique" et "Papier + Crayon"). M. Bechir, qui possède les 2 types d'interfaces, permet d'établir une communication entre M. ALi et M. Salah : il sert de passerelle entre les 2 réseaux. Enfin, chaque personne parle pendant un temps puis attend une réponse de son interlocuteur : ils communiquent en échangeant des paquets d'informations.**

# Notions élémentaires de réseau

## Réseau

Support permettant les échanges. Ce terme générique peut aussi bien désigner un câble coaxial reliant 2 machines qu'une ligne spécialisée (dont l'autre extrémité est reliée à une installation complexe, mais qui est vue comme un support ordinaire du point de vue d'une interface).

Le terme de réseau désigne aussi la totalité de l'installation (machines, interfaces et câblage).

## Interface

Dispositif assurant la connexion de la machine à un réseau. Les interfaces utilisées avec les réseaux informatiques sont les cartes réseaux, les modems et les ports parallèles.

## Routeur

Machine assurant l'interconnexion de plusieurs réseaux. On parle aussi de passerelle (en anglais: "router" et "gateway"). Il peut s'agir d'une machine du réseau dans laquelle on a installé plusieurs interfaces, ou d'un appareil spécialisé n'assurant que cette fonction.

# Notions élémentaires de réseau

## Paquet

Unité de transport d'information.

## Protocole

Ensemble de règles régissant les échanges d'informations.

## Adresse

Identification des éléments intervenant dans la communication (interfaces et réseaux). L'analogie postale est particulièrement bonne : le nom de la rue correspond à l'adresse de réseau et le numéro de la maison à l'adresse de l'interface.

# Notions élémentaires de réseau

## Adresses sur Internet :

De la même manière qu'une adresse postale :

comporte un pays (par défaut le pays ou la lettre a été postée), une ville, une rue, un numéro.

Pour repérer un machine sur internet, il faut un nom de machine et un nom de domaine.

Exemples:

mail.cck.rnu.tn



domaine=cck.rnu.tn

loria.loria.fr



domaine=loria.fr

ftp.inria.fr



domaine=inria.fr

www.pagesjaunes.tn



domaine=pagesjaunes.tn


# Notions élémentaires de réseau

## Numeros IP (Internet Protocol) :

Une adresse IPv4 est de la forme:

**XXX.XXX.XXX.XXX** ou **XXX** est un nb entre 0 et 255.

exemples :

mail.cck.rnu.tn  **193.95.30.165**

loria.loria.fr  **152.81.144.13, 152.81.144.12**

ftp.inria.fr  **192.93.2.6**

www.pagesjaunes.tn  **193.95.30.130**

Remarques :

- \* A une adresse peut correspondre plusieurs numéros de machines
- \* Un domaine peut être constitué par des plages d'adresses non contiguës d'où retrouver le nom d'une machine à partir de son numéro n'est pas un problème trivial mais il y a des outils. (<http://www.nic.fr/cgi-bin/whois>)

# Notions élémentaires de réseau

## Numeros IP (Internet Protocol) :

Une adresse IPv4 est de la forme:

**XXX.XXX.XXX.XXX** ou **XXX** est un nb entre 0 et 255.

exemples:

**cartan.iecn.u-nancy.fr -> 193.50.42.9**

**loria.loria.fr -> 152.81.144.13, 152.81.144.12**

**ftp.inria.fr -> 192.93.2.6**

**www.pagesjaunes.fr -> 193.252.242.65, 193.252.242.1**

Remarque :

**\* A une adresse peut correspondre plusieurs numéros de machines il y a des outils. (<http://www.nic.fr/cgi-bin/whois>)**

## Structure des adresses IP

Les **adresses IP** sont des **nombre de 32 bits** qui contiennent 2 champs :

- Un **identificateur de réseau** (NET-ID): tous les systèmes du même réseau physique doivent posséder le même identificateur de réseau, lequel doit être unique sur l'ensemble des réseaux gérés.
- Un **identificateur d'hôte** (HOST-ID): un noeud sur un réseau TCP/IP est appelé hôte, *il identifie une station de travail, un serveur, un routeur ou tout autre périphérique TCP/IP au sein du réseau.*

La concaténation de ces deux champs constitue une **adresse IP unique** sur le réseau.

Pour éviter d'avoir à manipuler des nombres binaires trop longs, les adresses 32 bits sont divisées en 4 octets. Ce format est appelé la **notation décimale pointée**, cette notation consiste à découper une adresse en quatre blocs de huit bits. Chaque bloc est ensuite converti en un nombre décimal.

Chacun des octets peut être représenté par un nombre de 0 à 255.

Ex : 130.150.0.1

# Notions élémentaires de réseau

## Classes d'adresses

La communauté Internet a défini **trois classes d'adresses** appropriées à des réseaux de différentes tailles. Il y a, a priori, peu de réseaux de grande taille (**classe A**), il y a plus de réseaux de taille moyenne (**classe B**) et beaucoup de réseaux de petite taille (**classe C**). La taille du réseau est exprimée en nombre d'hôtes potentiellement connectés.

**Le premier octet d'une adresse IP permet de déterminer la classe de cette adresse.**

Les adresses disponibles (de 0.0.0.0 à 255.255.255.255) ont donc été découpées en plages réservées à plusieurs catégories de réseaux.

Pour éviter d'avoir recours aux organismes NIC à chaque connexion d'un nouveau poste, chaque société se voit attribuer une plage d'adresse pour son réseau. Le nombre d'adresses disponibles dans chaque plage dépend de la taille du réseau de la société. Les grands réseaux sont dits de classe A (IBM, Xerox, DEC, Hewlett-Packard), les réseaux de taille moyenne sont de classe B (Microsoft en fait partie !), et les autres sont de classe C.

# Notions élémentaires de réseau

## Classes d'adresses

Classe	Début en binaire	Valeurs	Identificateur de réseau	Identificateur d'hôte
A	0...	1 à 126	a	b,c,d
B	10...	128 à 191	a,b	c,d
C	110...	192 à 223	a,b,c	d
D	1110...	224 à 239	multicast	a,b,c,d
E	1111...	240 à 255	réservées	expérimental

## Notions élémentaires de réseau

Par exemple, l'adresse d'un poste appartenant à un réseau de classe A est donc de la forme :

**0AAAAAAA**.xxxxxxx.xxxxxxxx.xxxxxxxx, avec A fixé par le NIC et x quelconque.

Exemple

IBM a obtenu l'adresse 9 (en fait, on devrait dire 9.X.X.X, mais il est plus rapide de n'utiliser que la valeur du premier octet). 9 est bien de classe A car  $9d=00001001b$

Cela signifie que chaque adresse IP du type  $00001001.xxxxxxxx.xxxxxxxx.xxxxxxxx$ , avec x prenant la valeur 0 ou 1, fait partie du réseau d'IBM.

# Notions élémentaires de réseau

## Identification du réseau

L'adresse IP se décompose, comme vu précédemment, en un numéro de réseau et un numéro de noeud au sein du réseau.

Afin de s'adapter aux différents besoins des utilisateurs, la taille de ces 2 champs peut varier.

On définit ainsi les 5 classes d'adresses notées A à E:

## Classes d'adresses

Classe A	0	7 bits N° de réseau	24 bits N° d'hôte
Classe B	10	14 bits N° de réseau	16 bits N° d'hôte
Classe C	110	21 bits N° de réseau	8 bits N° d'hôte
Classe D	1110	28 bits N° de groupe	
Classe E	1111	27 bits Usage futur	

Les systèmes appartenant au même réseau ont une partie d'adresse commune : la partie d'adresse du réseau

Adresses multi-destinataires (routeurs, multicast...)  
Adresses expérimentales

# Notions élémentaires de réseau

## Adresses réservées

Les adresses réservées ne peuvent désigner une machine TCP/IP sur un réseau.

**L'adresse d'acheminement par défaut** (route par défaut.) est de type **0.X.X.X**. Tous les paquets destinés à un réseau non connu, seront dirigés vers l'interface désignée par **0.0.0.0**.

**NB** : 0.0.0.0 est également l'adresse utilisée par une machine pour connaître son adresse IP durant une procédure d'initialisation (DHCP).

**L'adresse de bouclage** (*loopback*) : l'adresse de réseau 127 n'est pas attribuée à une société, elle est utilisée comme adresse de bouclage dans tous les réseaux. Cette adresse sert à tester le fonctionnement de votre carte réseau. Un ping 127.0.0.1 doit retourner un message correct. Le paquet envoyé avec cette adresse revient à l'émetteur.

Toutes les adresses de type 127.X.X.X ne peuvent pas être utilisées pour des hôtes. La valeur de 'x' est indifférente. On utilise généralement **127.0.0.1**

**L'adresse de réseau** est une adresse dont tous les bits d'hôte sont positionnés à 0 (ex 128.10.0.0 adresse de réseau du réseau 128.10 de classe B). Elle est utilisée pour désigner tous les postes du réseau. On utilise cette adresse dans les tables de routage.

**L'adresse de diffusion** est *une adresse dont tous les bits d'hôte sont positionnés à 1* (ex : 128.10.255.255 adresse de diffusion du réseau 128 de classe B).

Elle est utilisée pour envoyer un message à tous les postes du réseau.

# Notions élémentaires de réseau

## Les adresses "privées"

Les adresses suivantes (RFC 1918) peuvent également être librement utilisées pour **monter un réseau privé** :

A 10.0.0.0

B 172.16.0.0 à 172.31.255.255

C 192.168.0.0 à 192.168.255.255

Aucun paquet provenant de ces réseaux ou à destination de ces réseaux, ne sera routé sur l'Internet.

## Récapitulatif Classes d'adresses

**Tableau Récapitulatif**

	N°réseau	N°Hôte	
Classe A <i>126 réseaux</i>	<b>1 à 126</b>	0.0.1 à 255.255.254	$2^{24}-2=16\,777$ <i>16 777 adresses</i>
Classe B <i><math>2^{14}-2=16\,384</math> réseaux</i>	<b>128.1</b> à 191.254	0.1 à 255.254	$2^{16}-2=65\,534$ <i>65 534 adresses</i>
Classe C <i><math>2^{21}-2=2\,190\,000</math> réseaux</i>	<b>192.0.1</b> à 223.255.254	1 à 254	$2^8-2=254$ <i>254 adresses</i>

# Notions élémentaires de réseau

## masque de réseau (netmask)

Le rôle du masque de réseau (netmask) est d'identifier précisément les bits qui concernent le N° de réseau d'une adresse (il "masque" la partie hôte de l'adresse).

**Un bit à 1** dans le masque précise que le bit correspondant dans l'adresse IP fait partie du **N° de réseau** ; à l'inverse, **un bit à 0** spécifie un bit utilisé pour coder le **N° d'hôte**.

Ainsi, on a un masque dit "par défaut" qui correspond à la classe de ce réseau.

Exemple: dans un réseau de classe A sans sous-réseau, le premier octet correspond à l'adresse du réseau donc le **netmask** commence par 11111111 suivi de zéros soit **255.0.0.0**.

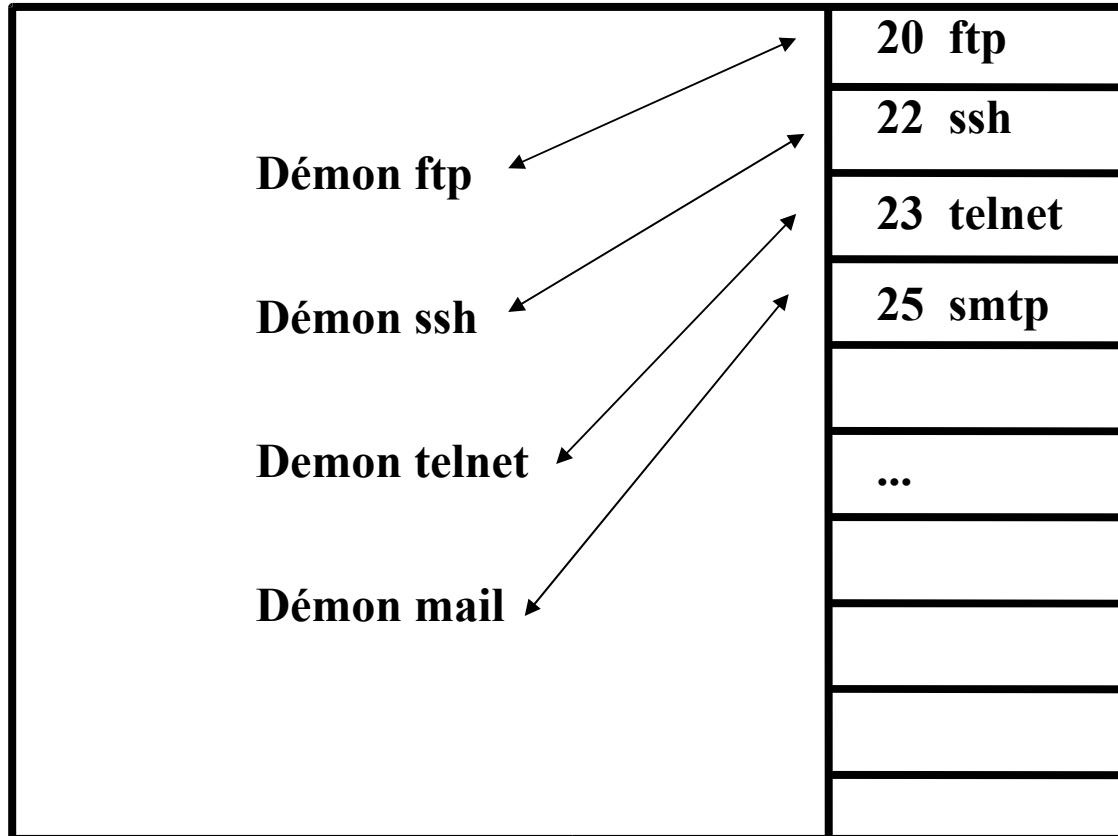
D'où le tableau suivant :

<i>Classe</i>	<i>Netmask</i>
<b>A</b>	<b>255.0.0.0</b>
<b>B</b>	<b>255.255.0.0</b>
<b>C</b>	<b>255.255.255.0</b>

# Notions élémentaires de réseau

## Notion de ports et de démons

On peut voir les ports comme des oreilles



## Notions élémentaires de réseau

### Le fichier: `/etc/services` :

```
# Network services, Internet style
ftp-data    20/tcp
ftp         21/tcp
ssh        22/tcp      # SSH Remote Login Protocol
ssh        22/udp      # SSH Remote Login Protocol
telnet     23/tcp
smtp       25/tcp      mail
tftp       69/udp
www        80/tcp      http      # WorldWideWeb HTTP
pop3       110/tcp     pop-3     # POP version 3
pop3       110/udp     pop-3
ntp        123/udp      # Network Time Protocol
imap2      143/tcp      imap      # Interim Mail Access Proto v2
imap2      143/udp      imap
snmp       161/udp      # Simple Net Mgmt Proto
z3950      210/tcp      wais      # NISO Z39.50 database
z3950      210/udp      wais
imap3      220/tcp      # Interactive Mail Access
imap3      220/udp      # Protocol v3
...
```

# Notions élémentaires de réseau

## Démon :

### Fonctionnement client-serveur:

- un serveur qui attend des requêtes d'éventuels clients
- un client qui émet des requêtes vers un serveur

**Un démon est un programme qui tourne en tâche de fond (background) qui attend des requêtes.**

**Un démon est un serveur dans un modèle client-serveur.**

**On communique avec un démon via le(s) numéro(s) de port(s) qu'il 'écoute'.**

### exemple:

**httpd qui est de démon serveur de pages WWW 'écoute' le port 80 et répond aux requêtes http qui lui sont soumises (s'il a le droit :)**

# Commandes

## les aides.

### **man**

man est la commande la plus utile du système car elle permet de connaître toutes les options possibles de chaque commande.

Ex: *man ls*

On notera que les man sont toujours mis jour sous les différents BSD ce qui pas toujours le cas sous linux.

### **info**

Semblable au man mais est aujourd'hui le format officiel de documentation sous Linux.

### **whatis**

Comme son nom l'indique permet d'avoir une description succincte sur un programme.

### **HOWTO**

le Howto est une particularité linux, qui explique comment installer ou configurer certaines programmes, fonctionnalités ou services.

On trouve généralement ces documentations au format HTML dans /  
*usr/share/doc/HOWTO*

# commandes

## Manipulation des fichiers et des répertoires :

ls cd mv pwd mkdir chmod chown du

Les outils de recherche :

find locate grep

## Commandes sur les processus :

top ps kill

## Commandes sur les périphériques :

- df dd mount/umount

## Commandes de connexion distance :

ssh (commande qui permet de se connecter de façon sécurisé ( les données qui transitent seront cryptées entre les deux machine. Ex: *ssh toto@192.168.0.1* )

scp (commande qui permet de copier de façon sécurisé un fichier ou un répertoire sur le compte d'une autre machine.

Ex: *scp fichier toto@192.168.0.1: scp -r repertoire toto@192.168.0.1)*

## Autres commandes système :

- *tar et gzip bzip2 bunzip2, ...*

# commandes

**ping** : permet de vérifier si une machine est présente sur le réseau. Un « ping 127.0.0.1 » permet de vérifier que la carte réseau fonctionne normalement

**arp** : permet de mettre en correspondance les adresses IP et les adresses MAC

**route** : permet de voir, d'ajouter ou de supprimer les routes déclarées sur la machine

**ifconfig** : permet de connaître la configuration réseau de la machine, mais aussi de la changer

**netstat** : permet de connaître les ports en écoute sur la machine, les connexions actives et d'autres choses

# commandes

## La commande arp

### *Description de la commande*

La commande **arp** permet de visualiser ou modifier la table du cache de l'interface. Cette table peut être statique et (ou) dynamique. Elle donne la correspondance entre une adresse *IP* et une adresse Ethernet.

À chaque nouvelle requête, le cache *ARP* de l'interface est mis à jour. Il y a un nouvel enregistrement. Cet enregistrement a une durée de vie (ttl ou *Time To Leave*).

Voici un exemple de cache *ARP* obtenu avec la commande **arp -va** :

```
? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0 >Entries: 1 Skipped: 0  
Found: 1 On voit l'adresse IP et l'adresse MAC correspondante. Il n'y a  
qu'une entrée dans la table. Voici les principales options de la commande  
arp :
```

**arp -s** (ajouter une entrée statique), exemple : **arp -s 192.168.1.2  
00:40:33:2D:B5:DD**

**arp -d** (supprimer une entrée), exemple : **arp -d 192.168.1.2**

## La commande **ifconfig**

La commande **ifconfig** permet la configuration locale ou à distance des interfaces réseau de tous types d'équipements (unité centrale, switch, routeur). La ligne de commande est :  
`ifconfig interface adresse [parametres]`.

Exemple : `ifconfig eth0 192.168.1.2` (affecte l'adresse 192.168.1.2 à la première interface physique).

Voici les principaux arguments utilisés :

*interface* logique ou physique, il est obligatoire,

`up` active l'interface

`down` désactive l'interface

# commandes

## La commande netstat

La commande **netstat**, permet de tester la configuration du réseau, visualiser l'état des connexions, établir des statistiques, notamment pour surveiller les serveurs.

Liste des paramètres utilisables avec **netstat** :

Sans argument, donne l'état des connexions,

-a afficher toutes les informations sur l'état des connexions,

-i affichage des statistiques,

-c rafraîchissement périodique de l'état du réseau,

-n affichage des informations en mode numérique sur l'état des connexions,

-r affichage des tables de routage,

-t informations sur les sockets *TCP*

-u informations sur les sockets *UDP*.

# commandes

**traceroute** : permet de déterminer la route prise par les paquets

**telnet** : permet de se connecter à distance sur une machine

**who** : permet de connaître les utilisateurs connectés sur la machine

**last** : permet de consulter l'historique des connexions

**finger** : permet d'obtenir des informations sur les utilisateurs d'une machine

**tcpdump** : permet la capture de paquets sur le réseau

**nmap** : permet de scanner les ports d'une machine

# commandes

## netstat -a :

TCP

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State
*.ftp	*.*	0	0	0	0	LISTEN
*.telnet	*.*	0	0	0	0	LISTEN
*.shell	*.*	0	0	0	0	LISTEN
*.login	*.*	0	0	0	0	LISTEN
*.exec	*.*	0	0	0	0	LISTEN
*.finger	*.*	0	0	0	0	LISTEN
*.32773	*.*	0	0	0	0	LISTEN
*.32774	*.*	0	0	0	0	LISTEN
*.nfsd	*.*	0	0	0	0	LISTEN
cartan.965	altair.nfsd	8760	0	8760	0	ESTABLISHED
cartan.964	loria.nfsd	8760	0	8760	0	TIME_WAIT
cartan.963	loria.nfsd	8760	0	8760	0	ESTABLISHED
cartan.1023	altair.1022	8760	0	8760	0	TIME_WAIT
cartan.34458	altair.6000	8760	0	8760	0	ESTABLISHED

# Fichiers de configuration

*/etc/rc.d/init.d/ : le répertoire /etc/rc.d/init.d/ contient tous les scripts de démarrage des différents services.*

*Pour démarrer un service on lancera typiquement la commande:*

*/etc/rc.d/init.d/nom\_service start*

*/etc/inetd.conf sous \*BSD ou /etc/xinetd sous Linux : ``inetd'' (respectivement xinetd) démarre les programmes fournissant des services Internet. Au lieu de démarrer ces services au moment de l'initialisation du système, et de les laisser inactifs jusqu'à ce qu'il y ait une demande de connexion, on ne démarre que inetd (xinetd) et celui ci écoute sur tous les ports nécessaires aux services listés dans ses fichiers de configuration. Lorsqu'une requête arrive, inetd (xinetd) démarre le service correspondant. à cause de la façon dont il fonctionne, inetd (xinetd) est aussi appelé le super-serveur.*

- /etc/passwd* : contient la liste des comptes sur le système
- /etc/shadow* : contient les mots de passe cryptés des utilisateurs
- /etc/group* : définit les groupes auxquels appartient chaque utilisateur.
- /etc/fstab* : contient des informations sur les différents systèmes de fichiers
- /etc/syslog.conf* : le service syslog sert à logger diverse informations dans des fichier
- /etc/hosts* : hosts - Correspondances statiques de noms d'hôtes. ex: 127.0.0.1 localhost
- /etc/networks* : liste d'adresses et de noms de réseaux
- /etc/resolv.conf* : **définit la machine sur laquelle devront être transmis les requêtes DNS**
- /etc/services* : services est un fichier de texte ASCII fournissant une correspondance entre un nom intelligible décrivant un service Internet et l'ensemble numéro de port / protocole utilisé.

# Fichiers de configuration

## Le fichier `/etc/hosts`

Le fichier hosts donne un moyen d'assurer la résolution de noms

Exemple de fichier host

```
127.0.0.1 localhost localhost.localdomain
```

```
192.168.1.1 machine1.domaine1.org machine1
```

## Le fichier `/etc/networks`

Il permet d'affecter un nom logique à un réseau

```
localnet 127.0.0.0
```

`foo-net 192.168.1.0` Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse.

**route** add *foo-net* au lieu de **route** add **192.168.1.0**.

## Le fichier `/etc/host.conf`

Il donne l'ordre dans lequel le processus de résolution de noms est effectué. Voici un exemple de ce que l'on peut trouver dans ce fichier :

```
order hosts,bind
```

La résolution est effectuée d'abord avec le fichier host, en cas d'échec avec le *DNS*.

## configuration

Configuration de la carte réseau :

Fichier de configuration :

*/etc/sysconfig/network-scripts/ifcfg-eth0*

**DEVICE**="eth0"

**IPADDR**="XXX.XXX.XXX.XXX" # adresse IP ou vide si DHCP

**NETMASK**="255.255.255.0"

**ONBOOT**="yes"

**BOOTPROTO**="static" # ou DHCP si on est en DHCP !

# Services

**nfs** : Le partage de fichiers pour les clients Unix

Ce service permet à des machines Unix se partager des répertoires et fichiers.

## Les fichiers de configuration du client NFS

Il n'y a pas de fichier particulier. Il suffit que les programmes soient installés (portmap et nfs-common). Pensez à lancer le **portmap**, sinon le montage restera en attente.

`/etc/init.d/portmap start`

Les répertoires exportés par un serveur peuvent être << montés >> manuellement ou à la demande.

### Exemple Unix de montage NFS

Prenons la configuration précédente ( fichier /etc/exports ci-dessus)

Le client cli1 monte (importe) /tmp de ns1 sur le répertoire local /tempo en utilisant la commande suivante

```
$mount -t nfs ns1:/tmp /tempo. -t indique le type de SGF - arborescence NFS -
```

Une fois montée, l'accès à la ressource est transparent.

En fin d'utilisation, le client démonte l'arborescence /tmp en utilisant la commande `$umount /tempo`

A chaque opération de montage ou démontage, le fichier local /etc/mstab est mis à jour. Il contient la liste des systèmes de fichiers montés (arborescence NFS ou non).

Attention : NFS utilise un cache. Si vous ne voulez pas perdre de données, utiliser une procédure de << démontage >> des disques ou alors un << shutdown >> du poste client. ~~Dans les autres cas, vous risquez de perdre les informations logées en cache.~~

# Services

**nfs** : Le partage de fichiers pour les clients Unix

Ce service permet à des machines Unix se partager des répertoires et fichiers.

Le fichier `/etc/exports` décrit ce que le serveur exporte, vers quelles machines le serveur exporte, avec quelles autorisations. Il s'agit d'un fichier texte, qui est éditable avec n'importe quel éditeur. Il centralise la liste de l'ensemble des ressources offertes par cette machine. Notez cependant que le nom de partage est automatiquement celui de la ressource (on ne peut pas partager sous un autre nom), et qu'enfin, on peut ponctuellement partager une ressource sans passer par ce fichier (à la volée : voir `exportfs`).

La structure d'une ligne est de la forme :

PointDeMontage client1(option) clientn(option)

- PointDeMontage est le volume local à exporter,
- Client1 ... Clientn définissent les ordinateurs qui ont le droit d'accéder au volume exporté,
- Option: définit le type d'accès et les permissions.

# Services

Exemple de fichier /etc/exports :

```
# Exporte le répertoire /tmp vers la machine "voisin" avec possibilité read write (rw)
```

```
# rw est l'option par défaut
```

```
/tmp      voisin(rw)
```

```
#Exporte "/tmp" en lecture seule vers toutes les machines du réseau
```

```
/tmp     *(ro)
```

```
/rep     *.archinet.edu(rw)
```

```
/usr/local/test  *.archinet.edu(ro)
```

Le dossier /rep est exporté en lecture et écriture pour tous les ordinateurs du domaine archinet.edu. Le dossier /usr/local/test en lecture uniquement.

# Services

## nfs :

### /etc/exports

Ce fichier comporte la liste des répertoires partagés sur la machine, pour qui ils le sont, et quels sont leurs permissions.

```
llw@PTC01tm) 1.8 File: /etc/exports
/home/serveur 19.18.2.0/255.255.255.0(rw)
/home/public *(ro)
```

Pour le premier répertoire, seuls les ordinateurs de la gamme d'adresses spécifiée auront accès en écriture. Pour le deuxième répertoire, tout le monde a accès en lecture seule.

Parmi les options particulières, on peut noter celles-ci:

- all\_squash tous les users qui se connectent auront les droits de nobody
- no\_root\_squash idem, sauf root
- no\_all\_squash tous les users gardent leurs droits sur ce répertoire.
- anonuid=501 attribuent à l'utilisateur l'identité d'un autre (=>sécurité!)
- anongid=501 (Anonymous User IDentifier & Anonymous Group IDentifier)

## Configuration et utilisation du client Unix/Linux

### Programmes client

Comme vu plus haut, n'oubliez pas d'avoir **portmap** actif sur votre client. Les programmes clients à utiliser sont : **mount** et **showmount**

### Le fichier `/etc/fstab`

Ce fichier contient une table des volumes montés sur le système. Il est utilisé par les daemons `mount`, `umount`, `fsck`. Les volumes déclarés sont montés au démarrage du système. Voici un extrait de fichier :

### *Exemple*

```
/dev/hda1 / ext2 defaults 1 1 /dev/hda2 swap swap defaults 0 0 /dev/fd0 /mnt/floppy ext2  
noauto 0 0 /dev/cdrom /mnt/cdrom iso9660 user,noauto,ro 0 0 ns1:/usr/local/man /doc  
nfs rsize=8192,wsize=8192,timeo=14,intr
```

La dernière ligne indique que le volume `/usr/local/man`, situé sur le serveur `ns1`, et qui contient les pages du manuel est un volume `nfs`, monté sous le nom de `local` de `/doc`.

Ce fichier évite d'avoir à << monter >> manuellement des systèmes de fichiers ou d'avoir à indiquer les points de montage, bien que cela puisse s'avérer parfois nécessaire (utilisation ponctuelle d'une ressource...). L'option `auto` permet de préciser si le montage doit être fait automatiquement au démarrage de la machine. L'option `noauto` permet d'indiquer le montage tel qu'il doit être fait, lors d'une demande manuelle de montage (pratique pour les disquettes, CD et autres lecteurs amovibles).

## Montage manuel de système de fichiers

La commande souvent utilisée est de la forme **mount -t TypeDeSGF *NomDeMontage VolumeMonté***.

Vous pourrez avoir toutes les options avec la commande **man *mount*** ou une aide plus brève avec **mount --help**.

Exemple de montage : **mount -t nfs ns1:/usr/local/man /doc** .

La forme standard de la commande **mount** est **mount -t type *périphérique répertoire*** avec :

Type : type de sgf (fat, vfat, nfs, ext2, minix....) pour nous c'est nfs

Périphérique : nom du fichier exporté sous la forme  
NomServeur:NomDossierExporté

Répertoire : nom du répertoire local de montage/

Le type de fichier que vous montez est de type nfs, vous utiliserez l'exemple de la commande ci-dessous :

**mount -t nfs *serveurNFS:/usr/share/doc /mnt/doc***

# Services

Commentaire : la ligne de commande monte le répertoire exporté /usr/share/doc du serveur serveurNFS, sur le répertoire local du client /mnt/doc.

Le mtab est modifié chaque fois que l'utilisateur << monte >> ou << démonte >> un système de fichiers. Le système tient à jour une table des volumes montés.

Attention, l'accès à la commande **mount** n'est, par défaut, autorisée que pour root.

Il faut rajouter l'option user dans le fichier /etc/fstab, afin qu'un autre utilisateur puisse accéder à cette commande.

Exemple : /dev/cdrom /mnt/cdrom iso9660 noauto,ro  
devient /dev/cdrom /mnt/cdrom iso9660 user,noauto,ro

La prise en compte des modifications est dynamique.

La commande **mount** sans paramètres donne la liste des volumes montés.  
La commande consulte la table maintenue à jour dans le fichier mtab.

# Services

## La commande showmount

Cette commande permet d'interroger un hôte distant sur les services *NFS* qu'il offre, et notamment les volumes qu'il exporte.

**showmount** -e AdresseIP\_ou\_NomIP lancée à partir d'un client nous affichera la liste des ressources offertes par sAdresseIP\_ou\_NomIP (=serveur).

Sur le serveur, **showmount** -a nous affichera la liste des clients connectés sur chacune de nos ressources.

De même, sur le serveur, la command **showmount** -e affiche le liste des partages en cours.

## Autres commandes d'administration

**rpcinfo** : (par exemple **rpcinfo** -p consulte le catalogue des applications *RPC* (nfsd, mountd sont des applicatifs *RPC* parmi d'autres).

**nfsstat** : fournit des statistiques d'utilisation de *NFS*.

La commande **exportfs** permet elle aussi d'obtenir la liste des partages en cours, de relancer le service (pour la prise en compte d'éventuelles modifications du fichier /etc/exports, voir même d'effectuer un partage à la volée (sans passer par /etc/exports).

**exportfs** -v affiche les partages en cours.

**exportfs** -r active les changements fait dans le fichier de configuration de partage NFS (il fait relire le fichier /etc/exports par le programme serveur).

**exportfs** machine:/repertoire offre à la volée à machine (qui peut être aussi bien un nom de machine qu'étoile, ou un réseau) le partage /repertoire. On peut passer des options avec -o.

# Services

## exportfs :

Cette commande tapée seule affiche la liste des shares ainsi que les utilisateurs autorisés. exportfs -a relit la configuration du fichier /etc/exports.

La liste des options suivantes est extraite de man exportfs.

### OPTIONS

- a Export or unexport all directories.
- a options,... Specify a list of export options in the same manner as in exports(5).
- i Ignore the /etc/exports file, so that only default options and options given on the command line are used.
- r Reexport all directories. It synchronizes /var/lib/nfs/xtab with /etc/exports. It removes entries in /var/lib/nfs/xtab which are deleted from /etc/exports, and removes any entries from the kernel export table which are no longer valid.
- u Unexport one or more directories.
- v Be verbose. When exporting or unexporting, show what's going on. When displaying the current export list, also display the list of export options.

# Services

Les services nfs et nfslock :

nfs est le service qui permet à votre machine Linux d'utiliser nfs, soit en version client (voir précédemment) ; soit en version serveur, où ce sont vos fichiers et répertoires qui sont partagés sur le réseau.

nfslock est aussi un service qui règle les accès concurrents ; il limite le nombre de connexions simultanées à un dossier partagé.

le fichier **/var/lib/nfs/rmtab** :

Ce fichier détient la liste des clients connectés à votre machine, ceux qui ont monté un répertoire partagé de votre machine dans leur arborescence.

Lorsque le client se déconnecte, en clair, lorsqu'il utilise la commande umount, la ligne correspondant à son adresse IP et à son point de montage disparaît. Attention, il se peut que cette ligne ne soit pas effacée si la machine du client se crashe ou reboot.

# Services

## nfsstat :

La commande nfsstat rassemble et affiche les statistiques d'utilisation de nfs.  
En fait, seules les infos du côté serveur sont disponibles.

```
Server nfs v3:
null      getattr      setattr      lookup      access      readlink
0         0% 0         0% 0         0% 0         0% 0         0% 0
read      write        create        mkdir        symlink      mknod
0         0% 0         0% 0         0% 0         0% 0         0% 0
remove    rmdir        rename        link         readdir      readdirplus
0         0% 0         0% 0         0% 0         0% 0         0% 0
fsstat    fsinfo       pathconf     commit
0         0% 0         0% 0         0% 0         0%
```

## Services - Application

Créer un répertoire partage qui contient un ensemble de fichiers.

Partager ce répertoire à un ensemble d'utilisateurs

# Installation d'un client/serveur SSH

## Installation du serveur SSH

**SSH** (**S**ecure **S**hell, port 22) est un aujourd'hui la façon la plus utilisée pour se connecter sur une autre machine. En effet lors de la connexion, le mot de passe ainsi que les données sont cryptées, ce qui évite aux sniffers de pouvoir capturer les mots de passe et les données transitant sur le réseau.

### installation

```
rpm -Uvh openssh-server-*.rpm
```

### démarrage

```
service sshd start
```

## Mode de fonctionnement de SSH

L'établissement du dialogue entre le client et le serveur suit un protocole particulier :

3. établissement d'une couche transport sécurisée
4. chiffrement des données à l'aide de clefs symétriques pendant la transaction

Le client peut s'authentifier en toute sécurité, et accéder aux applications conformes aux spécifications du protocole.

# Services : SSH

## Configuration

Le fichier de configuration se trouve dans `/etc/ssh/sshd_config` les options les plus intéressantes sont:

<code>#Port 22</code>	Spécifie le port que le serveur doit utiliser.
<code>Protocol 2,1</code>	Protocole utilisé
<code>#LoginGraceTime 600</code>	Temps de connexion maximum
<code>#PermitRootLogin yes</code>	permet ou interdit la connexion ``root``
<code>#RSAAuthentication yes</code>	Méthode d'authentification.
<code>#AuthorizedKeysFile .ssh/authorized_keys</code>	fichier utilisé pour ``l'autologin``
<code>#PermitEmptyPasswords no</code>	permet ou non les mots de passe vide
<code>X11Forwarding yes</code>	permet ou non d'exporter le DISPLAY

## Remarques

Il plus que préférable d'ajouter ces lignes à votre ``sshd\_config``:

```
ClientAliveInterval 15
```

```
ClientAliveCountMax 3
```

afin d'éviter qu'une connexion morte (ex: client débranché) ne fasse l'objet d'une récupération de session. **Connexion SSH sur une autre machine**

```
ssh toto@192.168.0.2
```

# Services : SAMBA

## SAMBA :

Ce protocole permet de faire communiquer des PC sous windows avec d'autres sous linux.

### Installation

```
# rpm -qa | grep samba  
samba-server-3.0.2a-3mdk  
samba-common-3.0.2a-3mdk  
samba-client-3.0.2a-3mdk
```

### test de l'installation

```
# /etc/rc.d/init.d/smb start
```

Lancement du service SaMBa : [ OK ]

Lancement du service NMB : [ OK ]

Deux démons sont lancés, nécessaires au fonctionnement du serveur Samba : smbd et nmbd.

- smbd permet le partage des fichiers et imprimantes
- nmbd permet quant à lui le parcours du réseau et la résolution de noms Netbios...

## Automatiser le lancement de Samba

```
# chkconfig --level 345 smb on
```

## Le fichier de configuration

```
/etc/samba/smb.conf
```

Le fichier de configuration est découpé en grandes sections indiquées de cette manière : [section]. Les différents paramètres sont ensuite inscrits de la manière suivante : paramètre = valeur. Toute ligne commencée par un "#" ou ";" est considérée comme un commentaire. Par convention, le ";" est utilisé pour commenter les lignes de configuration.

# Services : SAMBA

## Section de configuration générale

La première section est annoncée par [global]. Elle contient les éléments généraux de la configuration du serveur Samba : nom du groupe de travail, réseaux autorisés, utilisateurs administrateurs...

- **workgroup** : nom du groupe de travail ou du domaine
- **netbios name** : nom netbios du serveur Samba, par défaut égal au nom de la machine (hostname)
- **server string** : description affichée lors du parcours réseau, il s'agit d'un commentaire.
- **printing** : système d'impression utilisé pour le serveur d'impression : sous Linux on trouvera lprng et de plus en plus, cups (le choix réalisé dans cet article).
- **log file / max log size / log level** : configuration des logs du serveur : respectivement le nom du fichier de log, sa taille maximum et le niveau des logs (plus le niveau est élevé, plus la quantité d'informations est importante)
- **hosts allow (deny)** : entrer ici la liste des adresses IP des machines (ex : 192.168.0.3) ou réseaux (ex : 192.168.0.) autorisés à se connecter au serveur Samba (et inversement si on utilise le paramètre hosts deny). Le paramètre est important surtout si votre machine est accessible de l'extérieur. Le protocole Netbios fait l'objet de nombreuses attaques. Pensez également à configurer votre firewall pour bloquer les ports 137, 138 et 139 de l'extérieur.

# Services : SAMBA

- **security** : c'est une des options les plus importantes du fichier, qui pourra être bloquante si elle est mal renseignée. Elle indique le mode de discussion du client Windows avec le serveur Samba. Dans les versions 3.x de Samba, le défaut est user, et share pour Samba 2.x. Dans la version qui nous intéresse, on utilisera le mode user si les noms de comptes utilisés pour se connecter au serveur ont un compte équivalent sur la machine Linux (existence d'une entrée dans /etc/passwd). Dans le cas contraire, on préférera share. Il existe également 2 autres types possibles : domain et server que nous n'aborderons pas ici et qui sont réservés dans un fonctionnement de Samba en tant que contrôleur de domaine. Ci-dessous quelques explications sur les implications de ce choix :
- **share** : ce mode ne nécessite pas d'authentification par un compte valide. Si le paramètre guest only est renseigné, alors tout nouvel utilisateur sera identifié par le biais de cet utilisateur invité.
- **user** : dans ce cas de figure, l'utilisateur doit s'authentifier systématiquement. Son compte windows devra disposer d'un compte correspondant sur le serveur (on parle aussi en bon français de mapping)
- **encrypt passwords / unix password sync / passwd program / passwd chat** : configuration des mots de passe. On aura recours aux mots de passes encryptés. La synchronisation des mots de passe permet la synchronisation entre le mot de passe de l'utilisateur Samba et son compte sur le système Linux. En l'autorisant, on permet à l'utilisateur de modifier son mot de passe à partir de la machine cliente et donc d'un poste client sous Windows. Enfin passwd program et passwd chat indiquent le programme utilisé pour réaliser cette modification ainsi que le dialogue qui s'établira avec le serveur. Les paramètres par défaut conviennent parfaitement.

# Services : SAMBA

## Section de configuration des partages de fichiers

Nous allons maintenant passer aux sections de partages de fichiers. Nous aurons une section par partage défini. A l'intérieur de chacune de ces sections, nous trouverons les options qui définissent le dit partage.

Nous allons donner un nom à chaque partage. Attention il existe un nom de partage de fichiers spécifique : [homes]. Il définit le partage des répertoires personnels des utilisateurs, sans avoir à spécifier le chemin dans les options. De manière générale, on aura donc [monpartage].

Ci-dessous le détail des options les plus courantes utilisées :

- **path** : chemin d'accès du partage - il n'est pas à spécifier pour le partage [homes]
- **comment** : commentaire décrivant le partage qui apparaît lors du parcours du réseau Samba (voisinage réseau sous Windows), il s'agit uniquement d'un commentaire
- **browseable** : le partage sera visible lors du parcours du réseau
- **read only** : limite l'accès en lecture uniquement
- **write list** : limite l'accès en écriture aux données du partage aux utilisateurs et/ou groupes d'utilisateurs spécifiés. Un groupe sera mentionné de cette façon : @nom\_du\_groupe.

## Configuration des partages d'imprimantes

Votre serveur Samba peut également vous servir à partager des imprimantes. Nous traiterons ici du cas où ces imprimantes sont gérées par cups.

Le partage d'imprimante peut se faire à plusieurs niveaux :

- partage de la ressource (partage intitulé [printers]), mais qui nécessite d'avoir installé les drivers sur le poste client,
- partage des drivers, dans ce dernier cas vous n'avez plus besoin d'installer de driver sur le client Windows, le partage est intitulé [print\$].

Vous pouvez décider de partager uniquement la ressource ou bien les drivers et la ressource.

# Services : SAMBA

[global]

...

# système d'impression utilisé

printing = cups

# administrateur des imprimantes

printer admin = root

# partage des ressources d'impression

[printers]

comment = All Printers

path = /var/spool/samba

create mask = 0700

guest ok = Yes

printable = Yes

browseable = Yes

# partage des drivers des imprimantes

[print\$]

path = /var/lib/samba/printers

browseable = yes

read only = yes

write list = root

# Services : SAMBA

Pour vérifier que le lien a bien été réalisé, vous avez plusieurs possibilités.

la commande **smbclient** va vous lister, entre autres, les imprimantes partagées : # smbclient -L localhost

Password:

Anonymous login successful

Domain=[MONGROUPE] OS=[Unix]

Server=[Samba 3.0.2a]

Sharename	Type	Comment
-----	----	-----
print\$	Disk	
public	Disk	Public Stuff
IPC\$	IPC	IPC Service (Samba Server 3.0.2a)
ADMIN\$	IPC	IPC Service (Samba Server 3.0.2a)
hp4050n	Printer	hplaser4000 Printer hplaser4000

Anonymous login successful

Domain=[MONGROUPE] OS=[Unix] Server=[Samba 3.0.2a] ...

# Services : SAMBA

## Monter des ressources du serveur dans un système de fichiers Linux

Il est possible d'utiliser un partage de fichiers smb (Samba, Windows©) comme faisant partie du système de fichiers Linux local : il suffit de monter la ressource. Pour que cela fonctionne, il faut que le noyau de Linux ainsi que Samba aient été compilés avec le support du système de fichiers smbfs (ce qui est le cas du noyau et des paquetages Samba de Mandrake ainsi que de nombreuses autres distributions).

L'opération peut être réalisée avec la commande classique mount mais uniquement en tant que root.

### Exemple :

```
# mount -t smbfs \\\pingu\\homes /mnt/samba -o username=anne. Cet exemple permet de monter le répertoire personnel de l'utilisateur anne partagé par le serveur samba sur le répertoire /mnt/samba. Le montage nécessite de spécifier le type de système de fichiers (-t smbfs) et l'identité utilisée pour accéder au partage (-o username=anne). Ce système de fichiers peut être démonté grâce à la commande umount.
```

```
# umount /mnt/samba. Ce montage peut être effectué en "non root" grâce à la commande smbmount, à condition que le répertoire de montage soit accessible en écriture pour l'utilisateur, ainsi que le partage lui-même..
```

### Exemple :

```
$ smbmount \\\pingu\\homes /home/anne/samba. Le système de fichiers sera démonté toujours en utilisateur grâce à la commande smbmount.
```

```
$ smbmount home/anne/samba
```

# Services : APACHE

# Services : APACHE

# Services : APACHE