

Cours 1 : systemctl start 3I015 & NFS

Christophe Gonzales

3I015 — Principes et pratiques de l'administration des systèmes

- 1 Présentation générale de l'UE
- 2 Systèmes de fichiers – principe et mise en œuvre
- 3 Network File System

Objectif principal

Principes et pratique de l'administration des systèmes (Unix et Windows)

Compétences attendues

- Installation d'un parc informatique
- Gestion des utilisateurs
- Démarrage, arrêt de systèmes et de services en réseau
- Sécurisation du parc informatique
- Déploiement automatique
- Virtualisation / machines virtuelles



Accent mis sur la mise en œuvre pratique de ces concepts et sur les fichiers système utilisés !

Partie Unix

- Cours et TME : Christophe Gonzales

Partie Windows + Virtualisation

- Cours : Malika Maoui-Henda et Bruno Lesueur
- TME 1 du jeudi matin : Malika Maoui-Henda
- TME 2 du jeudi après-midi : Bruno Lesueur
- TME 3 du vendredi : Bruno Lesueur



Pas de TD, uniquement des TME

Évaluation des connaissances

note finale 1ère session = 40% contrôle continu + 60% examen

contrôle continu

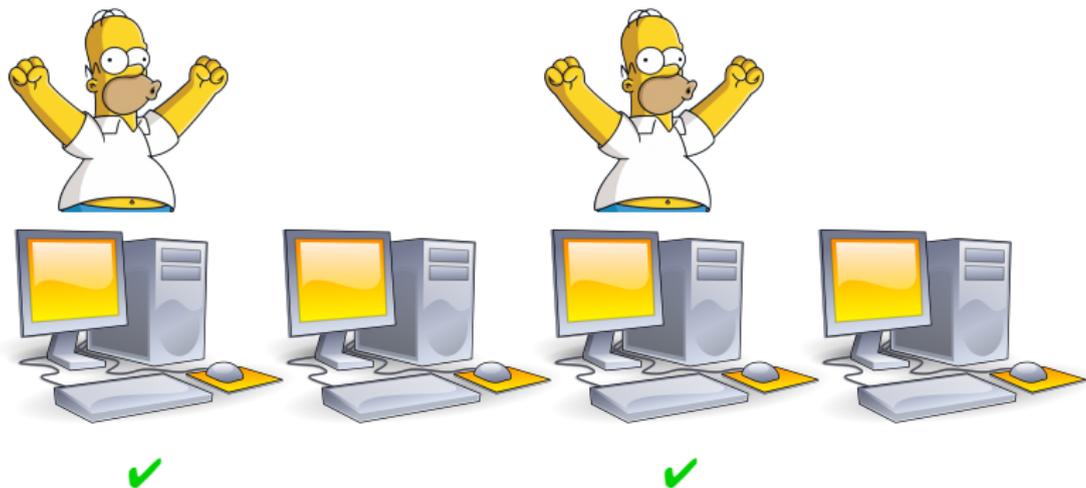
- pas de partiel
- 2 notes d'inspection des installations Unix et, éventuellement, d'interrogation orale (après les TMEs n°4 et n°7)
- 1 note de compte-rendu pour la partie Windows

note de contrôle continu = moyenne des 3 notes

- note pondérée 2ème session =
40% contrôle continu + 60% examen 2ème session

note finale 2ème session = max(note pondérée 2ème session, examen 2ème session)

Parc informatique : implications



L'utilisateur doit pouvoir travailler sur n'importe quel poste de travail !

Mise en œuvre de la partie Linux en TME (1/2)

3 machines virtuelles



msLDAP

Serveur maître LDAP
Serveur NFS
/users
Client NFS
Serveur ad-hoc

ssLDAP

Serveur NFS
/nfs_tmp
/nfs_bin
/backup
Client NFS
Serveur esclave LDAP
Serveur de sauvegardes
Passerelle vers internet

internet

c1LDAP

Poste client LDAP
Poste client NFS

c2LDAP

Déploiement
automatique (FAI)

- Mise en réseau des 3 machines virtuelles (semaine 1)
- Mise en place des clients/serveurs NFS (semaines 1 & 2)
- Création « bas niveau » d'utilisateurs (semaine 3)
- Mise en place des clients/serveurs LDAP (semaines 3 & 4)
- Mise en place de backups et d'un service (semaine 5)
- Déploiement automatique (semaine 6)
- Sécurisation des machines (semaine 7)

Exigences de qualité

- Utilisateurs \implies travail possible sur chaque poste client
 - Robustesse aux pannes
 - Sauvegarde / restauration des fichiers utilisateurs
 - Sécurité du réseau
-



Pour y parvenir :

- 1 Lire les énoncés de TME à l'avance
- 2 Réfléchir à ce que vous devrez faire en TME, éventuellement, faire des recherches sur internet

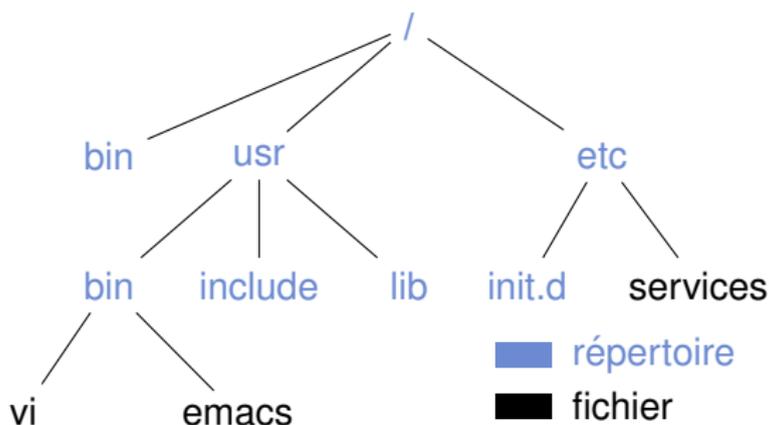


En salle de TME, aucun accès internet !

② Systèmes de fichiers

Système de fichiers sous Linux vu par l'utilisateur

C'est une arborescence



Accès au fichier :

- 1 Partir de la racine et descendre l'arborescence
- 2 Rajouter un « / » dans le nom à chaque arête suivie

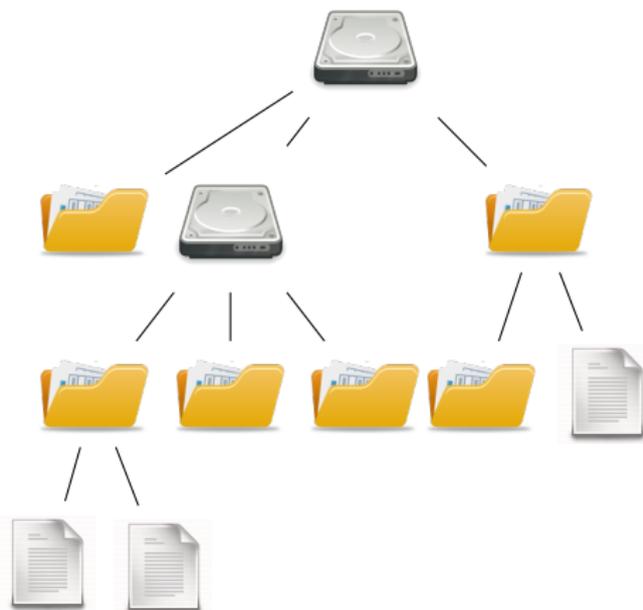
Exemple : `/usr/bin/emacs`

- Le système de fichiers est **virtuel**
 - ⇒ permet d'inclure différents systèmes de fichiers
[linux (ext4,btrfs), windows (ntfs, fat32, exfat), etc.]
de manière transparente pour l'utilisateur :
= perçus comme des **répertoires**
- Mécanisme d'inclusion : le montage (**mount** en anglais)

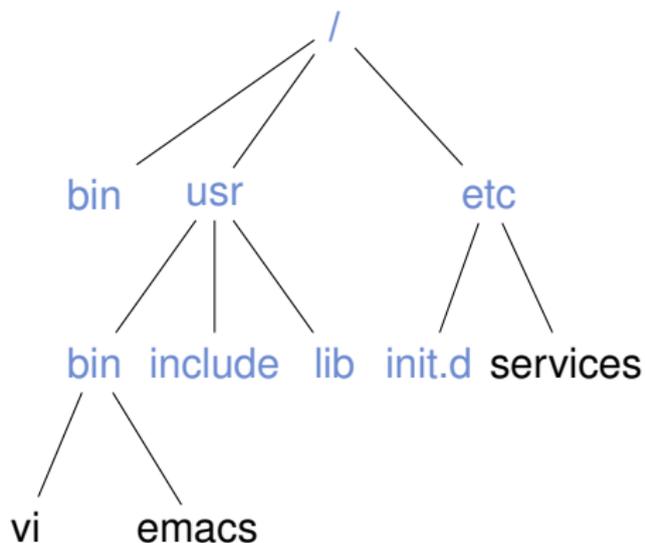
qu'est-ce qui caractérise un système de fichiers à monter ?

- c'est un **device** (par ex., une partition de disque dur)
- ce disque dur peut être distant

2 vues du système de fichiers



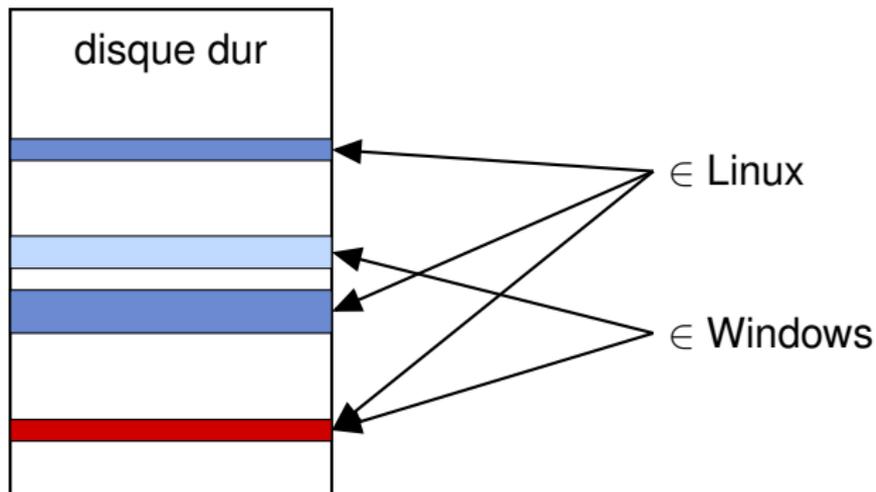
Vue administrateur



Vue utilisateur

Système de fichiers « local » (1/5)

- Système de fichiers local \implies disque dur
- **1er problème** : ordinateur avec 1 Linux + 1 Windows

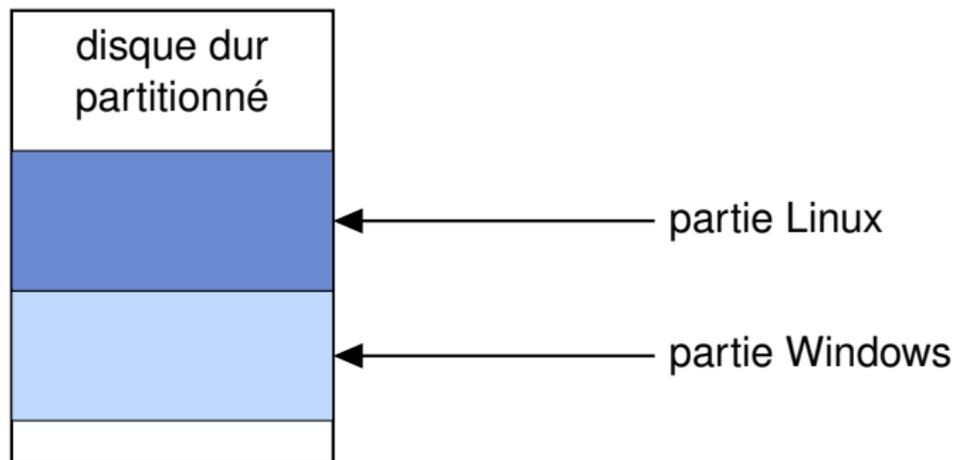


\implies Comment empêcher que Linux et Windows écrivent au même endroit (secteur rouge) ?

\implies **Solution** : partitionnement du disque dur

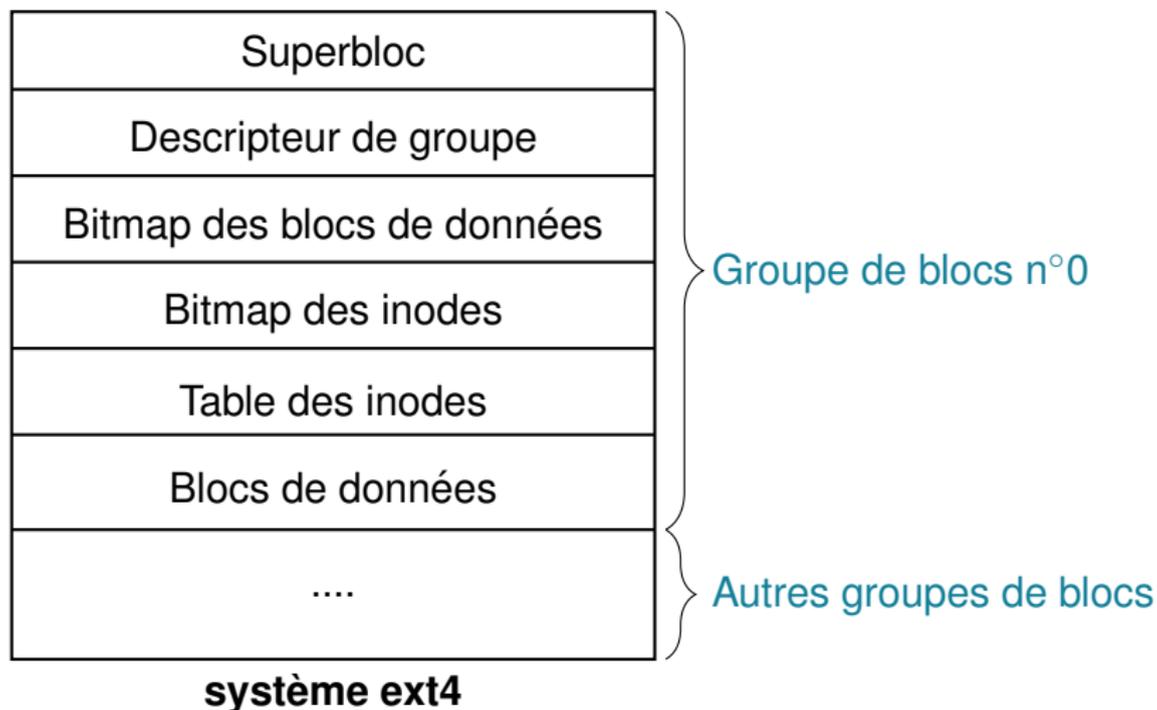
Partitionnement

Réservation d'espace pour un système de fichiers sur le disque dur (infos inscrites au début du disque).



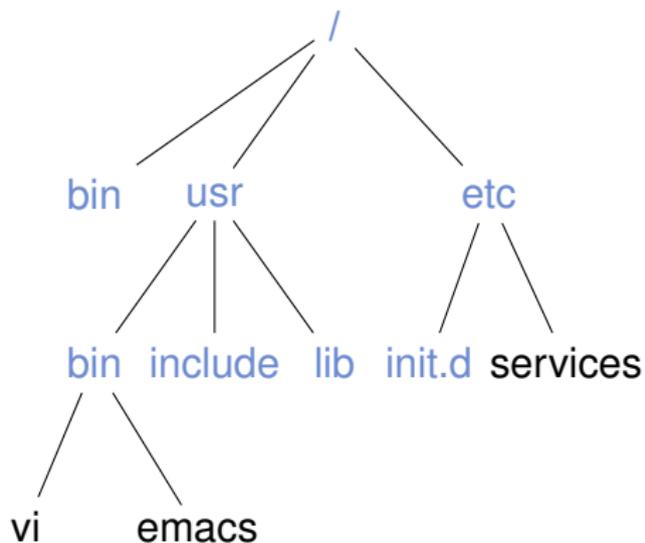
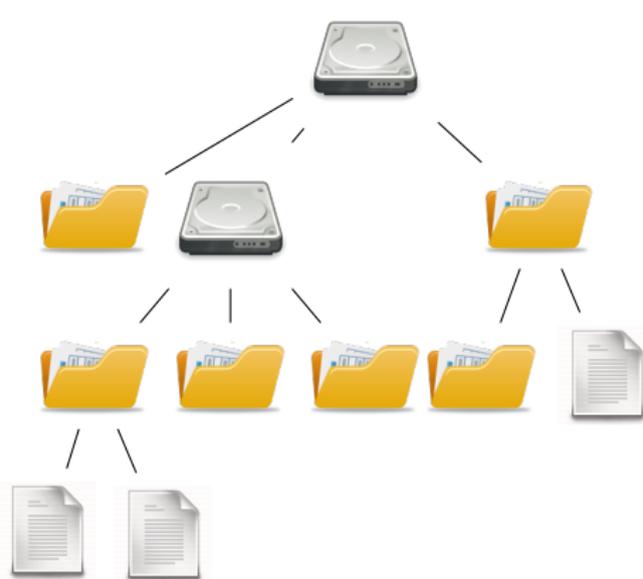
⇒ Linux lira/écrira uniquement dans la partie en bleu foncé
Windows lira/écrira uniquement dans la partie en bleu clair

- **2ème problème** : comment retrouver un fichier dans la partition ?
- **Solution** : formatage de la partition \implies organisation



Système de fichiers « local » (4/5)

- **3ème problème** : comment faire pour que les partitions soient perçues comme des répertoires ?
- **Solution** : montage de la partition



⇒ il faut créer le répertoire de montage !

Mise en place de systèmes de fichiers locaux

4 étapes essentielles :

- 1 Partitionner le disque dur (`fdisk`)
- 2 Formater les partitions (`mkfs`)
- 3 Créer les répertoires de montage (`mkdir`)
- 4 Monter les partitions (`mount`)

Partitionnement (1/3)

- 3 types de partitions : primaire, étendue, logique



4 partitions non logiques au plus !



1 partition étendue au plus !



la partition étendue sert de conteneur de partitions logiques



- 1 créer au plus 3 partitions primaires
- 2 dès la 3ème créée, créer 1 partition étendue s'étendant sur tout le reste du disque
- 3 créer des partitions logiques dedans si besoin

Partitionnement (2/3)

- l'utilitaire `fdisk`

```
fdisk /dev/sda
```

```
Command (m for help):
```

- quelques commandes:

d	détruire une partition	l	liste des id de partitions
m	aide	n	ajout d'une nouvelle partition
p	afficher la table des partitions	q	sortir sans sauvegarde
t	changer le type d'une partition	w	sortir en sauvegardant

- **Exemple :**

```
Command (m for help):p
```

```
Device      Boot      Start          End      Blocks Id System
/dev/sda1   *                2048    39063551    19530752 83 Linux
/dev/sda2                39063552    97656831    29296640 83 Linux
/dev/sda3                97658878    500117503    201229313  5 Extended
/dev/sda5                97658880    107421695     4881408 82 Linux swap / Solaris
/dev/sda6                107423744    400000000    146288128  7 HPFS/NTFS/exFAT
```

Partitionnement (3/3)

- créer une nouvelle partition :

```
Command (m for help): n
```

```
Partition type:
```

```
p primary (2 primary, 1 extended, 1 free)
```

```
l logical (numbered from 5)
```

```
Adding logical partition 7
```

```
First sector (400000001, default 500117503):
```

```
Using default value 916881408
```

```
Last sector, +sectors or +sizeK,M,G (400000001-500117503,  
default 500117503): +10M
```

```
Command (m for help): p
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	39063551	19530752	83	Linux
/dev/sda2		39063552	97656831	29296640	83	Linux
/dev/sda3		97658878	500117503	201229313	5	Extended
/dev/sda5		97658880	107421695	4881408	82	Linux swap / Sol
/dev/sda6		107423744	400000000	146288128	7	HPFS/NTFS/exFAT
/dev/sda7		400000001	400020480	10240	83	Linux

```
(m for help): t
```

```
Partition number (1-7): 7
```

```
Hex code (type L to list codes): b
```

```
Changed system type of partition 7 to b (W95 FAT32)
```

Comment choisir la taille des partitions ?

/boot	moins de 1 Go
/var	dépend des serveurs (mail, web, <i>etc.</i>) — ~5-20 Go
/tmp	dépend des applications exécutées — ~5-20 Go
/	environ 20-25 Go
/home	dépend du nombre d'utilisateurs, des quotas
swap	dépend des applications s'exécutant en même temps, maintenant souvent inutile



le swap peut être créé à la volée en tant que fichier
(cf. les commandes `mkswap`, `swapon`, `swapoff`)

- Mise en place du système ext4 = formatage : `mkfs`

Exemple : `mkfs -t ext4 /dev/sda6`

 formatage \implies perte de toutes les données de `/dev/sda6`

- 1 / est toujours « monté » au démarrage de Linux
- 2 pour monter un device formaté /dev/sda6 en /toto :
 - a créer le répertoire /toto (point de montage) s'il n'existe pas :

```
mkdir /toto
```
 - b utiliser la commande de montage `mount`

```
mount /dev/sda6 /toto
```



La commande `mount` ne monte le device que jusqu'à l'extinction de la machine !

⇒ au prochain reboot, on devra refaire l'étape **b** (`mount`)

Comment faire un montage automatique au démarrage ?

- fichier `/etc/fstab` contient des instructions de montage
- ses instructions sont exécutées au démarrage

⇒ étape **c** : éditer `/etc/fstab` pour rajouter le point de montage `/toto`

⇒ Au prochain démarrage de Linux, `/toto` sera monté !

Anatomie d'un fichier /etc/fstab

```
[root@msLDAP gonzales]# more /etc/fstab
```

```
#<file system> <mount point> <type> <options> <dump> <pass>
/dev/sda1      /                ext4  errors=remount-ro  0      1
/dev/sda2      /home           ext4  defaults            0      2
/dev/sda3      none            swap  sw                  0      0
/dev/sda6      /toto           ext4  defaults            0      2
```

champ	signification
1	nom du device
2	point de montage
3	type de système de fichier (ext4, nfs, swap, etc.)
4	options de montage
5	le device doit-il être backupé ($\neq 0$) ou non (0)
6	ordre dans lequel vérifier les devices au boot



champ 6 = 1 pour /, 2 pour les devices ext4 « locaux », 0 sinon



un des buts en TME : choisir les bonnes options !

Dans le TME n°1, vous aurez à :

- 1 sélectionner le disque dur à installer (`/dev/sda`)
- 2 partitionner le disque dur avec `fdisk`
- 3 formater la ou les partition(s) avec `mkfs`
- 4 créer le ou les point(s) de montage avec `mkdir`
- 5 monter la partition avec `mount`
- 6 mettre à jour le fichier `/etc/fstab`

- 1 Toujours tester **manuellement** !
- 2 **Uniquement quand c'est ok**, pérenniser l'installation.
- 3 **Vérifier** que la pérennisation est correcte !

Pérennisation des montages

Ajouter les montages à effectuer dans le fichier `/etc/fstab`

- Vérification de la pérennisation :

Tester des `umount` (démontage) et des `mount` des montages ajoutés et modifier `/etc/fstab` si erreurs :

Si `/dev/sda6` monté en `/toto`, faire
`umount /toto` puis `mount /toto`



Ne pas éteindre ou redémarrer le système tant que le `mount /toto` produit des erreurs !

Mode = type + droits

```
[root@msLDAP gonzales]# ls -l /
```

```
drwxr-xr-x  2 root root   4096 janv.  4  2014 bin
drwxrwxrwt 14 root root   4096 janv. 12 17:09 tmp
lrwxrwxrwx  1 root root     30 janv.  4  2014 vmlinuz -> boot/vmlinuz-3.11.0-12-gen
-rwxr-xr-x  1 root root 188296 avril 29  2013 toto
```

- 3 classes d'ayants droit : **u**(utilisateur), **g**(roupe), **o**(reste du monde)
- droits d'accès : **r**(ead), **w**(rite), **x**(execute)
- absence de droit : -
- commande pour modifier les droits : `chmod`

rajouter les droits en exécution pour l'utilisateur : `chmod u+x /toto`

enlever les droits en lecture sauf pour l'utilisateur : `chmod go-r /toto`

définir explicitement les droits : `chmod g=rx,o= /toto`

`chmod 650 /toto`

Extra droits : set-uid, set-gid

Bit set-uid (s)

Permet d'exécuter un programme avec les droits de son propriétaire.

```
[root@msLDAP gonzales]# ls -l /usr/bin/passwd  
-rwsr-xr-x 1 root root 47032 juil. 26 2013 /usr/bin/passwd
```

⇒ exécuté avec les droits de root

Bit set-gid (s)

Permet d'exécuter un programme avec les droits du groupe du programme.

```
[root@msLDAP gonzales]# ls -l /usr/bin/ssh-agent  
-rwxr-sr-x 1 root ssh 129120 avril 30 2014 ssh-agent
```

⇒ exécuté avec les droits de ssh pour tout utilisateur et avec ceux de root pour root

Affichage : s si le droit x existe, S sinon

Extra droits : le sticky bit (t)

sticky bit pour un répertoire

Il interdit la suppression d'un fichier du répertoire à tout utilisateur autre que le propriétaire du fichier.

```
[root@msLDAP gonzales]# ls -la /tmp
drwxrwxrwt 14 root root 4096 janv. 12 18:39 .
-r----- 1 toto titi 6240 janv. 12 07:58 planning.xlsx
```

⇒ tout le monde peut écrire dans /tmp
mais seul toto peut supprimer planning.xlsx

sticky bit pour un fichier

Il indique la résidence d'un processus en mémoire centrale (data) et de swap (code) après son exécution.

Affichage : t si le droit x existe pour l'ayant droit o, T sinon

Quelques commandes utiles

commande	utilité
parted	partitionnement (plus évolué que fdisk)
mount (sans option)	liste des devices montés
df (disk free)	affiche l'espace utilisé/disponible des devices montés
dumpe2fs	affichage des infos d'un système ext(4)
tune2fs	ajustement des paramètres d'un système ext(4)
dd	permet de copier des octets d'un fichier/device dans un autre
ls	affiche la liste des fichiers d'un répertoire
file	détermine le type d'un fichier
stat	affiche des infos contenues dans l'inode d'un fichier, répertoire

③ Le Network File System

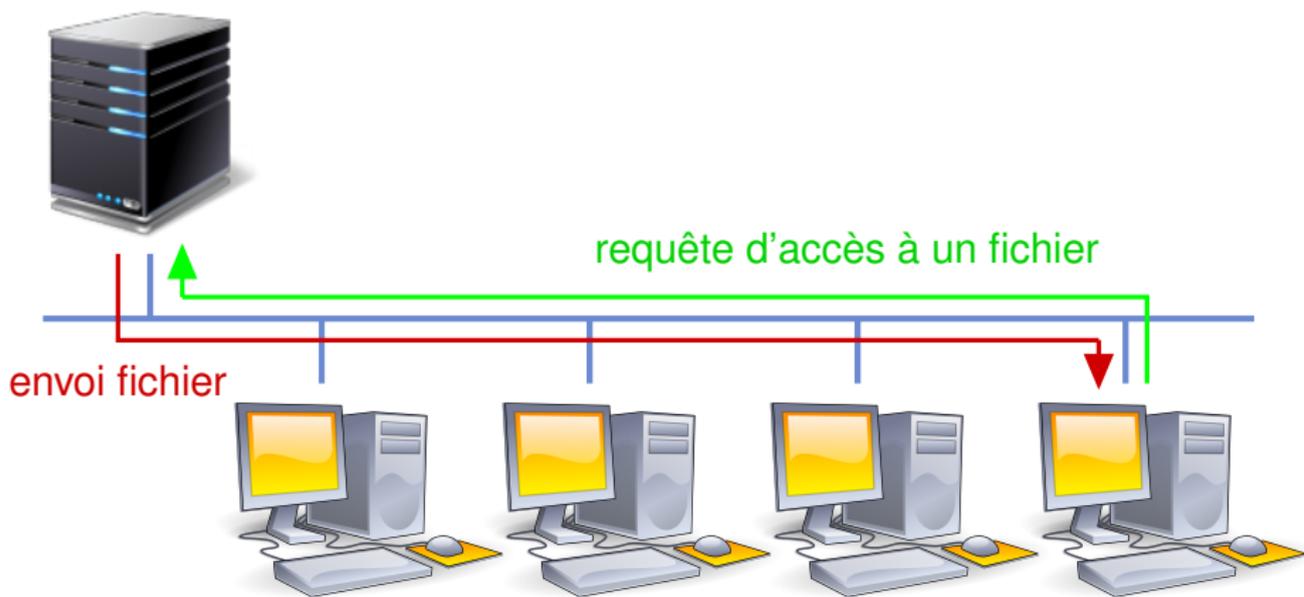
Principe général



⇒ les données utilisateur ne doivent pas résider sur son poste de travail

⇒ { les placer sur un serveur (NFS)
postes de travail = clients

Architecture client/serveur

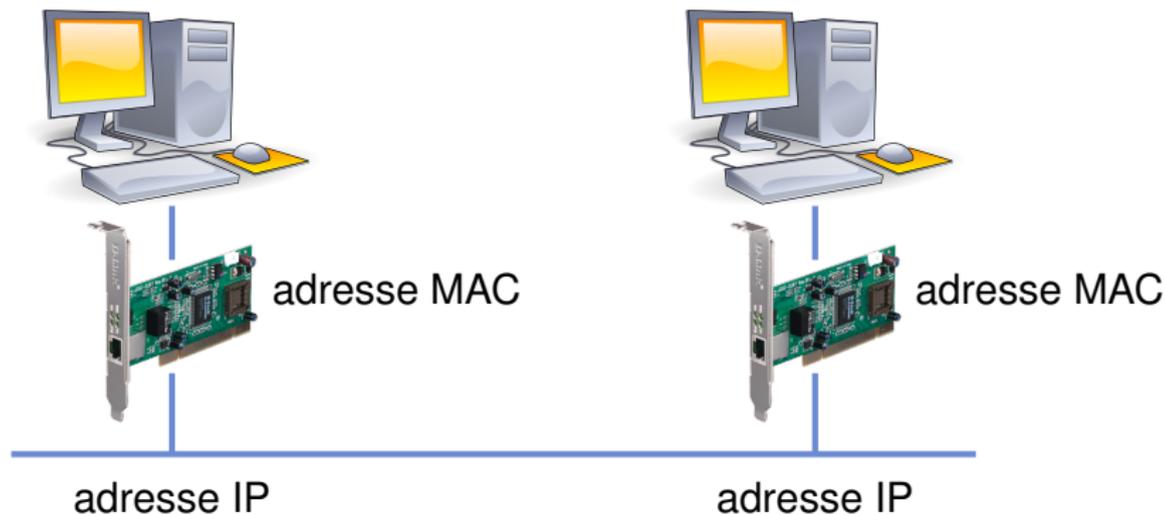


- 1 Mise en place du réseau
- 2 Mise en place du serveur NFS
- 3 Mise en place du client NFS



Bien tester le bon fonctionnement de chaque étape avant de passer à la suivante !

Mise en place du réseau



⇒ associer aux cartes réseau une adresse IP, des tables de routage, *etc.*

3 services :

- systemd-networkd ❌
- NetworkManager ❌
- networking ✔

Noms des interfaces réseau

- Carte réseau \implies identifié par adresse MAC
Exemple : 08:00:27:79:61:b8
- Identification plus “user-friendly” : nom d’interface
- Anciens noms des interfaces (jusqu’à Debian 8, Ubuntu 15.10) : eth0, eth1, wlan0, *etc.*
- Nouveaux noms : enp0s3, enp0s8, ens1, *etc.*

[<https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames>]

systemd-networkd

- Fichiers de configuration :
/etc/systemd/network et /lib/systemd/network
- Exemple : /etc/systemd/network/10-enp0s3.network
[Match]
MACAddress=08:00:27:79:61:b8

[Network]
Address=192.168.X.Y/24
- Inconvénient : restart n'update pas les interfaces configurées
⇒ besoin de rebooter pour mettre à jour les interfaces

[<http://www.freedesktop.org/software/systemd/man/systemd.network.html>]



ci-dessus : remplacer 192.168.X.Y par votre adresse IP !

NetworkManager

- Fichiers de configuration :

```
/etc/NetworkManager/system-connections
```

- Exemple :

```
/etc/NetworkManager/system-connections/enp0s3
```

```
[connection]
```

```
id=enp0s3
```

```
interface-name=enp0s3
```

```
type=ethernet
```

```
[ethernet]
```

```
mac-address=08:00:27:79:61:B8
```

```
[ipv4]
```

```
method=manual
```

```
address1=192.168.X.Y/24
```

Service networking

- Fichiers de configuration :

`/etc/default/networking` et `/etc/network/interfaces`

- Exemple : `/etc/network/interfaces`

```
auto enp0s3
iface enp0s3 inet static
    name enp0s3
    address 192.168.X.Y
    network 192.168.X.0
    netmask 255.255.255.0
    broadcast 192.168.X.255
```

Fichier `/etc/default/networking` :

```
CONFIGURE_INTERFACES=yes
```

Utilisation de systemd

```
systemctl action nom_de_service
```

Exemple : `systemctl start networking.service`

- Commandes *systemd* valables pendant la session courante (jusqu'à extinction/redémarrage de la machine) :

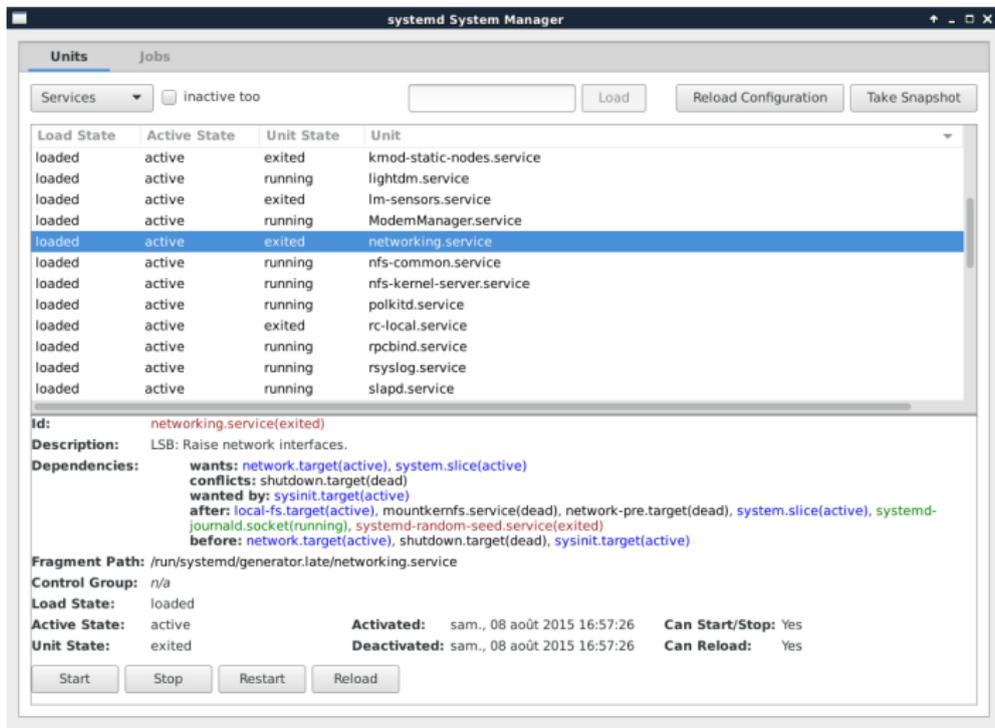
action	commande
démarrage d'un service	<code>systemctl start service</code>
arrêt d'un service	<code>systemctl stop service</code>
redémarrage d'un service	<code>systemctl restart service</code>
état d'un service	<code>systemctl status service</code>

- Commandes *systemd* de pérennisation (prochain boot) :

action	commande
activer un service	<code>systemctl enable service</code>
désactiver un service	<code>systemctl disable service</code>

systemd : interface graphique

● Interface graphique : systemadm



The screenshot shows the 'systemd System Manager' window. The 'Units' tab is active, displaying a table of units. The 'networking.service' unit is selected and highlighted in blue. Below the table, the details for 'networking.service(exited)' are shown, including its description, dependencies, and control group.

Load State	Active State	Unit State	Unit
loaded	active	exited	kmod-static-nodes.service
loaded	active	running	lightdm.service
loaded	active	exited	lm-sensors.service
loaded	active	running	ModemManager.service
loaded	active	exited	networking.service
loaded	active	running	nfs-common.service
loaded	active	running	nfs-kernel-server.service
loaded	active	running	polkitd.service
loaded	active	exited	rc-local.service
loaded	active	running	rpcbind.service
loaded	active	running	rsyslog.service
loaded	active	running	slapd.service

Id: networking.service(exited)
Description: LSB: Raise network interfaces.
Dependencies: wants: network.target(active), system.slice(active)
conflicts: shutdown.target(dead)
wanted by: sysinit.target(active)
after: local-fs.target(active), mountkernfs.service(dead), network-pre.target(dead), system.slice(active), systemd-journald.socket(running), systemd-random-seed.service(exited)
before: network.target(active), shutdown.target(dead), sysinit.target(active)
Fragment Path: /run/systemd/generator/late/networking.service
Control Group: n/a
Load State: loaded
Active State: active
Unit State: exited
Activated: sam., 08 août 2015 16:57:26
Deactivated: sam., 08 août 2015 16:57:26
Can Start/Stop: Yes
Can Reload: Yes

● En TME : on utilisera la ligne de commande !

- 1er test : la commande ifconfig

```
[root@msLDAP /]# ifconfig
enp0s3  Link encap:Ethernet HWaddr 08:00:27:79:61:B8
        inet addr:192.168.X.Y Bcast:192.168.X.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 b) TX bytes:4756 (4.6 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:120 errors:0 dropped:0 overruns:0 frame:0
        TX packets:120 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:12448 (12.1 KiB) TX bytes:12448 (12.1 KiB)
```

- 2ème test : la commande ping

```
[root@msLDAP /]# ping 192.168.X.Z
PING 192.168.X.Z (192.168.X.Z) 56(84) bytes of data.
64 bytes from 192.168.X.Z: icmp_req=1 ttl=64 time=4.40 ms
64 bytes from 192.168.X.Z: icmp_req=2 ttl=64 time=1.07 ms
```



modifier /etc/network/interface tant que ping affiche :

```
[root@msLDAP /]# ping 192.168.X.Z
PING 192.168.X.Z (192.168.X.Z) 56(84) bytes of data.
From 192.168.X.Y icmp_seq=2 Destination Host Unreachable
From 192.168.X.Y icmp_seq=3 Destination Host Unreachable
From 192.168.X.Y icmp_seq=4 Destination Host Unreachable
```

Mise en place de l'architecture NFS

- Serveur \implies service `nfs-kernel-server`
- Client \implies service `nfs-common` (démarré automatiquement)
- `nfs-kernel-server/nfs-common` s'appuient sur l'organisateur de services portmapper (service `rpcbind`)
- le portmapper utilise la couche ISO RPC (Remote Procedure Call)
- services RPC utiles :
 - `rpcidmapd` : associe noms d'utilisateurs et (UID,GID)
 - `rpcgssd` : sécurité utile pour NFSv4



Ces services doivent être démarrés au boot

- Serveur NFS \implies à quels fichiers permet-il d'accéder ?

fichier `/etc/exports`

- **Syntaxe** : répertoire_à_exporter clients(options)
- **Exemple du manuel** (man 5 exports) :

```
/          master(rw) trusty(rw,no_root_squash)
/projects  proj*.local.domain(rw)
/usr       *.local.domain(ro)
/home/joe  pc001(rw,all_squash,anonuid=150,anongid=100)
/pub       *(ro,insecure,all_squash)
/toto      192.168.1.1(rw,root_squash)
```



Choisir les bonnes options = un des challenges en TME !

Serveur NFS : option de squashing

UID=100,GID=100



requête

UID=20,GID=50



traitée avec UID,GID=100

- 1 utilisateur sur poste client lit /toto/titi
⇒ requête d'accès à /toto/titi avec UID=20,GID=50
- 2 squash ⇒ avant traitement : UID,GID changés en
UID,GID anonymes = 100,100
- 3 traitement requête avec droits de UID=100,GID=100

Exemples d'options

/etc/exports de 192.168.X.1 :

```
/toto 192.168.X.2(rw,root_squash)
/titi 192.168.X.3(ro,root_squash)
/tutu 192.168.X.0/24(ro,no_root_squash)
```

	machine	user	accès
/toto	192.168.X.2	joe	lecture/écriture, droits d'accès de joe
/toto	192.168.X.2	root	lecture/écriture, droits d'accès anonyme
/titi	192.168.X.2	joe	aucun accès
/titi	192.168.X.3	joe	lecture, droits d'accès de joe
/titi	192.168.X.3	root	lecture, droits d'accès anonyme
/tutu	192.168.X.2	joe	lecture, droits d'accès de joe
/tutu	192.168.X.2	root	lecture, droits d'accès de root

Les options ro et rw

- /toto : répertoire exporté par NFS
 /toto/titi : fichier régulier

Question : /toto/titi accessible en écriture ?

option export	ls -l titi	écriture ?
ro	-rw-rw-rw-	✘
rw	-rw-rw-rw-	✔
rw	-r--r--r--	✘

⇒ autorisation en 2 temps :

- 1 Il faut avoir les droits rw dans /etc/exports
- 2 sur le serveur NFS, il faut avoir les droits en écriture
(-rw-rw-rw-)

Pensez au piratage : limiter les accès

- 1 À moins d'avoir besoin des droits en écriture, choisissez `ro` plutôt que `rw`
- 2 En salle utilisateur, vous pouvez vous faire pirater
 - ⇒ un utilisateur peut passer `root`
 - ⇒ `root_squash` lui bloque des accès
- 3 En salle serveur, si l'on a besoin qu'un admin d'une machine cliente ait des accès aux fichiers système : `no_root_squash`

Serveur NFS : idmapd

idmapd : convertisseur identifiant nfs ↔ nom

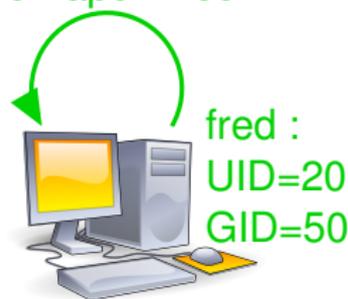
idmapd : UID=500,GID=30



requête : fred

traitée avec :
UID=500
GID=30

idmapd : fred



- NFSv4 transmet des noms de la forme user@domain
- idmapd traduit ces noms en UID/GID et réciproquement
⇒ permet d'avoir des UID/GID différents suivant les machines

- Fichier de configuration : `/etc/idmapd.conf`

`Domain = nfs_3i015`



même nom de domaine pour le serveur NFS et ses clients !

- Éventuellement fichier `/etc/default/nfs-common` :

`NEED_IDMAPD=yes` ou `NEED_IDMAPD=no`



FAI (TME n°6) ne supporte pas idmapd !

⇒ en TME, `NEED_IDMAPD=no`

Mise en place du serveur NFS : résumé

① éditer les fichiers `/etc/exports` et `/etc/idmapd.conf`

② relancer le service `nfs` :

```
systemctl restart nfs-kernel-server.service
```



tant qu'il y a des erreurs : modifier `/etc/exports`

③ vérifier que tout est ok :

exécuter la commande `exportfs`

⇒ affiche les répertoires exportés



tant que les répertoires ne sont pas ok, modifier

`/etc/exports` et relancez le service `nfs-kernel-server`

④ mettre à jour les services (`nfs`, *etc.*) exécutés au démarrage

Client NFS : importation

- Client NFS \implies à quels fichiers accède-t-il ?

fichier `/etc/fstab`

- **Syntaxe** : la même que pour les devices « locaux » **mais** :

- nom de device = serveur:répertoire_exporté

- type de système de fichiers = nfs



options : voir `man mount` et `man 5 nfs`



bonnes options = un des challenges en TME

- **Exemple d'entrée de la fstab** :

```
192.168.X.2:/toto /titi nfs defaults 0 0
```

\implies répertoire `toto` de `192.168.X.2` importé en répertoire `titi`

- Tester avec la commande `mount (/titi)`

Quelques options d'importation NFS



réseau \implies possibilité de pannes !

\implies prévoir le comportement en cas de panne

● Extrait de `man 5 nfs` :

`soft` si problème NFS, après délai de retransmission : la requête du client échoue et une erreur est retournée à l'application qui a émis la requête

`hard` si problème NFS, la requête du client est réémise indéfiniment

`bg` lors d'un mount, si problème NFS, réessaye le montage pendant un certain temps, en background (rend la main). Si problème non résolu après ce délai, le montage échoue

`fg` lors d'un mount, si problème NFS, réessaye le montage sans rendre la main

- 1 mettre en place le réseau (service networking) sur le serveur et le client
- 2 vérifier le bon fonctionnement avec `ping`
- 3 installer le serveur nfs (fichiers `/etc/exports` et `/etc/idmapd.conf`, démarrage du service `nfs-kernel-server`)
- 4 vérifier le bon fonctionnement avec `exportfs`
- 5 mettre en place le client (fichiers `/etc/fstab`, `/etc/idmapd.conf`)
- 6 vérifier le bon fonctionnement avec `mount`
- 7 mettre à jour les services démarrés au boot (`nfs-kernel-server`, *etc.*)

... voilà, votre réseau et votre NFS fonctionnent

- 8 tester leur robustesse en simulant des pannes