Administration Système

Amal KAMMOUN (Diapos de Xavier MONNIN)

Bureau E202
Université Paris 13
amal.kammoun.2@gmail.com

Objectifs du cours

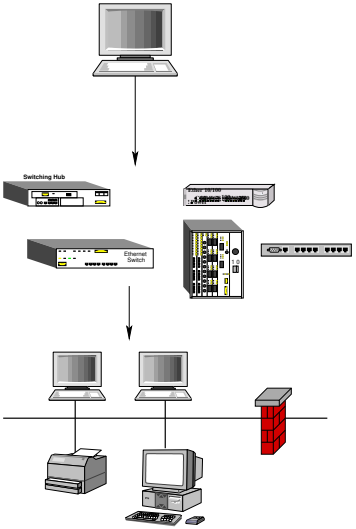
- Acquérir un savoir-faire théorique :
  - en matière d'administration des systèmes d'exploitation
  - au sein d'un réseau informatique
- Acquérir un savoir-faire pratique en TP via :
  - la mise en place et l'administration d'un petit réseau informatique (machines PC/Linux)
  - l'utilisation du logiciel d'émulation réseau Marionnet (`http://www.marionnet.org/`)
  - l'utilisation du logiciel de virtualisation VirtualBox

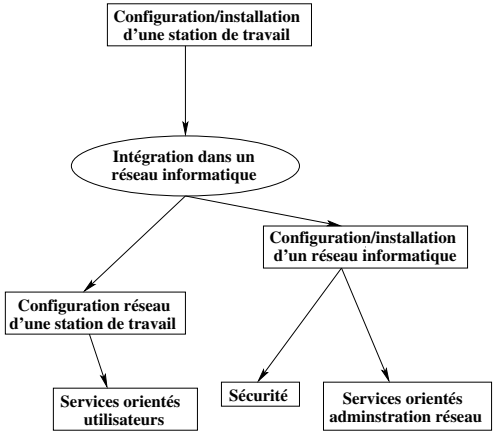
Programme des enseignements (1)

- Administration d'une station de travail
- Administration d'un serveur
- Administration d'un réseau informatique de taille moyenne

$\longrightarrow$ centrée sur les systèmes UNIX (Linux – Debian) Mais des concepts généraux

Programme des enseignements (2)





Répartition des enseignements

Cours :

- 6 séances de 1h30

TPs :

- 10 séances de 3h

Administration d'un réseau informatique

Des connaissances variées dans différents domaines :

- Systèmes d'exploitation
- Programmation au niveau du système
- Réseaux et services informatiques
- Sécurité

Administration système

Configuration et administration des systèmes UNIX :

- De nombreuses divergences entre les constructeurs
  Chaque système :
  - possède ses spécificités
  - mais tente de suivre les standards (norme POSIX, ANSI C...)

Complexité de l'administration

- diversité des systèmes
- partage des ressources
- sécurité

Nécessite :

- une bonne connaissance des caractéristiques principales des différentes versions (et donc des commandes UNIX !)
- une expertise, les spécificités de chaque système

Définitions

- **Système :** ensemble des programmes permettant d'accéder à une machine et d'utiliser ses périphériques
- **Réseau :** ensemble des dispositifs (câbles, switchs, routeurs, stations de travail...) connectés entre eux et formant une entité globale vue de l'extérieur
- **Service :** ressource (DNS, Web, NIS, LDAP, Messagerie...) offerte par un programme situé sur une machine (serveur) et, accessible par des machines (clientes) situées sur le même réseau ou à l'extérieur

Bref historique des systèmes UNIX

- Version 1 en 1970, (Laboratoire Bell, AT&T)
- Version 6 en 1975
- Version 7 en 1978

Trois branches principales :

- BSD (Berkeley, Software Distribution)
- Recherche (Laboratoire Bell)
- System (AT&T, Commercial) dit System V

$\rightarrow$ Au total : plus de 150 UNIX depuis 1970

Man

Les pages de manuel des commandes UNIX sont réparties en chapitres appelés des sections :

- section 1 : commandes normales
- section 2 : appels systèmes
- section 3 : fonctions de programmation C
- section 4 : périphériques et pilotes de périphériques
- section 5 : format de fichiers système
- section 6 : jeux
- section 7 : divers
- section 8 : commandes de gestion du système

getopt(3) $\rightarrow$ la commande getopt de la section 3 du manuel

Autres sources

- *Frequently Asked Questions* (FAQ)
- *Forums* et *Newsgroups* (`comp.unix.*`, `fr.comp.os.*`)
- *MailingLists*
- Les documentations constructeurs (`www.ibm.com`, `docs.sun.com`, `ftp.lip6.fr/pub/linux`...)

Quelques tâches d'un administrateur système

- Gérer les comptes utilisateurs (tâche simple et automatisable)
- Assister et éduquer les utilisateurs (répondre à leurs questions, documentation à jour pour les outils en place)
- Gérer les logiciels :
  - Installer
  - Configurer
  - Mettre à jour (*patcher*)
- Gérer le matériel :
  - Panne
  - Remplacement
  - Ajout

Quelques tâches d'un administrateur système

- Assurer la sécurité du système et des utilisateurs :
  - Sauvegardes fiables et régulières
  - Contrôle d'accès
  - Utilisations abusives de ressources
- Vérifier l'adéquation du matériel avec son utilisation (identifier les goulets d'étranglement)
- Assurer la maintenance de premier niveau :
  - Diagnostiquer une panne
  - Appel de la maintenance constructeur
- Gérer quotidiennement (multiples tâches, petites ou grosses)

Autres facettes du métier

- Diplomatie, police
- Aspects légaux (chiffrement, Cnil...)
- Enquêtes judiciaires (vol, saccage, piratage informatique, articles pédophiles...)
- Relations commerciales
- Politique d'utilisation des machines

$\rightarrow$ L'administrateur est en première ligne lorsqu'un problème surgit
C'est lui qu'on incrimine naturellement lorsque quelque chose ne fonctionne pas

Connaissances de base

$\rightarrow$ Expert Unix

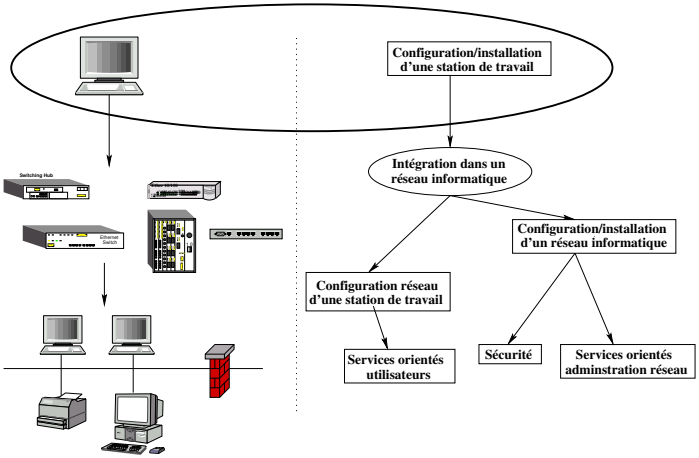
- Environnement utilisateur
- Aide en ligne
- Système de fichiers
- Utilisation du shell
- Utilisation d'un éditeur de texte
- Commandes de base
- Programmation shell

Administrateur système

3 qualités nécessaires :

- Technicité
- Rigueur
- Bon sens

Méthodologie d'administration



Administration système

- Administrer un système est une lourde responsabilité
- L'ampleur de la tâche est variable selon les sites

Dans tous les cas, il faut :

- Veiller au bon fonctionnement du réseau
- Avoir à l'esprit la sécurité du système et du réseau
- Mettre à disposition les outils nécessaires aux utilisateurs
- Gérer et tenir compte du comportement des utilisateurs :
  - Éviter les abus de pouvoir
  - Éviter la paranoïa : la plupart des utilisateurs gênant les fonctionnement d'une machine n'en ont pas conscience
  - Établir des règles de conduite avec les utilisateurs

Méthodologie d'administration (1)

- Garantir l'intégrité des bases de données système et leur mise à niveau
- Consigner :
  - Commandes tapées lors d'installation (notamment sur les serveurs)
  - Opérations effectuées sur les systèmes lors des configurations spécifiques
- Se tenir informer des évolutions des systèmes et du domaine (mesures à prendre en cas de problème de sécurité)
- Documenter

Méthodologie d'administration (2)

- Identifier les bases de données système
- Conserver une version de référence avant toute modification `/etc/inetd.conf.orig` pour `/etc/inetd.conf`
- Assurer une sauvegarde régulière (quotidienne ou hebdomadaire) automatique de ces fichiers sur des supports robustes (CD, bandes, disques externes...)
- Corriger les *bugs* des logiciels en appliquant les patches ou les mises à jour

Administration d'un système d'exploitation

Administration d'un système et d'un réseau :

$\longrightarrow$Différent de l'administration d'un ordinateur mono-utilisateur

- Utiliser les outils graphiques d'administration spécifiques proposés par les constructeurs ne permet pas d'acquérir l'expérience nécessaire pour s'adapter aux évolutions ou aux changements des systèmes
- Administration à l'aide de `scripts` (bash, perl, python...)
  - Création de centaines de comptes utilisateurs
  - Modification ou mise au jour de la configuration de plusieurs systèmes

Utilisation du compte `root`

Pour éviter les erreurs aux conséquences catastrophiques :
utiliser le compte `root` uniquement lorsque c'est nécessaire Quelques règles :

- Vérifier les commandes tapées avant leur exécution
- Utiliser `rm -i` plutôt que `rm` (placer un alias dans l'environnement du root)
- NB :
  Invite `root` : #
  Invite utilisateur ordinaire : $

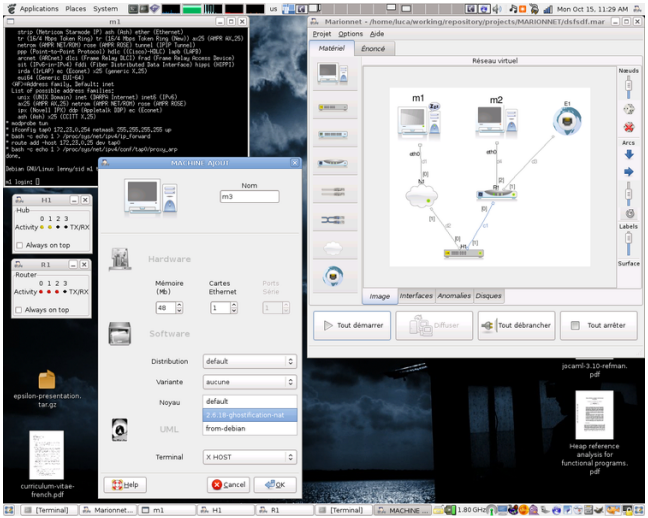
En travaillant sous root, vous ferez une erreur ...
... un jour ! ;-)

Présentation de Marionnet

- Système d'émulation d'un réseau informatique basé sur des machines Linux (Debian)
- Possibilité de configuration d'un réseau (switch, hub, station de travail)
- Simulation d'incidents
- Accès aux systèmes Linux pour une configuration complète

`http://www.marionnet.org`

Capture d'écran



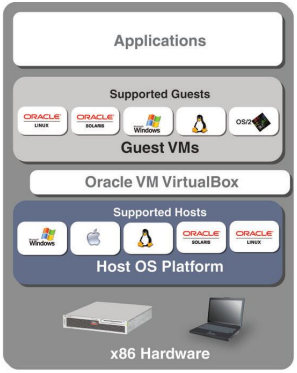
Exécution de Marionnet

- Dans les salles de TP, commande `marionnet`
- Dans une machine virtuelle VirtualBox
  Image disponible sur la page :

`http://marionnet.org/download/Marionnet.ova`

Présentation de Virtualbox : Virtualisation

$\longrightarrow$ Créer/Utiliser des machines virtuelles



Terminologie

Système hôte (host) : système d'exploitation principal qui permet de faire fonctionner VBox

Système invité (guest) : système d'exploitation installé à l'intérieur d'une VM

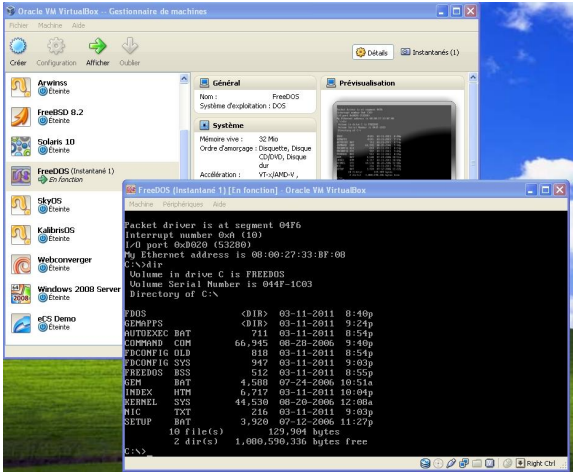
Machine virtuelle : ordinateur virtuel créé par VBox

VDI (Virtual Disk Image) : fichier (unique) contenant le Système invité

Présentation de VirtualBox

- `https://www.virtualbox.org`
- Logiciel libre (GPL)
- Système hôte :
  - Architecture Intel/AMD
  - Systèmes Linux, Windows, MacOS, Solaris.
- Système invité :
  - La plupart des SE disponibles sur architecture Intel : Linux, Windows, DOS, Unices, etc.
  - `https://www.virtualbox.org/wiki/Guest_OSes`

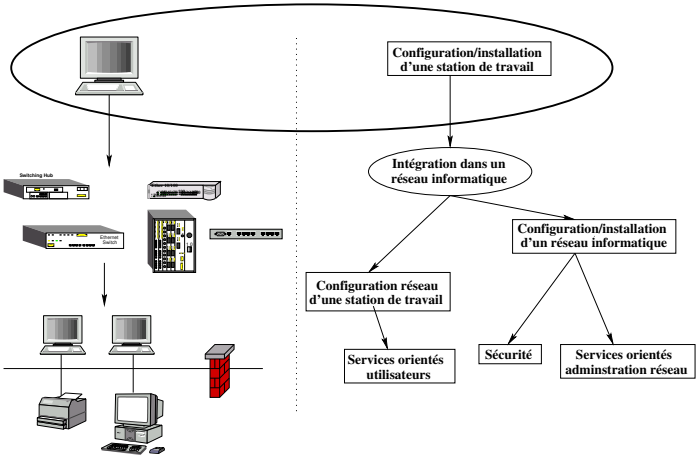
Capture d'écran



2 Administration d'une station de travail

- Administration des utilisateurs et des groupes
- Démarrage d'une station de travail
- Exécution automatique de tâches
- Système de fichiers
- Périphériques
- Partitionnement des disques
- Mémoire
- Génération d'un noyau
- Administration des packages
- Observation des activités du système

Administration d'une station de travail



Installation et configuration d'une station de travail

Quatre grandes étapes :

- Partitionnement de l'espace et installation du système de fichiers
- Chargement du système sur le disque
- Redémarrage du système
- Configuration du système par l'administrateur

Administration des utilisateurs

- Chaque utilisateur doit être défini sur la machine pour pouvoir l'utiliser
- Création d'un compte utilisateur par un administrateur (le super-utilisateur)
- Caractéristiques du super-utilisateur :
  - `root`
  - UID=0
  - tous les droits lui sont attribués
  - shell privilégié avec `su`

Caractéristiques d'un utilisateur (1)

- Identifiant : règles d'attribution dépendant de l'administrateur (nom de l'utilisateur, initiales, etc.)
- Mot de passe (8/10 caractères en général) : séquence complexe ne se trouvant dans aucun dictionnaire
- UID (valeur entière entre 0 et 65535 – SVR4 $2^{32} - 1$) : identification unique de l'utilisateur.
  - 0 à 999 : Comptes systèmes (`bin`, `daemon`, etc)
  - à partir de 1000 : Utilisateurs

Caractéristiques d'un utilisateur (2)

- GID primaire (entre 0 et 65535, >= 1000 en général) : groupe de l'utilisateur à la connexion
- Commentaires
- Répertoire de connexion
- Programme exécuté au login : un shell se trouvant sur la partition /

Gestion des comptes

- Gestion spécifique des utilisateurs au niveau de la sécurité
- Récapitulation de l'ensemble de ces informations : `/etc/passwd`
- Mot de passe pouvant être déporté dans `/etc/shadow` pour System V (uniquement lisible par le `root`)
- Opérations d'administration : Création, Suppression, Modification des propriétés
- Ne pas supprimer les comptes systèmes (`bin`, `daemon`, `sys`, `root`...)
- Éviter les modifications sans vérifier leurs incidences

Création d'un compte utilisateur

- Récupération et détermination des informations nécessaires à leur création
- Répertoire de création : `/export/home/`*login*, `/home/`*login*, etc
- Commande de création : `useradd`, `adduser` $\longrightarrow$ Ajout d'une entrée dans le fichier `/etc/passwd`
- Environnement initial : recopie des fichiers se trouvant dans `/etc/skel` :
  - famille `sh` : lecture des fichiers `/etc/profile`, `/.profile`
  - famille `csh` : lecture des fichiers `/etc/csh.login`, `/etc/csh.cshrc`, `/etc/.login`, `/etc/.cshrc`, `/etc/.login`

Suppression d'un utilisateur

Suppression :

- des fichiers utilisateurs (répertoire de connexion) et sauvegarde (sur bande, par exemple)
- de la boîte aux lettres et sauvegarde
- des alias courrier
- des tâches d'impression et quotidienne (`cron`, `at`)
- de l'entrée dans `/etc/passwd`
- du *username* des groupes dans lesquels il apparaît

Modification des propriétés

Modification des propriétés répertoriées dans `/etc/passwd`

- Modification du nom complet : `chfn`
- Modification du shell de login : `chsh`
- Modification du mot de passe : `passwd`

Gestion des groupes

- Attribution à chaque utilisateur d'un groupe primaire
- Possibilité de partage de répertoire
- Informations sur les groupes dans `/etc/group` :
  - Nom du groupe
  - Mot de passe (optionnel)
  - GID (entre 0 et 65535)
  - Membres du groupe

A priori, pas de modification des groupes créés lors de l'installation du système (`bin`, `sys`, `daemon`...)

Démarrage d'une station de travail (1)

Trois états possibles pour une machine UNIX :

- Le système d'exploitation n'est pas actif (`telinit 6` ou `telinit 0`) :
  Après la phase d'arrêt ou d'allumage (Machine en mode EEPROM ou sur le Bios)
  Pas de processus lancé (possibilité de redémarrer le système)
  Possibilité de test et de réglage

Démarrage d'une station de travail (2)

- Mode *Single-user* ou maintenance (`telinit 1`) :
  Le système est chargé et partiellement initialisé
  Seul le `root` peut intervenir
  Pas d'autorisation de connexion pour les utilisateurs
- Mode multi-utilisateurs (`telinit 3` ou `telinit 5`) : Initialisation totale du système
  Tous les processus nécessaires sont lancés
  Autorisation de connexion pour tous les utilisateurs

Procédure de démarrage

1. Chargement des programmes de boot (chargeur primaire)
2. Initialisation du noyau (chargeur secondaire) : tests matériel
3. Démarrage du processus `init` : exécution de différentes tâches et passage dans un mode ou un *run-level*
   - BSD : 3 modes (*single-user*, *multi-user*, *poweroff*)
   - System V : une dizaine de modes

Mode *single-user*

Mode commun à tous les UNIX
Passage dans un mode de type maintenance (*logiciel*)
Outils de base pour administrer un système défectueux
Pas d'initialisation des services réseaux

Mode multi-utilisateurs

Lancement de tous les services locaux et réseaux
Suivant les UNIX, possibilité de 2 niveaux multi-utilisateurs :

- Pas de possibilité d'être serveur
- Ensemble de fonctionnalités

Configuration de la procédure de boot

- BSD : utilisation de scripts `rc.*` situés dans `/etc/rc.d`
  - Chaque script est composé de sections lançant les services
  - Consultation des scripts de manière statique par `init`
- System V : utilisation de *run-level* à partir du fichier de configuration `/etc/inittab`, `/etc/init/rc-sysinit.conf`
  - Ensemble des scripts regroupés dans le répertoire `/etc/init.d`
  - Lancement des scripts référencés dans l'arborescence `/etc/rc`*N*`.d` où *N* représente le *run-level*

Arrêt et redémarrage du système

- Arrêt des systèmes UNIX à l'aide des commandes `halt`, `shutdown` ou `reboot`
- Passage en mode 0 ou 6 (`telinit`)
  - Vidage des tampons et écriture sur le disque
  - Démontage des systèmes de fichiers
  - Terminaison des processus

  ou lecture des scripts se trouvant dans `/etc/rc[06].d`

Exemple de script de démarrage de service

```
#!/bin/bash
#
# syslog        Starts syslogd/klogd.
#
#
# chkconfig: 2345 12 88
# description: Syslog is the facility by which many daemons use to log \
# messages to various system log files.  It is a good idea to always \
# run syslog.
### BEGIN INIT INFO
# Provides: $syslog
### END INIT INFO

# Source function library.
```

```
. /etc/init.d/functions

[ -f /sbin/syslogd ] || exit 0
[ -f /sbin/klogd ] || exit 0

# Source config
if [ -f /etc/sysconfig/syslog ] ; then
        . /etc/sysconfig/syslog
else
        SYSLOGD_OPTIONS="-m 0"
        KLOGD_OPTIONS="-2"
fi

RETVAL=0

umask 077
```

```
start() {
        echo -n $"Starting system logger: "
        daemon syslogd $SYSLOGD_OPTIONS
        RETVAL=$?
        echo
        echo -n $"Starting kernel logger: "
        daemon klogd $KLOGD_OPTIONS
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/syslog
        return $RETVAL
}
```

```
stop() {
        echo -n $"Shutting down kernel logger: "
        killproc klogd
        echo
        echo -n $"Shutting down system logger: "
        killproc syslogd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/syslog
        return $RETVAL
}
```

```
status() {
        status syslogd
        status klogd
}
restart() {
        stop
        start
}

case "$1" in
start)
start
        ;;
stop)
stop
..
```

```
status)
rhstatus
;;
restart|reload)
restart
;;
condrestart)
[ -f /var/lock/subsys/syslog ] && restart || :
;;
*)
echo $"Usage: $0 {start|stop|status|restart|condrestart}"
exit 1
esac
exit $?
```

Lancement automatique de processus

Tâches à effectuer régulièrement : sauvegarde, mise à jour de base de données système, observation régulière du système...

Lecture par le démon `cron` du fichier `crontab` :

- System V et BSD récents : 3 types de files d'attente :
  - Exécution différée à une date et une heure précise : `at`
  - Exécution différée dans une file d'attente : `batch`
  - Exécution cyclique : définie dans le fichier `crontab`
- BSD anciens :
  - Exécution cyclique : définie dans le fichier `crontab`
  - Exécution différée à une date et une heure précise : `at`

Exécution différée avec at (1)

```
at -m HEURE
```

- Exécution à une heure précise d'une commande
- La commande envoie un mail à l'utilisateur
- Plusieurs options sont disponibles

Exécution différée avec at (2)

Format de l'heure :

- `HHMM`
- `HH:MM`
- `midnight` (0h)
- `noon` (12h)
- `teatime` (16h)
- `MMJJAA`
- `MM/JJ/AA`
- `JJ.MM.AA`
- `now + X minutes|hours|days|weeks`

Exécution différée avec at (3)

Exécution à une heure précise d'une commande lue sur l'entrée standard :

```
$ at 11:36
warning: commands will be executed using (in order)
                  a) $SHELL b) login shell c) /bin/sh
at> scriptSauvegarde.sh
at> <EOT>
job 1 at 2012-01-10 11:36
#--------------------------
$ at now + 1 hour
warning: commands will be executed using (in order)
                  a) $SHELL b) login shell c) /bin/sh
at> scriptSauvegarde.sh
at> <EOT>
job 3 at 2012-01-10 12:38
```

Exécution différée avec at (4)

```
$ at 11am tomorrow
warning: commands will be executed using (in order)
                  a) $SHELL b) login shell c) /bin/sh
at> scriptSauvegarde.sh
at> <EOT>
job 4 at 2012-01-11 11:00
```

Lecture de la commande dans un fichier avec l'option `-f` :

```
$ at -f scriptSauvegarde.sh 11:36
```

Exécution différée avec at (5)

Visualisation des commandes en exécution différée avec l'option `-l` ou la commande `atq` :

```
$ at -l
3       2012-01-10 12:38 a monnin
4       2012-01-11 11:00 a monnin
$ atq
3       2012-01-10 12:38 a monnin
4       2012-01-11 11:00 a monnin
```

Exécution différée avec at (6)

Suppression d'une commande avec l'option `-d` ou la commande `artrm`

```
$ at -d 3
$ at -l
4       2012-01-11 11:00 a monnin
$ atrm 4
$ at -l
$
```

Exécution cyclique avec `cron` (1)

Opération effectuées toutes les minutes par le programme `cron` :

- Examen du répertoire `/var/spool/cron/crontabs`
- Exécution des commandes placé dans les fichiers du répertoire.
  Le nom des fichiers correspond à des utilisateurs locaux ou déclarés sur le réseau
- Envoi d'un mail à l'utilisateur après exécution

Exécution cyclique avec cron (2)

Installation des commandes à exécuter cycliquement à l'aide de la commande `crontab` :

```
crontab Nom_de_fichier
```

Exemple :

```
$ crontab crontab_test
```

Visualisation de la liste des commandes à exécuter cycliquement :

```
crontab -l [-u nom_de_l'utilisateur]
```

Exemple :

```
crontab -l
```

Exécution cyclique avec cron (3)

Suppression de la liste des commandes à exécuter cycliquement :

```
crontab -r [-u nom_de_l'utilisateur}
```

Exemple :

```
crontab -r
```

Exécution cyclique avec `cron` (4)

Format du fichier `crontab` :

- Les commentaires sont marqués à l'aide du caractère #
- Chaque ligne (non commentée) correspond à une commande à exécuter
- Chaque ligne est composée de 6 colonnes
  - Les cinq premières définissent la date et l'heure d'exécution
  - La sixième colonne contient la commande à exécuter

Exécution cyclique avec cron (5)

- La date et l'heure sont définies de la manière suivante :
  - colonne 1 minute (0-59)
  - colonne 2 heure (0-23)
  - colonne 3 jour du mois (1-31)
  - colonne 4 mois de l'année (1-12)
  - colonne 5 jour de la semaine (0-6, 0 étant dimanche)

Chaque colonne peut contenir le caractère * (n'importe quelle valeur) ou une liste de valeurs séparées par des virgules

Exécution cyclique avec cron (6)

Exemple :

```
0 0 * * * find / -name core -print > /root/diskPicture.lst
5,20,35,50 * * * * /root/script-verif.sh
0 0 1 1 * /root/envoyerMailBonneAnnee.sh
0 19 * * 5 /root/envoyerMailBonWeekEnd.sh
```

Exécution cyclique avec cron (7)

Installation, affichage, suppression :

```
$ crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (crontab_test installed on Thu Jan 29 16:28:28 2011)
0 0 * * * find / -name core -print > /root/diskPicture.lst
5,20,35,50 * * * * /root/script-verif.sh
0 0 1 1 * /root/envoyerMailBonneAnnee.sh
0 19 * * 5 /root/envoyerMailBonWeekEnd.sh
$ crontab -r
$ crontab -l
no crontab for monnin
```

Structure du système

- Tout est fichier
- Arborescence de fichiers unique
- Les fichiers ne sont pas typés
- Montage : intégrer les partitions dans l'arborescence
  Permet d'affecter tout système extérieur (disquette, cdrom, rép. réseau...) à un répertoire créé pour cela dans l'arborescence
- 6 catégories de fichiers
  - normaux
  - répertoires
  - périphériques
  - liens
  - pipes
  - sockets

Organisation des répertoires du système d'exploitation

- / : racine de l'arborescence
- /boot : Noyau et configuration du noyau
- /dev : Périphériques
- /lost+found : Répertoire contenant les blocs et fichiers "perdus"
- /etc : Fichiers de configuration
- /lib : Librairies nécessaires au fonctionnement minimal du système (*single user*)

Organisation des répertoires du système d'exploitation (1)

- `/bin` : Exécutables nécessaires au fonctionnement minimal du système (*single user*) et complètement exploitables par un utilisateur
- `/sbin` : Exécutables système nécessaires au fonctionnement minimal du système (*single user*), avec accès ou utilisation restreint pour un utilisateur
- `/home` : Répertoires de connexion des utilisateurs
- `/root` : Répertoire de connexion du `root` (super utilisateur)

Organisation des répertoires du système d'exploitation (2)

- `/proc` : Répertoire contenant les processus s'exécutant sur le système et leur description
- `/sys` : Répertoire contenant les processus s'exécutant
- `/tmp` : Répertoire des fichiers temporaires sur le système et leur description (version évoluée)
- `/usr` : Répertoire contenant les applications supplémentaires généralement liées à des paquetages (contient les répertoires `bin`, `sbin`,`lib`, `share`)
- `/mnt` : Points de montage

Organisation des répertoires du système d'exploitation (3)

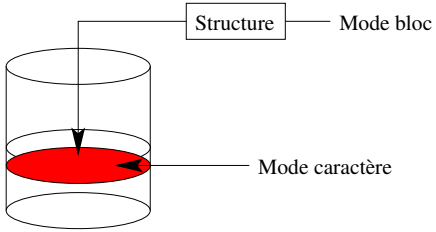
- `/var` : Répertoires dont le contenu varie pendant la session courante du système ou contenant des données sensibles (`/var/cache`, `/var/lib`), `/var/lock`, `/var/log`, `/var/mail`, `/var/run`, `spool`, `/var/tmp`, `/var/www`)
- `/opt` : Utilitaires et Applications supplémentaires quelconques
- `/usr/local` : Répertoire contenant les applications supplémentaires quelconques (contient les répertoires `bin`, `sbin`, `lib`, `share`)

Périphériques (1)

Représentation des périphériques à travers des fichiers spéciaux (*Device Drivers*) : blocs ou caractères

Accès aux périphériques :

- Mode **b**loc : indirect, à travers les structures (partitions)
- Mode **c**aractère (raw) : direct



Périphériques (2)

- Ajout de périphériques : création de nouveaux fichiers spéciaux
- Certains UNIX offrent des mécanismes d'auto-configuration des périphériques (Solaris)
- Sous Linux, les fichiers spéciaux sont :
  - soit déjà créés
  - soit créés dynamiquement
- Sous UNIX, tout est fichier

Nomenclatures des périphériques

Regroupement des périphériques différents suivant le type d'UNIX :

- `/dev/sda1` : disque en mode bloc
- `/dev/sg` ou `/dev/rsda1` : disque en mode caractère
- `/dev/tty0` : terminal asynchrone
- `/dev/disk` : (*sous debian) accès aux disques par identifiant (indépendant du matériel), chemin, label, identifiant unique universel (UUID - lié au matériel)

Pourquoi partitionner ?

Découpage du disque physique en disques virtuels :

- cohabitation de plusieurs systèmes de fichiers
- isolement de certaines parties du système (`/usr`, `/var`, `/home`)
- facilité de réalisation de certaines tâches (sauvegarde de données, consultation en lecture seulement...)
- exportation de partitions vers d'autres machines

$\longrightarrow$ Changement de taille d'une partition :
sauvegarde préalable des données

Identifier les besoins

$\longrightarrow$ Étape importante qui conditionne le bon fonctionnement du système

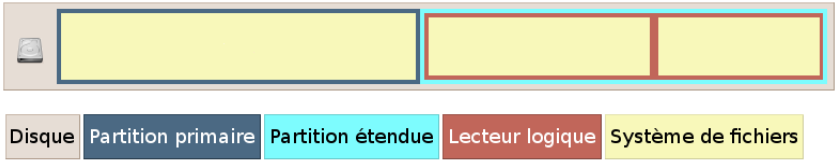
- Avoir une idée précise de l'utilisation future de la machine (serveur de fichier, station de travail, serveur de messagerie, machine de calcul)
- Évaluer :
  - la taille du système
  - les besoins en mémoire virtuelle (zone temporaire, swap)
  - l'espace disque alloué aux utilisateurs

Partitionnement

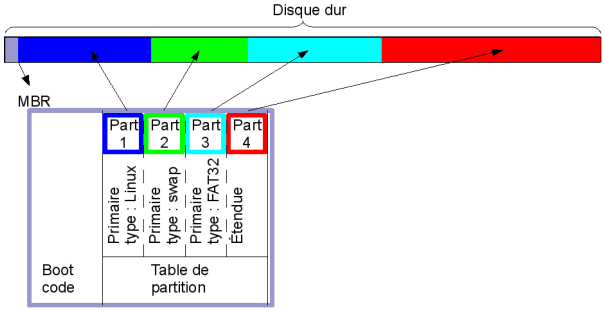
- Précautions :
  - Manuellement : Déterminer le bloc de départ et le bloc d'arrivée
  - Éviter les recouvrements de partition (risque d'erreur très important)
- Outils :
  - `fdisk`, `cfdisk`, `parted`
  - `gparted`

Partitions (type MBR (Master Boot Record))

→ Organiser les données

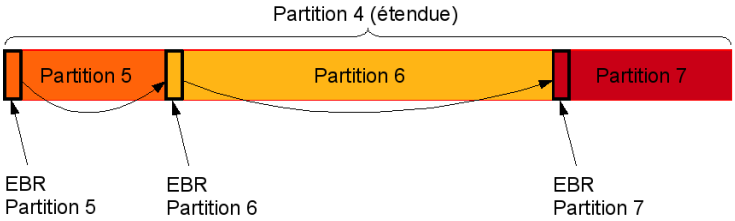


Partition primaire



- Contient au max 4 partitions :
  1 à 3 partitions principales puis une partition étendue
- Reconnue par le bios
- limitation à 2 To par partition

Partition étendue (logique)



- Contient des partitions secondaires
- Contenue dans l'Extended Boot Record (EBR)

Partitions (type GPT)

- Standard GPT : GUID Partition Table
- Nouvelle organisation des partitions
- Utilisation du système EFI (remplaçant du BIOS)
  - plus de limitation à 4 partitions
  - plus de partition étendue
  - limitation à 9,4 Zo par partition

$\rightarrow$ Système jeune

Gestion de volumes logiques

- RAID (Redundant Array of Independent Disks)
  - RAID 0 : Stripping (entrelacement de disques)
  - RAID 1 : Mirroring (miroir de disque)
  - RAID 1+0 (RAID 10) : Stripping et Mirroring
  - RAID 5 : Stripping sur disques indépendants avec parité répartie
  - RAID 6 : évolution du RAID 5, avec n informations redondantes ($n \geq 2$)
- LVM (Logical Volume Management)
  - resize partition
  - snapshot

(Voir description plus tard dans le cours)

Mémoire virtuelle

(*zone de swap*)

- Utilisation de l'espace disque comme mémoire virtuelle en supplément de la mémoire centrale
- Différentes manière de réserver la zone de swap :
  - Partition dédiée
  - Fichier local
  - Système de fichiers
  - Fichier distant

Gestion de la mémoire virtuelle

- Manipulation au niveau du noyau
- Sous Linux, la zone de swap n'est pas indispensable mais vivement conseillée (en principe, 2 fois la taille de la mémoire centrale)
- Opérations :
  - Détermination de la taille : `/sbin/swapon -s` – DEC OSF1, `/etc/swapinfo` – HP-UX, `/bin/free` – Linux, `/usr/sbin/swap -l` – Solaris
  - Ajout : `/etc/swapon`
  - Destruction : Impossible sur la plupart des systèmes, peu conseillée sur Linux, `/usr/sbin/swap -a` sous Solaris

Génération d'un système (1)

Le noyau

- Cœur du système : accès au matériel, à la mémoire, aux systèmes de fichiers
- La plupart des systèmes sont fournis avec un noyau *générique*
- Chargement du noyau en mémoire après exécution du code primaire (`bootstrap`)
- Puis, reconnaissance du matériel et chargement des pilotes

Génération d'un système (2)

Le noyau

- contient les éléments de base du système et permet le chargement de modules :
  - les pilotes
  - les gestionnaires réseaux
  - ...
- Contenu dépendant des constructeurs

Types de noyau

- Noyau statique :
  - Génération du noyau avec tous les modules nécessaires à un moment donné
  - Chargement intégral au démarrage
  - La modification de la configuration du noyau implique un redémarrage de la machine
- Noyau dynamique :
  - Noyau minimal pouvant charger des modules dynamiquement
  - Pas de redémarrage nécessaire (sauf pour certaines modifications de certains paramètres)

Chargement/déchargement de modules sous Linux (1)

- `/sbin/modprobe` : Manipulation des modules chargeables du noyau
  - Chargement d'un module (et des modules en dépendant) : `/sbin/modprobe usb-uhci`
  - Déchargement d'un module (et des modules en dépendant) : `/sbin/modprobe -r usb-uhci`

  Fichier de configuration : `/etc/modules.conf`, `/etc/modprobe.d` définition d'alias (`alias usb-controller usb-uhci`)

  `/sbin/rmmod usb-uhci`

Chargement/déchargement de modules sous linux (2)

- `/sbin/insmod` : installation d'un module chargeable du noyau, passé en argument
  `/sbin/insmod usb-uhci`
- `/sbin/rmmod` : déchargement d'un module du noyau, passé en argument
  `/sbin/rmmod usb-uhci`

Génération d'un nouveau noyau

- Télécharger les sources du noyau utilisé, les dépendances nécessaires à la compilation
- Déterminer les modules nécessaires
- Conserver une version de sauvegarde du noyau actuel
- Description du noyau à générer dans un fichier de configuration (`/usr/sys/conf` – BSD `/etc/system` – System V)
- Compilation du noyau
- Remplacement de l'ancien noyau

Génération d'un nouveau noyau Linux (1)

- Nettoyage des fichiers objets :
  `make clean ; make mrproper`
- Configuration du noyau (et des modules chargeables) :
  `cd /usr/linux/src ; make xconfig` (puis choix des paramètres)
- Génération du noyau :
  `make bzImage` ou `make vmlinuz`
- Génération des modules :
  `make modules`

Génération d'un nouveau noyau Linux (2)

- Installation des modules :
  `make modules_install`
- Installation du nouveau noyau :
  Copie du noyau (situé dans `/usr/src/linux/arch/i386/boot`) dans `/boot` sous un autre nom spécifique
  NB : Éviter les remplacements, toujours conserver le noyau précédent qui fonctionne

Génération d'un nouveau noyau Linux (4)

En pratique

→ Suivre les recommandations de sa distribution

Administration des packages

Distribution de logiciels sous forme de packages :

- formats propres aux systèmes d'exploitation

Installation automatique des packages lors de l'installation du système
Visualisation, ajout, suppression des packages installés :

- Solaris : `pkginfo`, `pkgadd`, `pkgrm`, `pkgchk`, `admintool`
- Linux : `linuxconf`, `pkgtool`, `Yasp`, `rpm`, `autorpm`, `urpmi`, `apt-get`, `apt-cache`, `dpkg`, `aptitude`

Gestion des logiciels

- Logiciels non standard, non livrés avec le système ou nouvelles versions
- Installation variable : Recopie de fichiers binaires, compilation des sources (utilisation d'`autoconf`, `automake`...)

Exemple d'installation de logiciel (debian)

- A partir d'une archive locale : `dpkg -i nom de l'archive`

  ```
  # dpkg -i zip_2.31-3_i386.deb
  ```

- A partir d'une archive présente dans les dépôts : `apt-get install nom de l'archive`

  ```
  # apt-get install zip
  ```

- Mise à jour des paquetages :

  ```
  # apt-get update
  # apt-get upgrade zip
  ```

- Recherche de paquetage :

  ```
  # apt-cache search zip
  ```

Exemple d'installation à partir des sources

- Exemple : `httpd-2.0.52.tar.gz`

```
tar xzvf httpd-2.0.52.tar.gz
(lecture du fichier INSTALL)
$ ./configure --prefix=/usr/local
$ make
$ make install
$ /usr/local/bin/apachectl start
```

Messages du système (1)

Consignation des opérations et problèmes rencontrés par le système :

- Messages des logiciels : le démon `syslogd` scrute le fichier `/dev/klog`, `/dev/log` et le port 514 (machines distantes)
- Journal : `/var/log/syslog` ou `/var/log/syslog`, et d'autres
- Fichier de configuration : `/etc/syslog.conf`
  - Définition de règles de journalisation : aiguillage des messages systèmes dans différents fichiers (`/var/log/user`, `/var/log/kernel/errors`, `/var/log/mail/info`, etc.)

Messages du système (2)

- Messages d'erreurs matériel :
  - BSD : commande `dmesg`
    Messages rangés dans `/usr/adm/messages`
  - System V : commande `dmesg`
    Messages rangés dans les fichiers présents dans `/var/log/*` (suivant les règles de journalisation de syslog, et les outils système)

Observation des activités du système

Tâche importante pour la sécurité et le confort des utilisateurs (éviter la surcharge du système) :

- Lister les utilisateurs (`who`, `finger`)
- Lister les processus du système (`ps`)
- Gestion des processus (`nice`, `kill`)
- Statistiques d'utilisation des ressources (`statcmd`, `vmstat`, `iostat` – BSD, `sar` – system V)
- Utilisation des disques (`du`, `df`)

Répertoires /proc et /sys

Origine : systèmes Unix, années 80

/proc :

- Présentation des différentes facettes des processus à un moment donné
- Regroupes des informations relatives aux processus dans des répertoires numériques (PIDs)
- Sous Linux : vue du système d'exploitation et du matériel
  Possibilité de paramétrage dynamique

/sys :

- Évolution de /proc
- Complète et réorganise /proc

Présentation générale de /proc (1)

- Contient des informations clés sur l'état du noyau et du système en général
- Varie en fonction des systèmes
- Arborescence d'objets
- Possibilités d'effectuer les opérations classiques `ls`, `cat`, `cd`...

$\longrightarrow$ Interface de lecture/écriture de variables et structures internes au noyau

Présentation générale de /proc (2)

```
$ ls -l /proc/partitions
-r--r--r-- 1 root root 0 janv.  9 14:02 /proc/partitions
$ cat /proc/partitions
major minor  #blocks  name

3     0   78150744 hda
3     1    5116671 hda1
3     2          1 hda2
3     5    2040223 hda5
3     6    5116671 hda6
3     7   32941251 hda7
3     8   32933218 hda8
```

Contenu de /proc

Quatre types d'objets :

- Répertoires :
  - Nom numérique : représentation des processus
  - Sous-ensemble du système (`scsi`, `sys`...)
- Fichiers réguliers :
  - informations en ASCII
  - exploitable avec des commandes comme `cat`, redirections E/S
- Liens symboliques : `self` et `mounts`, `/proc/PID/exe` (lien vers le binaire)
- Fichiers spéciaux (rares) : correspondent à un périphérique à piloter

Manipulations (1)

Lecture/écriture : appel de fonction ou de méthode associés aux objets implantés en mémoire

```
$ ls -l /proc/cmdline
-r--r--r--  1 root root 0 Jan 23 15:46 cmdline
$ file /proc/cmdline
cmdline: empty
$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-36-generic root=UUID=94e951f3-2cf2-4384-b
```

NB : Pas de possibilité de modifier l'arborescence avec les commandes `mkdir`, `ln`, `touch`...

Manipulations (2)

- Commande `cat` : invocation de l'appel système `read()`
- Suivi de l'évolution du contenu d'un fichier :

```
$ watch cat /proc/loadavg

Every 2.0s: cat /proc/loadavg          Tue Jan 23 15:49:32 2007
0.04 0.05 0.08 2/110 28710
```

Manipulations (3)

- Modification d'objets
  Appel système `write()`

```
$ cat /proc/sys/kernel/threads-max
16378
$ echo 4096 > /proc/sys/kernel/threads-max
$ cat /proc/sys/kernel/threads-max
4096
```

Manipulations (4)

- Chargement/déchargement de modules

```
# ls -l /proc/sys
total 0
dr-xr-xr-x  2 root root 0 Jan 23 16:00 debug/
dr-xr-xr-x  7 root root 0 Jan 23 16:00 dev/
dr-xr-xr-x  5 root root 0 Jan 23 08:42 fs/
dr-xr-xr-x  4 root root 0 Jan 23 15:56 kernel/
dr-xr-xr-x  8 root root 0 Jan 23 16:00 net/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 proc/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 vm/
# lsmod |grep sunrpc
# modprobe sunrpc
# lsmod |grep sunrpc
sunrpc                122788  0
# ls -l /proc/sys
total 0
dr-xr-xr-x  2 root root 0 Jan 23 16:00 debug/
dr-xr-xr-x  7 root root 0 Jan 23 16:00 dev/
dr-xr-xr-x  5 root root 0 Jan 23 08:42 fs/
dr-xr-xr-x  4 root root 0 Jan 23 15:56 kernel/
dr-xr-xr-x  8 root root 0 Jan 23 16:00 net/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 proc/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 sunrpc/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 vm/
```

Suivi de processus

- Répertoire numérique : pseudo-répertoire correspondant à un processus ou un thread du système
- Contenu :
  - Etat du système : `status`, `stat`, `wchan`, `auxv`
  - organisation de l'espace d'adresses : `statm`, `maps` `mem`
  - contexte d'exécution : `exe`, `cmdline`, `environ`
  - fichiers utilisés : `mounts`, `root`, `cwd`, `fd`
- Informations utilisées par `top` et `ps`

Analyse du système d'exploitation et du matériel (1)

Visualisation de l'utilisation courante des différents composants du systèmes (CPU, mémoire, disque...)

- exploitation du(des) processeur(s) : `/proc/stat`
- état de la mémoire : `/proc/meminfo`
- partitions : `/proc/partitions`
- zone de swap : `/proc/swaps`

Analyse du système d'exploitation et du matériel (1)

Visualisation de l'historique du système

- Paramètres de lancement du noyau : `/proc/cmdline`
- Utilisation de la machine : `/proc/uptime`

Egalement accès à travers des commandes :

- `lspci`, `lsusb`, `lspnp`...
- `uptime`, `free`, `tload`...

Paramétrage dynamique du système (1)

- Possibilité de modifications des fichiers de `/proc/sys` :
  - `echo`, `vi`...
  - `sysctl`
- Modification pour la session en cours
- Ajustement de paramètres de fonctionnement
  - du noyau (`kernel`)
  - de la gestion du système de fichiers (`fs`)
  - de la gestion de la mémoire virtuelle (`vm`)
  - du réseau (`net`)

Paramétrage dynamique du système (2)

Utilisation de `echo` :

```
$ cat /proc/sys/kernel/threads-max
16378
$ echo 4096 > /proc/sys/kernel/threads-max
$ cat /proc/sys/kernel/threads-max
4096
```

Utilisation de `sysctl` :

```
# sysctl kernel.threads-max
kernel.threads-max = 16378

# sysctl  -w kernel.threads-max=4096
# sysctl kernel.threads-max
kernel.threads-max = 4096
```

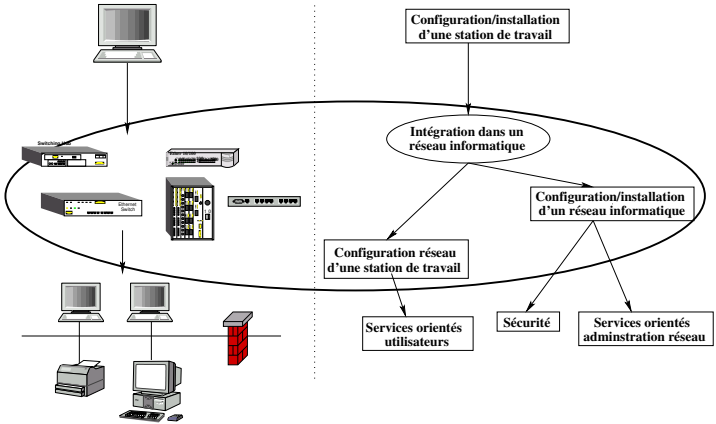
Présentation générale de /sys

- Nouveau système de fichiers orienté vers la description du matériel
- Publication de l'arborescence des composants matériels et des périphériques logiciels
- Similaire à /proc

Contenu de /sys

- `block` : utilisation des périphériques en mode bloc
- `bus` : énumération des différents bus de la machine
- `class` : organisation des périphériques suivant leur fonction
- `devices` : hiérarchisation des composants, identifiés en fonction de leur position dans le bus
- `firmware` : ACPI (gestion de l'énergie), EDD (disques visible par le BIOS)
- `power` : gestion d'énergie

Conception du réseau



Réseau : rappels

Protocoles sur les systèmes UNIX : TCP/IP et UDP/IP

- IP : Interconnexion des réseaux et routage des paquets
  Supporté par plusieurs couches physiques dont Ethernet
- TCP (Transmission Control Protocol) et UDP (User Datagram Protocol) : protocoles de transport en mode connecté/non connecté s'appuyant sur les services de la couche IP

Interface Ethernet/IP

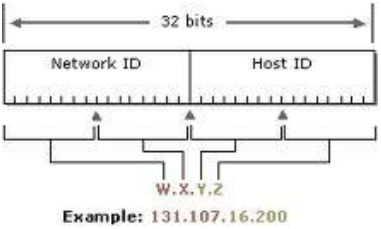
Communication entre deux machines à travers l'interface physique ethernet :

- IP : Protocole de convergence (Ethernet, PPP, ATM, ...)
- Applications : uniquement connaissance des adresses IP
- Établissement d'une correspondance adresse IP / Adresse physique Ethernet (MAC)
- Utilisation des protocoles ARP (Adress Resolution Protocol) et RARP (Revers ARP)
- Interrogation/manipulation du cache ARP/RARP au niveau du système d'exploitation : `arp`, `rarp`

Adresse IPv4

- 4 octets (32 bits) : notation décimale pointée A.B.C.D (par ex. 194.254.167.1)
- Unique au monde :
  - Configuration par logiciel
  - Associée au chaque interface réseau
- Plusieurs classes : A, B, C, D, E
  Attribution :
  - le RIPE (Réseau IP Européen)

Adressage IPv4



- *Adresse réseau* (***network id***) :
  - correspond à la classe
  - détermine le réseau de la machine
  - assigné par une autorité nationale ou internationale
- *Adresse du host* (***host id***) :
  - correspond au masque réseau
  - détermine la machine sur le réseau
  - assignée par l'administrateur du réseau

Classes d'adressage (1)

- Classe A : 1 octet $\rightarrow$ réseau, 3 octets $\rightarrow$ machine
  - $2^{24} - 2 >$ 16 millions de postes
  - Premier octet compris entre 0 et 127
  - Premier bit toujours à 0
  - Ex : 48.27.49.13

- Classe B : 2 octet $\rightarrow$ réseau, 2 octets $\rightarrow$ machine
  - $2^{16} - 2 >$ 65534 postes
  - Premier octet compris entre 128 et 191
  - Deux premiers bits toujours égaux à 10
  - Ex : 131.16.1.23

Classes d'adressage (2)

- Classe C : 3 octet $\rightarrow$ réseau, 1 octets $\rightarrow$ machine
  - $2^8 - 2 > 254$ postes
  - Premier octet compris entre 192 et 223
  - Trois premiers bits toujours égaux à 110
  - Ex : 194.254.167.1

- Classe D : multicast
  - Premier octet compris entre 224 et 239
  - Quatre premiers bits toujours égaux à 1110
- Classe E : réservé IANA, adresses comprises entre 240.0.0.0 et 255.255.255.255

Adresses particulières

- 127.0.0.1 : loopback (pseudo-réseau), localhost
  Utilisé pour les tests logiciels et les communications internes inter-processus
- Adresse du réseau : tous les bits de la partie machine à 0
  194.254.167.0 désigne le réseau de classe C 194.254.167
- Adresse de diffusion (broadcast IP) : tous les bits de la partie machine à 1
  - 194.254.167.255 désigne toutes les machines du réseau 194.254.167.0
  - Permet la recherche d'une machine offrant un service, dont l'adresse est inconnue (serveur NIS, actualisation de la table de routage...)

Adresses particulières : les réseaux privés

Dans chaque classe d'adresse suivantes, il existe des plages particulières, dites « privées »

- Classe A : 10.0.0.0 à 10.255.255.255
- Classe B : 172.16.0.0 à 172.31.255.255
- Classe C : 192.168.0.0 à 192.168.255.255

Ces adresses ne sont pas directement utilisables sur Internet (non routées/routables) et ne peuvent donc servir que pour des réseaux locaux.

Masque de réseau et de sous-réseau

Le masque de **réseau** :

- Il s'écrit de la même manière qu'une adresse IP (4 octets en notation décimale pointée)
- Il permet de déterminer le réseau auquel appartient une adresse IP en faisant une opération binaire ET entre l'adresse et le masque.
- Il est déterminé par la classe d'adresse IP utilisée.

Le masque de **sous-réseau** :

- Il permet la subdivision logique d'un réseau de taille plus importante.
- Il est utilisé pour le routage au sein d'un même réseau.

Ces deux usages sont aujourd'hui généralement confondus : les réseaux ne sont que rarement utilisés d'un seul bloc.

Masque : Notation CIDR

La notation CIDR d'un masque permet son écriture sous une forme beaucoup plus courte que sa version décimale pointée. Elle correspond au nombre de bits du masque. Par exemple :

- /24 correspond à 255.255.255.0
  ou 11111111.11111111.11111111.00000000 sous forme binaire
- /16 correspond à 255.255.0.0
  ou 11111111.11111111.00000000.00000000 sous forme binaire
- /8 $\rightarrow$ 255.0.0.0
- ou encore /19 à 255.255.240.0

Familiarisez-vous avec cette notation, elle est régulièrement utilisée...

Administrer un réseau

- Concevoir (préliminaires) :
  - Plan du réseau
  - Mise en œuvre
- Assurer le bon fonctionnement :
  - Surveillance
  - Dépannage
- Offrir des services aux utilisateurs (ressources numériques, messagerie, stockage...)
- Recueillir les informations nécessaires à l'évolution du réseau

Conception du réseau (1)

- Réflexion sur l'utilisation du réseau
- Identifier les contraintes matérielles, financières
  $\longrightarrow$ Influence sur les choix techniques
- Quelle organisation matérielle et humaine ?

Une grande variabilité suivant le site

Conception du réseau (2)

- Plan du réseau
- Topologie et architecture
- Plan d'adressage, de nommage et de routage
- Architecture des services réseaux (Messagerie, Annuaire, DNS, etc...)
- Organisation des ressources humaines

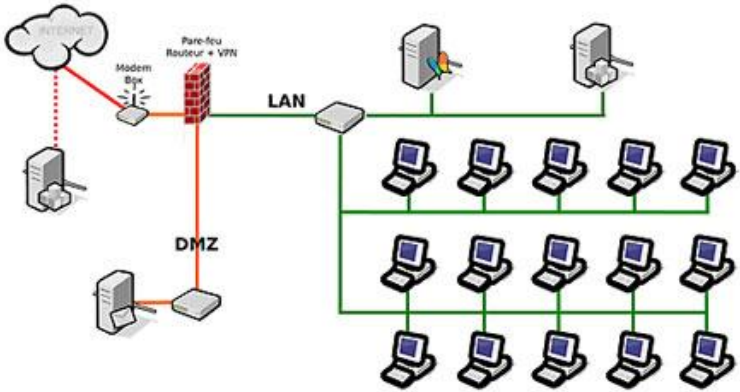
Plan du réseau

Élaboration du plan d'ensemble du réseau :

- Recenser les besoins actuels des utilisateurs
- Évaluer leurs besoins futurs
- Inventorier les ressources existantes
- Concevoir un réseau robuste aux évolutions

$\longrightarrow$ Éviter l'hétérogénéité des équipements

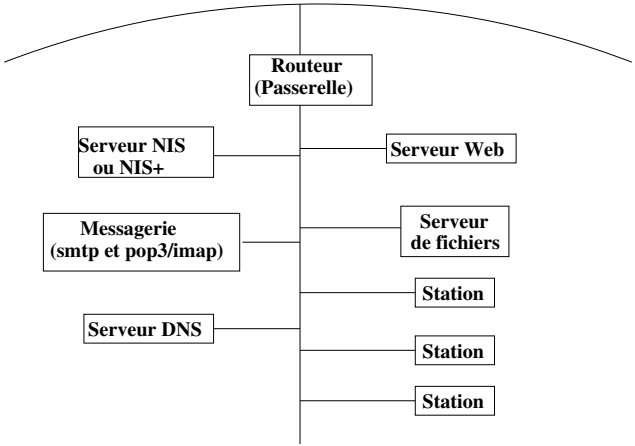
Exemple de plan de réseau



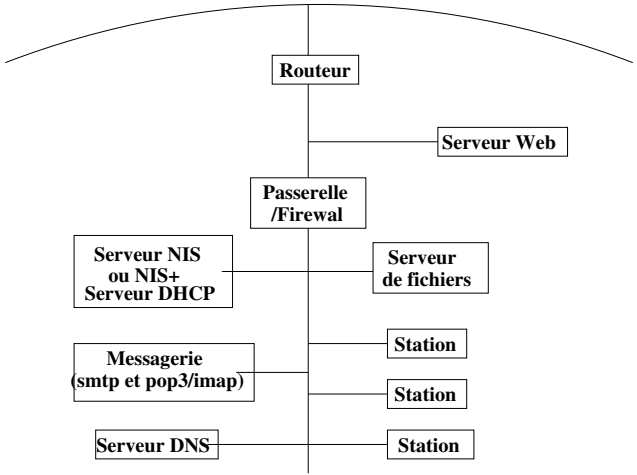
Organisation des ressources humaines

- Recenser les besoins en personnel en fonction de la taille du réseau, de la répartition géographique et des services nécessaires
- Définir la répartition des tâches
- Désigner une instance d'arbitrage
- Informer les utilisateurs sur les personnes à contacter
- Habituer les utilisateurs à contacter la personne compétente pour résoudre leur problème

Exemple de réseau public



Exemple de réseau privé



Plan d'adressage (1)

1) Site isolé : définir le plan d'adressage
   - Découpage des réseaux de classe C en sous réseaux (facilité l'administration)
   - Définition des plages d'adresses par site, par entité, par protocole

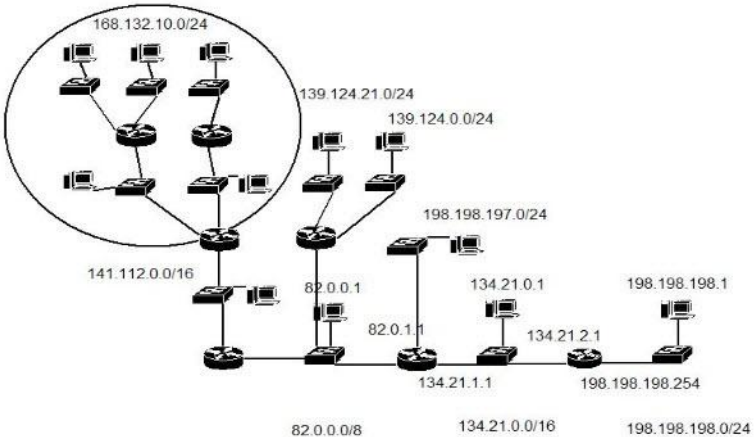
Plan d'adressage (2)

2) Site proche d'un site existant

- intégration dans le site tout en conservant une autonomie :
  - Partage de plages d'adresses
  - Partage des informations et de l'expérience
- Nécessité d'indépendance et d'isolement du site
  Réseau privé plutôt que public :
  - Sécurité
  - Économie d'adresses publiques et routables

Mise en œuvre : Réservation des numéros de réseaux de la classe

Exemple de plan d'adressage



Plan de nommage

- Choix du nom de domaine pour le site ou le groupe de sites
- Concertation avec les administrateurs du réseau dans lequel s'intègre le réseau

$\longrightarrow$ Hiérarchisation du nommage pour les sites importants ou les structures distinctes (création de sous-domaine)

- Mise en œuvre : Réservation du nom de domaine (NIC en France)

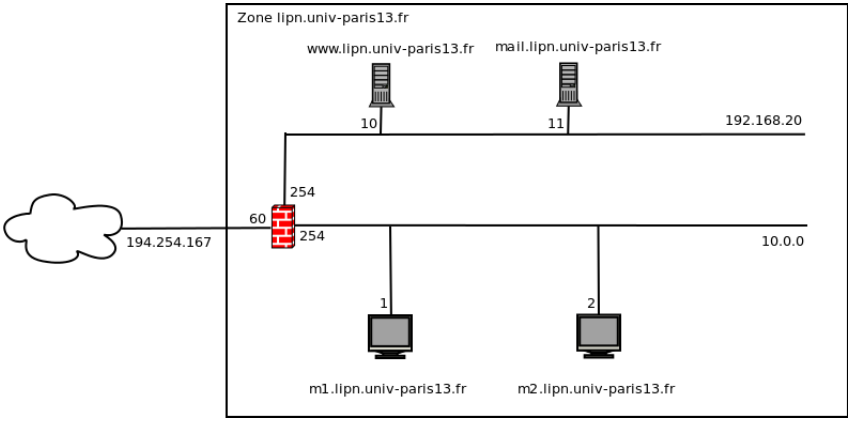
Noms de domaine

- Facilité d'emploi
- Manipulation de noms symboliques plutôt que d'adresses sur 32 bits
- Association adresse IP / nom principal et secondaire (cf. `/etc/hosts`)

$\longrightarrow$ Solution non satisfaisante avec l'explosion des réseaux

- Vers un modèle distribué : à partir de la manière de nommer les machines
- Une machine appartient à un réseau/domaine, un sous-réseau/sous-domaine : son nom en découle
- L'adresse IP d'une machine est connue en interrogeant le DNS du réseau (chaque administrateur gère ses machines sur son DNS)

Exemple : lipn.univ-paris13.fr



Routage

- Transmettre les informations d'un réseau (segment) à un autre

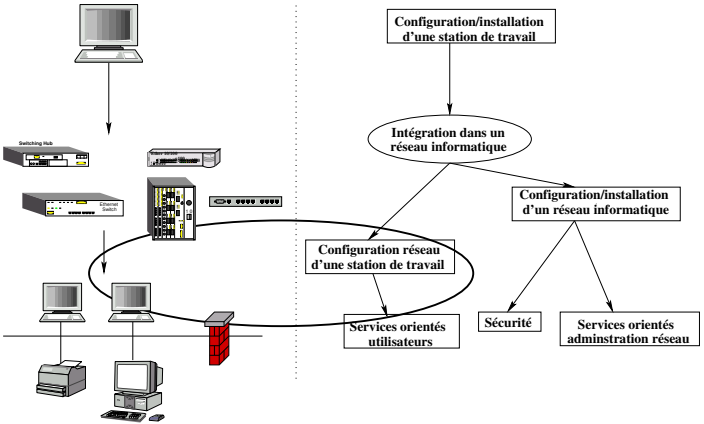
$\longrightarrow$ Trouver un chemin (une route) vers la destination finale à travers des relais (passerelles)

- Table de routage : Tableau contenant les passerelles permettant d'accéder aux réseaux connus
- Types de route :
  - vers une machine simple
  - vers un sous-réseau entier
  - vers une route par défaut (`default`)

Mise en œuvre du plan de routage

- Préparation des configurations des équipements de routage
- Sauvegarde sur les stations de travail

Intégration au réseau



Configuration de l'interface réseau (1)

(Possibilité de configuration du réseau à l'installation)
Dépendant des systèmes d'exploitation, mais en général :

- `/etc/hosts` : Association d'adresse IP et de noms de machine (hostname)

```
127.0.0.1          localhost
194.254.163.3      mail
10.10.0.105        lipn-maple  lipn-maple.lipn.univ-paris13.fr
```

Noms particuliers :

- *localhost* : nom par défaut de la machine
- *mailhost* : utilisé par le logiciel *sendmail*

Configuration de l'interface réseau (2)

- `/etc/networks` : Association d'adresses de réseaux et de noms

```
loopback          127
arpanet           10                arpa     # Historical
ig-edu.univ-paris13.fr              194.254.167
```

  Nom particulier : `loopback`, réseau par défaut

- `/etc/protocols` : Association du numéro de protocole à des noms (udp, icmp, etc.)

```
ip      0       IP      # internet protocol, pseudo protocol number
icmp    1       ICMP    # internet control message protocol
tcp     6       TCP     # transmission control protocol
egp     8       EGP     # exterior gateway protocol
udp     17      UDP     # user datagram protocol
```

Configuration de l'interface réseau (3)

- `/etc/services` : Association de numéro de service à des noms

```
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
```

Configuration de l'interface réseau (4)

- commande `/sbin/ifconfig` : configure et affiche les interfaces réseaux
  `/sbin/ifconfig eth0 <ADRESSE IP> netmask <MASQUE RESEAU> up`
- Exemple :

```
/sbin/ifconfig eth0 192.168.0.12 \
broadcast 192.168.0.255 netmask 255.255.255.0
```

Configuration de l'interface réseau (5)

- `inetd` et `/etc/inetd.conf`
  Contrôle d'un ensemble de démons (`rlogind`, `rshd`, `ftpd`, `telnetd`, etc...)

`/etc/inetd.conf`

```
# Ftp and telnet are standard Internet services.
#
ftp     stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd
telnet  stream  tcp  nowait  root  /usr/sbin/tcpd  in.telnetd
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell   stream  tcp  nowait  root  /usr/sbin/tcpd  in.rshd
login   stream  tcp  nowait  root  /usr/sbin/tcpd  in.rlogind
```

Configuration de l'interface réseau (6)

- Fichier(s) de configuration de l'interface réseau :
  Possibilité d'affectation de valeurs aux variables de configuration (masque réseau, adresse IP...)
  Installation de l'interface à l'aide de la commande `ifconfig`
  Dépendant des (types de) systèmes :
  - Solaris : `/etc/init.d/inetsvc` et `/etc/init.d/rootusr` (liens dans les répertoires `/etc/rc[0-6]/`)
  - Linux : `/etc/init.d/networking`
  - AIX : `/etc/rc.net`

Configuration de l'interface réseau (7)

- Lancement des démons internet :
  - `inetd/xinetd` : gestion des ports et des services
  - `routed` : gestion des routes
  - `rwhod` : gestion des utilisateurs présents sur le réseau local

  Tous ne sont pas obligatoirement lancés pour des raisons de sécurité

Configuration de l'interface réseau (8)

- Équivalence de machines :
  (mécanisme d'autorisation des *r-commandes* :
  `rlogin`, `rsh`, `rcp`, `rdump`, `rrestore`)
  Échec si pas d'autorisation mise en place sauf pour `rlogin`
  Mise en place des autorisations :
  - Déclaration de machines clientes dans `/etc/hosts` sur le serveur
  - Déclaration des autorisations concernant les utilisateurs dans le fichier `/etc/hosts.equiv`

Configuration de l'interface réseau (9)

Mise en place des autorisations (suite) :

- Autorisation de l'environnement de l'utilisateur avec `.rhosts` (dans leur répertoire `HOME`)
- Référencement dans `.rhosts` ou `/etc/hosts.equiv`

Exemples de fichier `.rhosts` :

```
nantes kammoun
bourbaki kammoun
painleve kammoun
```

Contraintes supplémentaires au niveau du propriétaire, des droits et de la nature du fichier

Configuration de l'interface réseau (10)

- fichier /etc/network/interfaces

/etc/network/interfaces

```
auto lo eth0
iface lo inet loopback

iface eth0-home inet static
   address 192.168.1.21
   netmask 255.255.255.0
   gateway        192.168.1.254
```

- Redémarrage du service :

```
# /etc/init.d/networking restart
```

Installation de routes (1)

- Initialisation de la table de routage : `/sbin/ifconfig`
  Création d'une route vers son propre réseau (la machine est sa propre passerelle)

Installation de route (2)

- Ajout de route : commande `route`
  Routage statique :

```
route add host 192.33.182.68

route add net 192.33.182.0 0 gw 192.33.182.68

route add default 192.168.0.0 1
```

  Routage dynamique : démon `routed`, `gated`, etc.
- Suppression de route :

```
route del 192.33.182.0
```

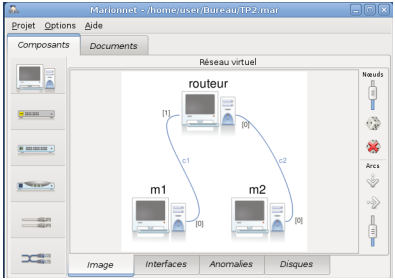
Configuration des routes

Après l'installation des interfaces ethernet
Dépendant du système :

- Solaris : `/etc/init.d/inetinit` avec consultation du fichier `/etc/defaultrouter`
- Linux : `/etc/rc.d/rc.inet1`
- HP-UX : `/etc/netlinkrc`

Création d'une passerelle sous Linux

- Machine Linux avec deux interfaces réseaux
- Activation du forwarding de paquets :
  `echo 1 > /proc/sys/net/ipv4/ip_forward`



- Configuration pour que les passerelles soient le routeur

Visualisation des routes (1)

Commande `traceroute` : Envoi de paquet du protocole ICMP
Récupération des réponses de chaque passerelle
Visualisation des routes définies sur la machine :
`/bin/netstat -r -n`
Linux :

```
$ netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.0.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
0.0.0.0         192.168.0.1     0.0.0.0         UG       40 0          0 eth0
```

Visualisation des routes (2)

Routes vers d'autres réseaux :

```
# netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.33.182.0    192.33.182.68   255.255.255.0   UG        40 0          0 eth0
192.33.182.0    0.0.0.0         255.255.255.0   U         40 0          0 eth0
10.10.0.0       0.0.0.0         255.255.0.0     U         40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         40 0          0 lo
0.0.0.0         192.33.182.254  0.0.0.0         UG        40 0          0 eth0
```

Visualisation des routes (3)

Visualisation des relais (les passerelles) : `traceroute`

```
traceroute to armen.biomath.jussieu.fr (134.157.72.23), 30 hops max, 40 byte
packets
 1  gw5-r.univ-paris13.fr (194.254.170.254)  2,792 ms  3,678 ms  1,718 ms
 2  195.83.240.205 (195.83.240.205)  8,323 ms  5,173 ms  5,505 ms
 3  aubervilliers1.rerif.ft.net (193.48.58.173)  221,376 ms  146,756 ms  131,263 ms
 4  stamand2.rerif.ft.net (193.48.53.189)  137,446 ms  210,720 ms  135,917 ms
 5  peer-renater.rerif.ft.net (193.48.53.217)  117,014 ms  148,342 ms  209,945 ms
 6  nio-n1.cssi.renater.fr (193.51.206.21)  150,090 ms  234,662 ms  151,598 ms
 7  jussieu.cssi.renater.fr (194.214.109.6)  167,079 ms  147,479 ms  185,449 ms
 8  rap-jussieu.cssi.renater.fr (193.51.12.78)  174,031 ms  229,732 ms *
 9  jussieu.rap.prd.fr (195.221.126.33)  229,504 ms  182,648 ms  183,462 ms
10  r-ps.reseau.jussieu.fr (134.157.254.3)  215,697 ms  249,651 ms  206,939 ms
11  r-biomath.chups.jussieu.fr (134.157.192.33)  132,791 ms  225,918 ms  214,403 ms
12  armen.biomath.jussieu.fr (134.157.72.23)  181,951 ms  290,827 ms  158,506 ms
```

Contrôle du réseau (1)

- `/usr/sbin/ping`
  Envoi d'un paquet avec écho à la machine spécifiée et, notification de la réception du paquet (protocole ICMP)
- Exemples :

```
$ ping 192.168.0.12
PING 192.168.0.12: 56 data bytes
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=0. time=2. ms
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=1. time=18. ms
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=2. time=27. ms
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=3. time=9. ms
^C
----192.168.0.12 PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 2/14/27
```

Contrôle du réseau (3)

- `/sbin/ifconfig`
  Option `-a` : liste les interfaces présentes sur la machine

```
eth0      Link encap:Ethernet  HWaddr 00:04:76:94:07:A3
          inet addr:192.168.0.26  Bcast:192.168.0.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6165981 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18422117 errors:0 dropped:0 overruns:0 carrier:2
          collisions:4176156 txqueuelen:100
          RX bytes:2059858794 (1964.4 Mb)  TX bytes:2577606712 (2458.1 Mb)
          Interrupt:11 Base address:0xcc00

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:97723 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97723 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:83338668 (79.4 Mb)  TX bytes:83338668 (79.4 Mb)
```

Contrôle du réseau (4)

- `/usr/bin/netstat`
  Affichage des états des différents composants du réseau sur la machine locale
- Exemples :

```
$ netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR   TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0 6210183      0      0      0 018576108      0      0      0 BMRU
lo    16436   0   98687      0      0      0   98687      0      0      0 LRU
```

Contrôle du réseau (5)

```
$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:login                 *:*                     LISTEN
tcp        0      0 *:shell                 *:*                     LISTEN
tcp        0      0 *:sunrpc                *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:telnet                *:*                     LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  3      [ ]         DGRAM                    103    /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     152    /dev/printer
unix  2      [ ACC ]     STREAM     LISTENING     195    /dev/gpmctl
unix  2      [ ACC ]     STREAM     LISTENING     24775  /tmp/.X11-unix/X0
```

Auscultation d'un réseau Ethernet (1)

Opérations dépendantes du système d'exploitation

DANGEREUX !

Nécessite

- le support du mode *promiscuous* : accès du niveau programmation à tous les paquets Ethernet
  Certains UNIX ne proposent pas ce mode *dangereux* pour la sécurité (possibilité de visualiser tous les paquets passant sur le réseau)
- le passage de l'interface en mode *promiscuous*

Auscultation d'un réseau Ethernet (2)

Capture des paquets (sniffers, bas niveau) : `/usr/sbin/snoop` sur Solaris, `tcpdump` sur Linux
Exemple de paquets capturés :

```
10.10.0.85 -> (broadcast)  ARP C Who is 10.10.0.22, 10.10.0.22 ?
    fwlipn -> lipn         NFS C GETATTR3 FH=0095
      lipn -> fwlipn       NFS R GETATTR3 OK
    fwlipn -> lipn         NFS C ACCESS3 FH=0095 (lookup)
      lipn -> fwlipn       NFS R ACCESS3 OK (lookup)
    fwlipn -> lipn         NFS C LOOKUP3 FH=0095 anass
      lipn -> fwlipn       NFS R LOOKUP3 OK FH=79B1
    fwlipn -> lipn         NFS C GETATTR3 FH=79B1
      lipn -> fwlipn       NFS R GETATTR3 OK
      lipn -> umr7030      TCP D=1134 S=22 Ack=865118141 Seq=1503225955 Len=284 Win=8760
         ? -> (multicast)  ETHER Type=0001 (LLC/802.3), size = 50 bytes
        ls -> BROADCAST    UDP D=161 S=4195 LEN=64
c.up13.fr -> (broadcast)  ARP C Who is 192.33.182.5, 192.33.182.5 ?
cc5.up13.fr -> 192.33.182.173 IP  D=192.33.182.173 S=194.254.164.5 LEN=34, ID=27277
```

Gestion des incidents

Pas de réseau : pourquoi ?
Plusieurs paramètres à évaluer :

- Problème matériel : carte, câble, machine allumée ou éteinte
- Interroger les machines distantes : `/usr/sbin/ping`
  - Interrogation par nom (problème de DNS)
  - Interrogation par adresse IP (problème de route, matériel)

Mise en place de la station dans un réseau

- Configuration de l'interface réseau
- Configuration de la résolution de noms
- Installation de la connexion :
  - vers son serveur de compte utilisateur
  - vers son(ses) serveur(s) de données/stockage
- Installation de la messagerie
- Installation des services annexes