Administration Système

Franck Talbart (franck.talbart@uvsq.fr)

- Version du 14 septembre 2020 -

Organisation

- Emploi du temps
  - S4 à S11 : Cours F. Talbart : 8 x 2h00
  - S13 18 déc. 14h-16h : Examen : 2h00
  - S10 à S12 : T.P. A. Maheo : 6 x 3h00
- Evaluation
  - Au moins un contrôle théorique
    - Questions de cours
    - Exercices pratiques
  - Interrogation surprise
  - Un (ou plusieurs) TP(s) noté(s) au hasard

Bibliographie

— ”Unix System Administration Handbook”, 4th Ed, E. Nemeth, G. Snyder, S. Seebass, T. Hein, Prentice Hall, 2010

— ”Unix Administration”, 3ème Ed, J-M Moreno, Dunod, 2003 (en français)

— ”Unix Administration”, J.-F. Bouchaudy, G. Goubet, Eyrolles, 2007 (en français)

— ”TCP/IP Network Administration”, 2nd Ed, C. Hunt, O’Reilly, 1998

— ”Practical Unix and Internet Security”, 2nd Ed, S. Garfinkel, G. Spafford, 1996

— ”Linux magazine”

— ”The Network Administrator’s Guide”, 2nd Ed., Olaf Kirch et Terry Dawson, 2000, `http://www.tldp.org/guides.html` (The Linux Documentation Project, voir ce site pour d’autres références)

— Slides au format PDF sur http ://franck.talbart.fr

Thèmes abordés

GÉNÉRALITÉS

We all know Linux is great... it does infinite loops in 5 seconds.

Linus Torvalds

Bref historique d'Unix

- Fin des années 60 : Ken Thompson, Bell Labs, système personnel pour PDP-7
  Denis Ritchie, système multi-utilisateur pour PDP-11
- 1975-1980 : division en trois branches
  - Bell Labs : groupe Unix
    Ajout du file system switch, des streams.
    Développent 9 versions puis passent à Plan 9.
  - Berkeley University : Berkeley Software Distribution Ajout de la mémoire virtuelle (4.1), du réseau (sockets, TCP-IP ; 4.2).
  - Branche séparée chez Bell Labs : SYS III, puis SYS V, détenteurs de la marque.
    Ajout de la mémoire partagée, des IPC.
- Nombreuses versions selon constructeurs :
  - Sun : BSD 4.2 + compatibilité et utilitaires SYS V (SunOS), puis SYS V (Solaris) ; "Oracle Solaris" depuis le rachat de Sun en 2010.
  - HP (HP-UX), IBM (AIX), Silicon Graphics (Irix), ...
  - Sur PC : SCO (SYS V commercial) ; FreeBSD, NetBSD ; Minix puis Linux (POSIX).

Systèmes étudiés

Brièvement :

- Unix System V
  - HP-UX
  - Solaris 2
- Unix BSD
  - SunOS (Solaris 1)
  - FreeBSD

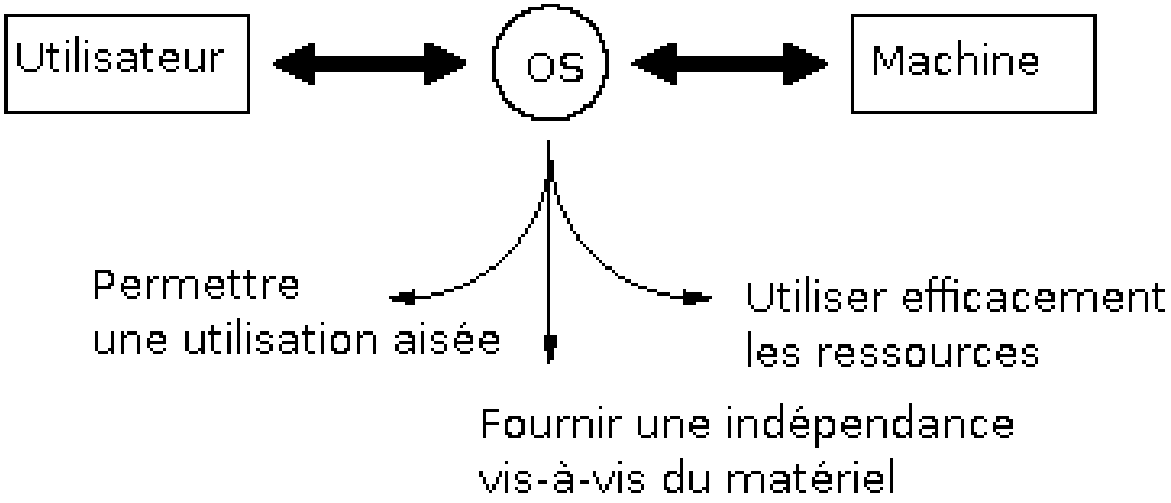
Plus fréquemment :

- Linux

  Différentes distributions :
  - RedHat et ses dérivés (Fedora, RedHat Enterprise, CentOS,
  - Debian et ses dérivés (Ubuntu, Xandros, ...),
  - Suse, ...

Rôle d'un système d'exploitation

- Machine virtuelle
  - abstractions de haut niveau
  - plus facile à manipuler que la machine physique
- Gestion des ressources matérielles (processeur, mémoire, périphériques, ...)
- Partage de ressources
- Contrôles



Notions de base sur Unix

- Multi-tâches multi-utilisateurs
  - Un utilisateur a des droits restreints, `root` a tous les droits.
- Processus : chaque tâche a un environnement, un espace mémoire avec un adressage qui lui sont propres .

  Un séquenceur gère la transition et la priorité entre les processus.
- Mémoire virtuelle
  - L'espace d'adresse d'un processus est indépendant de la mémoire physique.
  - Quand il n'y a plus de "pages" libres on utilise le *swap*.
  - Pages effectivement utilisées $<=$ mém. physique + swap
- Mécanismes simples et puissants
  - en particulier pour la gestion des entrées/sorties et des processus.
- Interface standard : POSIX
- Un grand nombre de logiciels libres disponibles.

Tâches de l'administrateur

- Ajout et suppression d'utilisateurs
- Ajout et suppression de matériel, reconfiguration
- Sauvegardes et restaurations
- Installation de logiciels
- Surveillance du système
  - sécurité
  - monitoring
- Gestion de la documentation locale
- Aide aux utilisateurs

Principes de base de l'administration

- Tout système nécessite un administrateur
- Complexité de l'administration accrue par l'informatique omniprésente :
  - de plus de plus de machines
  - systèmes hétérogènes
  - nomadisme
- Pas de modification du système lui-même
  - modification de fichiers de configuration
  - lancement de services
- Pas de reconfiguration
  - le vendredi soir ou avant de partir en congés
  - après un ”pot”

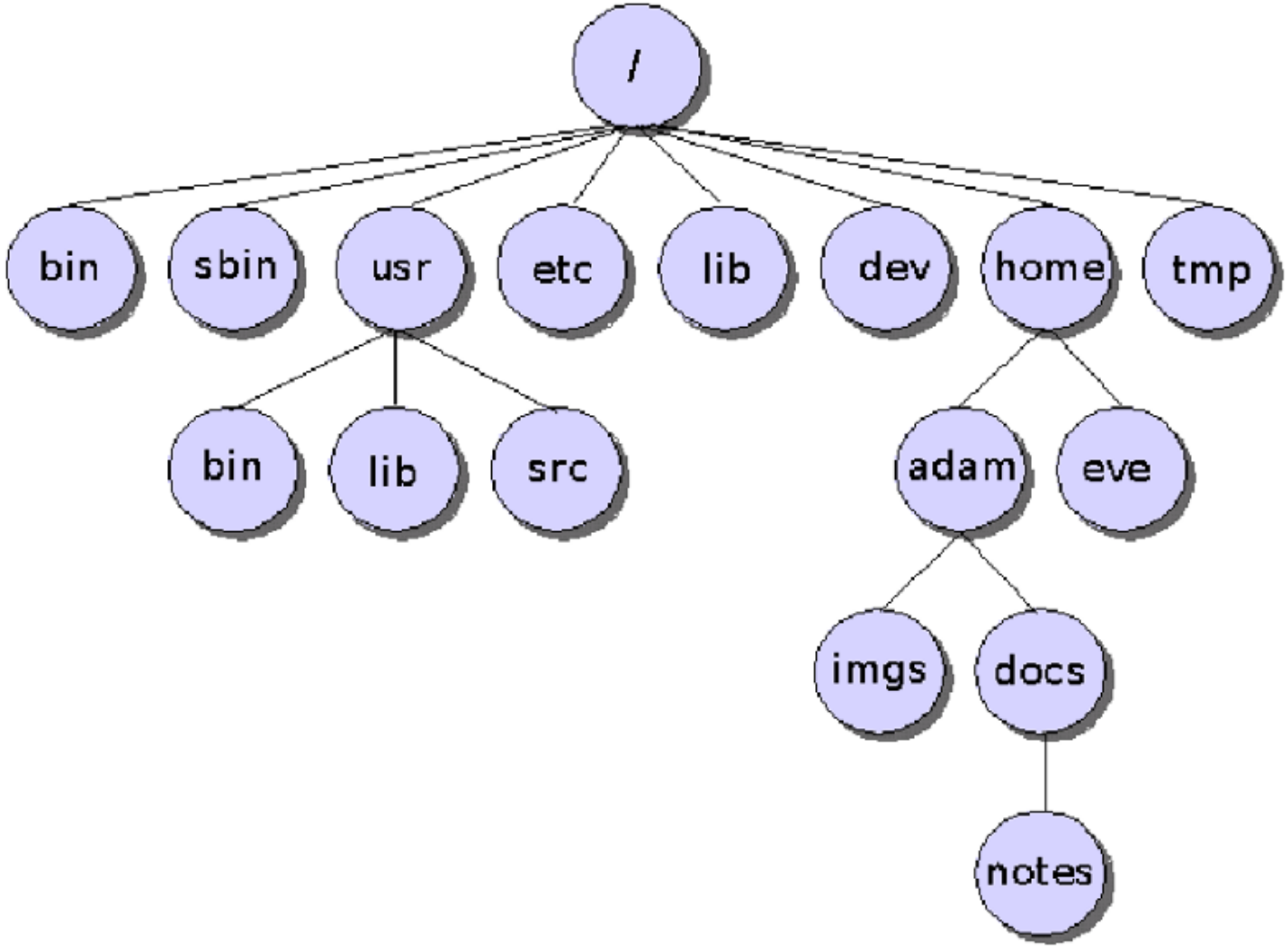
Le compte ’root’

- Unix surveille le comportement de tous les utilisateurs ...
- ... sauf ’root’ qui a tous les droits
- Connexion sous ’root’ :
  - connexion normale (login), déconseillée
  - `su`
  - `sudo`, `calife`, ...

Commandes d'administration

- Les commandes pour les utilisateurs sont contenues dans les répertoires `/bin` et `/usr/bin`
- Il existe des répertoires spécifiques pour les commandes d'administration :
  - `/sbin`
  - `/usr/sbin`
  - `/etc`
  - `/usr/etc`
- Les fichiers de configuration du système sont généralement placés dans le répertoire `/etc`

Arborescence Unix



Arborescence Unix

- Les répertoires classiques sous Unix sont :
  - `/bin`, `/sbin` : binaires et binaires système
  - `/lib` : les librairies utilisées par les binaires
  - `/usr/include` : les entêtes décrivant la librairie C
  - `/usr/share` : resources indépendantes de l'architecture
  - `/etc` : fichiers système
  - `/man` (ou `share/man`) : le manuel
  - `/var` : les données "variables"
  - `/tmp` : les fichiers (en principe) temporaires
- Ces répertoires peuvent se retrouver à plusieurs niveaux du système de fichier :
  - `/` : racine du système, tout ce qui y est directement lié
  - `/usr` : tout ce qui est lié à l'utilisation du système par les utilisateurs
  - `/usr/local` : tout ce qui "local" à un site
  - `/usr/local/samba` : tout ce qui est lié à un produit logiciel donné

Arborescence Unix

- Plusieurs versions du Filesystem Hierarchy Standard (FHS) : actuelle : 3.0
  - Toutes les distributions ne respectent pas strictement le standard
  - `/run` : nouveau répertoire qui centralise les fichiers résidant dans la RAM (tmpfs), comme un ramdisk
  - Autres répertoires dans tmpfs : `/dev/shm`, `/tmp`, `/var/lock`
  - Ces répertoires migreront dans `/run`

Les processus

— Leur structure :

Structures en mémoire identifiée par un numéro unique, le PID

Arborescence, partant du processus `init`, de PID 1 (commande `pstree`).

Autres informations : `PPID`, propriétaire, commande, répertoire courant, priorité,...

— Les voir : la commande `ps`, typiquement `ps auxww`

```
USER       PID   VSZ  RSS TTY      STAT START   TIME COMMAND
.........
root      1273  1376  424 ?        S    Apr26   0:00 /usr/sbin/apmd -p 10 -w 5 -W -P /etc/sysconfig/apm-scripts/a
root      1299  1632 1632 ?        SL   Apr26   0:00 xntpd -A -c /etc/ntp/ntp.conf
root      1310  2076  160 ?        S    Apr26   0:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
root      1371  1424   52 ?        S    Apr26   0:00 gpm -t imps2 -m /dev/mouse
root      1389  1616  484 ?        S    Apr26   0:01 crond
.........
```

— Les gérer : envoi d'un signal avec la commande `kill`

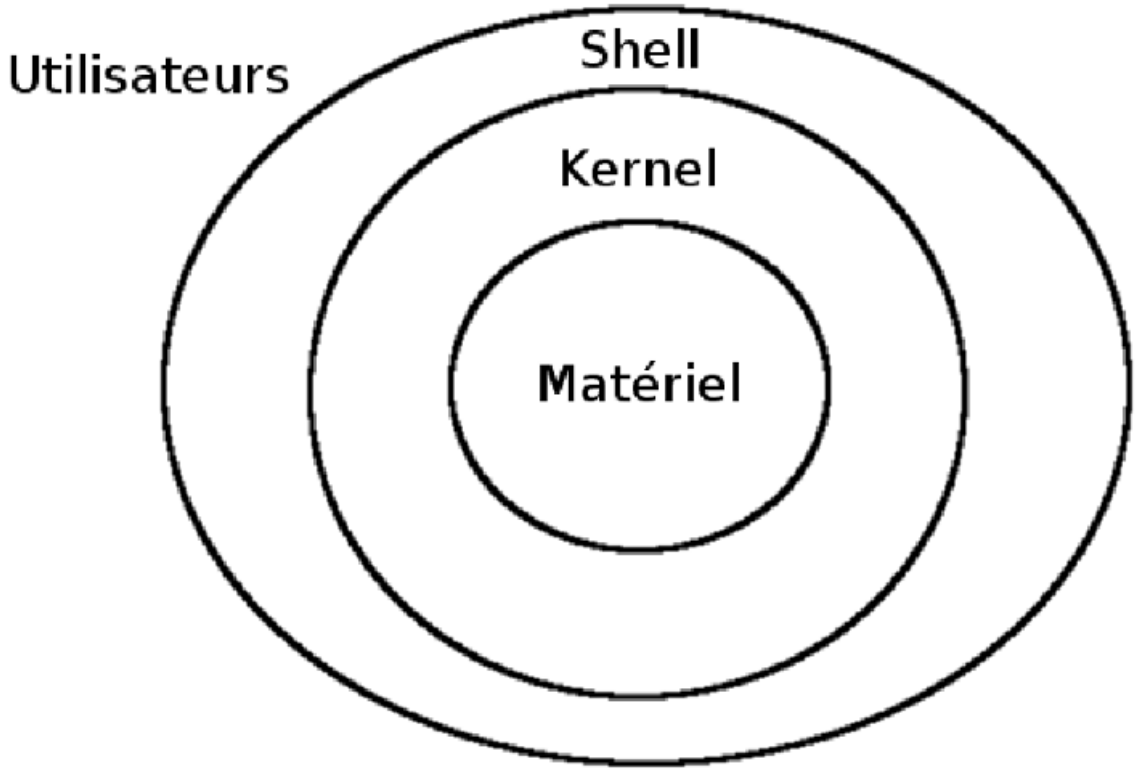
`kill -STOP` *`pid`*

`kill -CONT` *`pid`*

`kill -HUP` *`pid`*

`kill -KILL` *`pid`*

Les shells



Les shells

- Shells courant : sh, bash, csh, ksh, tcsh, zsh
- Modification dans `/etc/passwd`
- Changement de shell en ligne de commande : `/bin/bash`
- Installation via le système classique de package
- Configuration utilisateur : `/home/user/.fichier_de_conf`

Les variables d'environnement

- Configuration des logiciels du système
  - `LANG=fr_FR.UTF-8`
  - `export EDITOR=nano`
- Affichage des variables
  - `printenv`

```
.........
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PWD=/home/users/ftalbart
GDM_KEYBOARD_LAYOUT=fr latin9
LANG=fr_FR.utf8
GDM_LANG=fr_FR.utf8
CLUTTER_PAINT=disable-clipped-redraws:disable-culling
GDMSESSION=default
SHLVL=1
HOME=/home/users/ftalbart
.........
```

- La portée est locale

La Documentation

- Le manuel : `man` !
  - `man` *`commande`*
  - `man man`
  - `man 1 kill` : dans la section 1 (commandes utilisateur)
  - `man 2 kill` : dans la section 2 (appels système)
  - `man -k kill` : `-k` comme keyword, a.k.a. `apropos`

  Plus spécifique Linux :
- Documents dans `/usr/doc`, `/usr/share/doc`
- Sites Web :

  `http://www.lea-linux.org/`
- Google ! (les messages d'erreur par exemple)

LE SYSTÈME DE FICHIERS

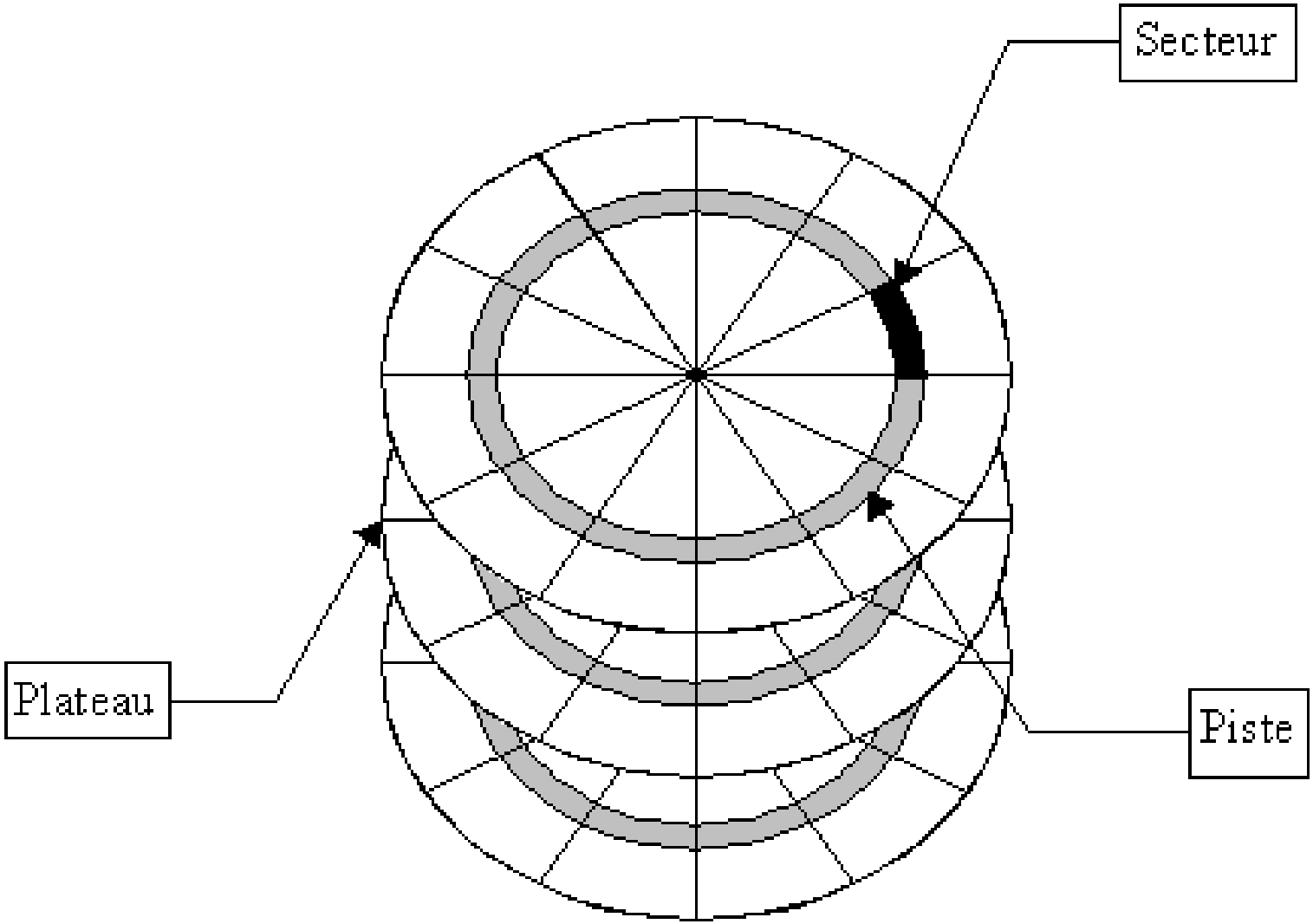
See, you not only have to be a good coder to create a system like Linux, you have to be a sneaky bastard too.

Linus Torvalds

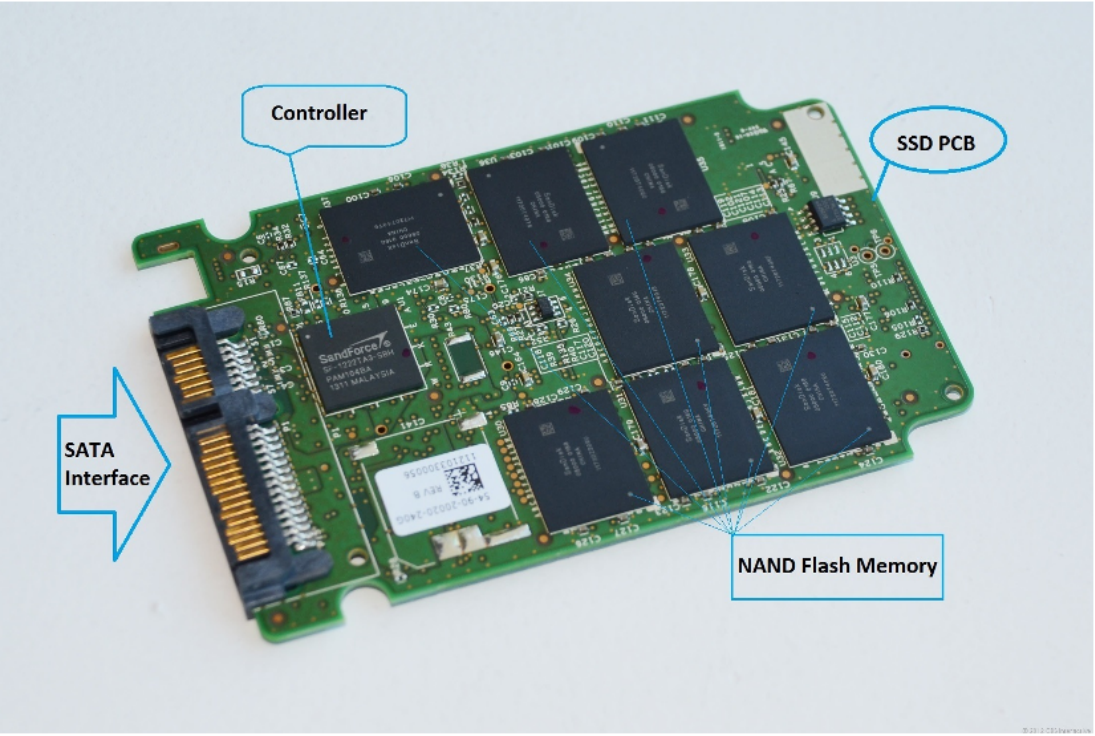
Gestion des disques et des fichiers

- Disque = unité de disques
- Composé de plateaux
- Chacun des plateaux contient plusieurs pistes
- Chaque piste contient plusieurs secteurs
- Cylindre = ensemble de pistes de même numéro situées sur tous les plateaux
- Géométrie d'un disque :
  - nombre de cylindres
  - nombre de pistes par cylindre
  - nombre de secteurs par piste

Gestion des disques et des fichiers



Gestion des disques et des fichiers



Gestion des disques et des fichiers

- Plus d'actions mécaniques
- Trim à la volé sur les dernières distributions (EXT4 et BTRFS)
  - Temps d'accès aléatoire : SSD : Environ 0,1 ms DD : De 2,9 à 12ms
  - Vitesse de lecture/écriture : SSD : De 27 Mo/s à 3 Go/s DD : De 12 à 260 Mo/s
  - Fragmentation : SSD : Aucun effet DD : Dépend du système de fichiers
  - Inconvénient : SSD : Sensible au nombre de cycles d'écriture DD : Coupures de courant qui peuvent rendre le lecteur irrécupérable sur certains (anciens) modèles (désactivation des caches sur certains modèles) Chocs et vibrations, sensibles aux champs magnétiques
- NVMe (Non-Volatile Memory express) : SSD utilisant le port PCI Express : performances accrues

Partitionnement d'un disque

- Un disque peut être décomposé en partitions
- Chaque partition peut contenir :
  - un système de fichiers (données et structures de contrôle)

    ou
  - une zone de swap
- Plusieurs types de systèmes de fichiers :
  - System V (basé sur le système de fichiers de la V7)
  - BSD (introduit dans 4.2BSD)
  - Ext2fs sous Linux (inspiré par le système de fichiers BSD)

Rappel (?) sur les fichiers et les i-nœuds

- Inode ou i-nœud : structure contenant
  - le numéro qui l'identifie
  - le type de la donnée référencée : répertoire, fichier, lien symbolique, pipe nommé, ...
  - les dates de création, modification, dernière lecture
  - le propriétaire et le groupe du fichier
  - le mode du fichier (droits, e.g. 0755)
  - le nombre de liens "hard links"
  - la liste des blocs contenant les données du fichier
- Référence aux i-nœuds : tables des répertoires

19	/tmp
29373	file-1.txt
29671	file-2.txt
29373	file-1h.txt
29712	file-1s.txt

Rappel (?) sur les liens

— “Hard Link” : nouvelle entrée, dans un répertoire, pour un i-nœud sur le même système de fichiers. Voir exemple `file-1.txt` / `file-1h.txt` page précédente.
Observation avec `ls` :

```
$ ln file-1.txt file-1h.txt
$ ls -li file-*
29373 -rw-------  2 fgilbert staff 220 Nov 21 12:05 file-1.txt
29373 -rw-------  2 fgilbert staff 220 Nov 21 12:05 file-1h.txt
```

— Lien symbolique : pseudo-fichier contenant le chemin du fichier sur lequel il pointe. Ce chemin peut être absolu ou relatif.
Observation avec `ls` :

```
$ rm file-1b.txt
$ ln -s file-1.txt file-1s.txt
$ ls -li file-*
29373 -rw-------  1 fgilbert staff 220 Nov 21 12:05 file-1.txt
29712 lrwxrwxrwx  1 fgilbert staff  10 Nov 21 12:09 file-1s.txt -> file-1.txt
```

Système de fichiers System V

- Un système de fichiers est composé de :
  - un secteur de boot
  - un superbloc
  - la table des i-nœuds
  - les blocs de données

Système de fichiers BSD

- Ensemble de groupes de cylindres
- Chaque groupe contient :
  - une copie du superbloc
  - des descripteurs de groupes
  - une partie de la table des i-nœuds
  - une partie des blocs de données
- Avantages :
  - robustesse (structures de contrôle dupliquées)
  - efficacité (routines d'allocation optimisées)

Détail de la structure

- Bloc de boot
- Groupe 1
  - Copie du superbloc
  - Descripteurs FS/groupe
  - Bitmap des blocs
  - Bitmap des inodes
  - Table des inodes
  - Blocs de données
- Groupe 2
- Groupe 3
- ...

Détail de la structure (2)

- Le superbloc
  - Localisation des bitmaps
  - Localisation de la table d'inodes
  - Nombre de blocs libres
  - Nombre d'inodes libres
  - Nombre de répertoires alloués
- Les descripteurs du FS pour chaque groupe
  - Nombre de blocs et d'inodes
  - Nombre de blocs libres et d'inodes libres
  - Taille des blocs
  - Nombre de blocs et d'inodes par groupe
  - Dates de dernier montage et de dernière écriture
  - Bit "clean"
  - Date de dernière vérification, intervalle de vérification obligatoire
  - Options de montage par défaut, emplacement du dernier point de montage

Nommage des disques

- Solaris 2 : `/dev/[r]dsk/c`*C*`t`*A*`d0s`*P*
- HP-UX : `/dev/[r]dsk/c`*S*`1d`*A*`s`*P*
- SunOS : `/dev/[r]sd`*XC*
- FreeBSD :
  - IDE : `/dev/[r]wd`*XC*, `/dev/[r]ad`*XC*
  - SCSI : `/dev/[r]da`*XC*
- Linux :
  - IDE : `/dev/hd`*CX*
  - SCSI : `/dev/sd`*CX*

Les différentes interfaces

- SATA (Serial Advanced Technology Attachment)
  - ATA (IDE) vers SATA : lecteurs CD, disques durs...
  - SATA : technologie utilisant le bus série et non plus parallèle. v3.2 : 1969 MB/S
  - Moins contraint par la longueur du câble
- SCSI (Small Computer System Interface)
  - Déporte la complexité
  - A utilisé la plupart du temps un bus parallèle, SCSI 3 : série
  - Fut sensible aux bruits et restreint la longueur du câble
  - Limitations réglés par SAS
- Serial-Attached SCSI (SAS)
  - Protocole SCSI sérialisé et amélioré

Description des disques

- Les disques connus sont généralement décrits dans un fichier (`/etc/disktab` ou `/etc/format.dat`)
- Champs décrivant un disque :
  - ty=nom : type du disque
  - se#N : taille des secteurs en octets
  - ns#N : nombre de secteurs par piste
  - nt#N : nombre de pistes par cylindre
  - nc#N : nombre de cylindres
  - rm#N : vitesse de rotation
  - b*n*#N : taille des blocs de la partition *n*
  - f*n*#N : taille des fragments de la partition *n*
  - p*n*#N : taille de la partition *n* en secteurs

Exemple de description (1)

— `/etc/format.dat` (SunOS) :

```
disk_type = "Micropolis 1558" \
   : ctlr = MD21 \
   : ncyl = 1218 : acyl = 2 : pcyl = 1224 \
   : nhead = 15 : nsect = 35 \
   : rpm = 3600 : bpt = 20833...
partition = "Micropolis 1558" \
   : disk = "Micropolis 1558" : ctlr = MD21 \
   : a = 0, 32025 : b = 61, 59850 : c = 0, 639450 \
   : g = 175, 547575
```

Exemple de description (2)

— `/etc/disktab` (HP-UX) :

```
MICROP_1588T_96MB:\
        :96 Mb reserved for swap & boot:\
        :ns#28:nt#15:nc#1318:\
        :s0#553560:b0#8192:f0#1024:\
        :se#512:rm#3600:
```

Formatage des disques

- Solaris 2 : `format`
- HP-UX : `mediainit`
- IRIX : `fx -x`
- SunOS : `format`
- FreeBSD, Linux : Moniteur du contrôleur, `sformat`

Partitionnement de disques

- Solaris 2 : `format`, `prtvtoc`
- HP-UX 9 : Pas de partitionnement
- HP-UX 10 et 11 : LVM
- IRIX : `fx`, `prtvtoc`
- SunOS : `format`, `dkinfo`
- FreeBSD : `fdisk`, `disklabel`
- Linux : `fdisk`, `parted`

Exemples - Linux

```
# /sbin/fdisk /dev/sda
Command (m for help): p
Disk /dev/sda: 64 heads, 32 sectors, 1010 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot   Begin    Start       End   Blocks   Id  System
/dev/sda1   *        1        1        51    52208   83  Linux native
/dev/sda2           52       52       152   103424   82  Linux swap
/dev/sda3          153      153       253   103424   82  Linux swap
/dev/sda4          254      254      1010   775168   83  Linux native
```

Systèmes de fichiers

- Création de système de fichiers :
  - `mkfs` (IRIX)
  - `mke2fs` (Linux)
  - `newfs` (Autres)
- Liste des systèmes de fichiers à monter :
  - `/etc/vfstab` (Solaris 2)
  - `/etc/checklist` (HP-UX 9)
  - `/etc/fstab` (Autres)
- Type de système de fichiers :
  - SunOS, Solaris2 : `ufs`
  - HP-UX : `hpfs`
  - BSD : `4.2`
  - Linux : `ext2`, `ext3`, `ext4`, `reiserfs`, ...
  - Multiples types de FS reconnus $\rightarrow$ nécessité du File System Switch ; e.g. driver VFS sous Linux.

Création d'un système de fichiers

— Généralement :

```
newfs fichier_spécial
mkfs fichier_spécial
```

— Sous HP-UX 9 :

```
newfs fichier_spécial type_du_disque
```

— Optimisations : options de `newfs` ou `mkfs`, par exemple :

  — `-i` *N* : nombre d'octets par i-nœud

  — `-c` *N* : nombre de cylindres par groupe

  — `-m` *N* : pourcentage de blocs réservés à root

  — `-o space|time` : optimisation

Montage de système de fichiers

- Montage : `mount`
- Démontage : `umount`
- Syntaxes :

```
mount [options] fichier_spécial répertoire
mount [options] fichier_spécial|répertoire
mount [options]
umount -a [-v]
umount [-v] fichier_spécial|répertoire
```

Options de montage (1)

- `-v` : verbeux
- `-r` : lecture seule
- `-t` type :
  - `ufs` : Solaris 2, BSD
  - `hfs` : HP-UX
  - `efs` : IRIX
  - `4.2` : SunOS
  - `ext2` : Linux
- `-u` : modification des options de montage

Options de montage (2)

- `-o options` : options de montage
  - `rw`, `ro`
  - `nosuid`
  - `nodev`
  - `noexec`
  - `quota`, `noquota`
  - `sync`, `async`
  - `remount`
- `-a` : montage de tous les systèmes de fichiers
- `-n` : pas de mise à jour de la liste des systèmes de fichiers montés (généralement `/etc/mtab`)

Liste des systèmes de fichiers

- `/etc/fstab`, `/etc/checklist`, `/etc/filesystems`
- Chaque ligne décrit un système de fichiers :
  - fichier spécial
  - point de montage (répertoire)
  - type du système de fichiers
  - options de montage
  - intervalle entre deux sauvegardes
  - ordre de vérification

Exemples (1)

— /etc/fstab (SunOS) :

```
/dev/sd0a  /        4.2 rw,grpid 1 1
/dev/sd0g  /usr     4.2 rw,grpid 1 2
/dev/sd1g  /users   4.2 rw,grpid 1 3
/dev/sd1h  /spare   4.2 rw,grpid 1 3
```

— /etc/vfstab (Solaris 2) :

```
#device            device             mount FS   fsck mount
#to mount          to fsck            point type pass at boot
/dev/dsk/c0t3d0s0 /dev/rdsk/c0t3d0s0 /    ufs  1    no   -
/dev/dsk/c0t3d0s3 /dev/rdsk/c0t3d0s3 /usr ufs  1    no   -
```

Exemples (2)

— /etc/checklist (HP-UX 9) :

```
/dev/dsk/c201d6s0   /               hfs     defaults          0 1
/dev/dsk/c201d4s0   /usr/local      hfs     defaults          0 2
default             /usr/local/swap swapfs  res=200000,pri=10 0 0
```

— /etc/filesystems (AIX) :

```
/:
        dev             = /dev/hd4
        vfs             = jfs
        log             = /dev/hd8
        mount           = automatic
        check           = false
        type            = bootfs
        vol             = root
        free            = true
```

Vérifications de cohérence

- Un système de fichiers peut être corrompu
- Vérification de la cohérence : `fsck`
- Cinq passes :
  - vérification des i-nœuds
  - vérification des répertoires
  - restauration des fichiers et/ou répertoires non connectés
  - vérification du nombre de liens
  - vérification des tables de blocs/i-nœuds libres
- Attention : `fsck` doit être exécuté uniquement sur des systèmes de fichiers non actifs !

Exemple d'exécution

```
# fsck -n /dev/rsd1g
** /dev/rsd1g (NO WRITE)
** Currently Mounted on /users
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
33913 files, 1783839 used, 231118 free (23262 frags, 25982
  blocks, 1.2% fragmentation)
```

Options de fsck

- Sous BSD :
  - `-p` : corrections automatiques
  - `-b bloc` : adresse du superbloc
  - `-y` : réponse 'oui' à toutes les questions
  - `-n` : réponse 'non' à toutes les questions
- Sous System V :
  - `-b` : redémarrage automatique si la racine est modifiée
  - `-y|-n` : idem BSD
  - `-q` : corrections automatiques
  - `-D` : recherche de blocs erronés dans les répertoires
  - `-f` : vérification rapide
  - `-s` : reconstruction de la liste des blocs libres

Autres commandes

- `clri fichier_spécial N` : remise à zéro d'un i-nœud
- `fsdb fichier_spécial` : débogueur de système de fichiers (System V) (debugfs sous Linux)
- `dumpfs fichier_spécial` : affichage des paramètres du système de fichiers (`dumpe2fs` sous Linux)
- `tunefs fichier_spécial` : modification des paramètres du système de fichiers (`tune2fs` sous Linux)
- `df` : affichage de l'espace disponible (blocs ou i-nœuds)
  options : `-k` (en kilo-octets), `-h` ("human-readable", récent).

Journalisation : `ext2` → `ext3`

— Journalisation : écriture synchrone, d'un historique des écritures asynchrones.

  En cas d'arrêt brutal : on examine / "rejoue" ce journal, plutôt que d'examiner la totalité du disque

— `ext3` : ajout d'une mécanique de journalisation à `ext2`

  Création : `mke2fs -j /dev/sdX`,

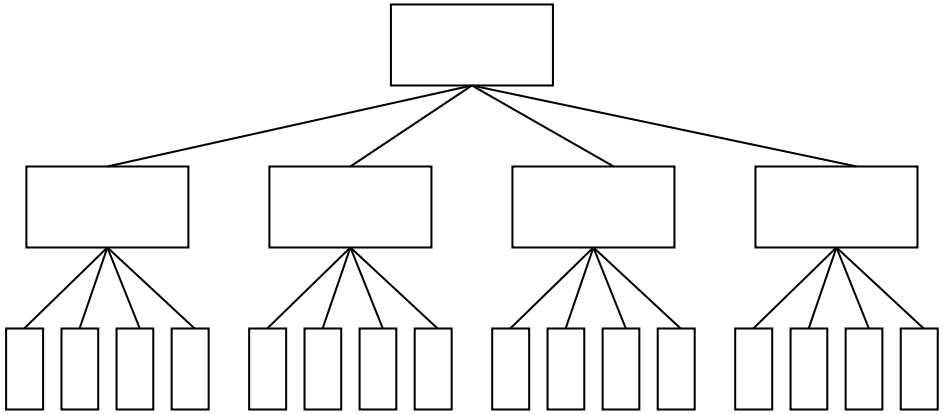
  ou `tune2fs -j /dev/sdX` pour ne pas réinitialiser.

— Performance (par rapport à ext2) :

  - lecture : +0%
  - copie : ≈ +15%
  - déplacement : ≈ +60%
  - destruction : ≈ +100%

Journalisation : reiserfs

— Représentation des répertoires et fichiers dans un arbre balancé (toutes les feuilles sont à la même profondeur) $\Rightarrow$ temps d'accès aux fichiers uniforme



— Journalisé. Commandé par le DOD, en principe très robuste

— `mkreiserfs`, `reiserfsck`, type de montage `reiserfs` dans `/etc/fstab`.

— Performance (par rapport à ext2) :

- lecture : +0%
- copie : $\approx$ +10%
- déplacement : $\approx$ +6%
- destruction : $\approx$ -45% !

ext4

— Successeur de ext3, développé en 2006

  Permet une transition vers BTRFS

— Améliorations (par rapport à ext3) :

  - Taille de volume jusqu'à un exbioctet (2^{60} octets)
  - Minimise la fragmentation par préallocation contiguë
  - Compatiblité ascendante et descendante avec ext3 (mais code entièrement nouveau)

Un petit mot sur ZFS et BTRFS

- ZFS
  - Conçu par SUN
  - Enorme capacité de stockage
  - Gestionnaire de volume, snapshot...
  - Correction d'erreurs efficace
- BTRFS
  - Conçu par Oracle et RedHat
  - Création de sous-volumes
  - Snapshot et intégration des données
  - Pourrait remplacer ext4

Partitions de swap (1)

- Unix utilise des partitions de swap pour stocker sur disque des données mémoire
- Création d'une partition de swap : comme une autre partition (sauf sous HP-UX 9)
- Sous Linux, après création : `mkswap fichier_spécial`
- Activation :
  - `swapon fichier_spécial` : SunOS, Linux, FreeBSD,
  - `swapon -e fichier_spécial` : HP-UX
  - `swap -a fichier_spécial` : Solaris 2, IRIX

Partitions de swap (2)

- Liste des partitions de swap à utiliser :
  - Solaris 2 : dans `/etc/vfstab` :

    ```
    fichier_spécial -    -    swap -    no   -
    ```

  - HP-UX 9 : dans `/etc/checklist` :

    ```
    fichier_spécial -    swap end  -    -
    ```

  - AIX : dans `/etc/swapspaces` :

    ```
    hd6:
          dev = /deb/hd6
    ```

  - Autres : dans `/etc/fstab` :

    ```
    fichier_spécial swap swap rw   0    0
    ```

- Ces partitions sont activées lors du démarrage du système (appel de `swapon -a` ou équivalent).

Taille de la mémoire virtuelle (1)

— FreeBSD : `swapinfo`

```
Device      1K-blocks       Used      Avail     Capacity  Type
/dev/wd0b       32550      25936       6550         80%    Interleaved
```

— HP-UX : `swapinfo`

```
             Kb      Kb      Kb   PCT  START/      Kb
TYPE    AVAIL    USED    FREE  USED   LIMIT RESERVE  PRI  NAME
dev     99425    7853   91572    8%  312390       -    0  /dev/dsk/c201
hold        0   21188  -21188
```

— IRIX : `swap -l`

Taille de la mémoire virtuelle (2)

— SunOS : `pstat -T`

```
290/1888 files
768/1018 inodes
 94/522 processes
40196/131036 swap
```

— Solaris 2 : `swap -l`

```
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t3d0s1   32,25      8 262632 243608
```

— Linux : `free`

```
             total       used       free     shared    buffers     cach
Mem:         63320      60068       3252      50216      16700      150
-/+ buffers/cache:      28360      34960
Swap:       136512       3192     133320
```

Désactivation du swap

— Désactivation possible sous certains systèmes

— Solaris 2 :

```
swap -d fichier_spécial
```

— Linux :

```
swapoff -a
swapoff fichier_spécial
```

Quotas disque

- Big Brother finally hits Unix !
- Principe : imposer une limite sur l'espace disque utilisable par chaque utilisateur et/ou groupe d'utilisateurs
- Limites sur :
  - le nombre de blocs
  - le nombre de fichiers
- Dépassement de limite : erreur
- Les quotas sont spécifiques à chaque système de fichiers

Limites

- Deux limites
- Limite ’douce’ :
  - peut être dépassée (avertissement)
  - devient équivalente à la limite absolue au bout d’un délai de grâce
- Limite absolue :
  - est supérieure à la limite ’douce’
  - ne peut pas être dépassée (erreur)
- Délai de grâce :
  - Délai laissé à l’utilisateur pour réduire son occupation sous la limite ’douce’
  - 7 jours par défaut

Définition des quotas

- Les quotas sont spécifiques à chaque système de fichiers
- Les limites peuvent être différentes (voire inexistantes) sur des systèmes de fichiers différents
- Fichiers de définition des quotas :
  - `quota.user` : limites par utilisateur (OSF/1, Linux, BSD)
  - `quota.group` : limites par groupe (OSF/1, Linux, BSD)
  - `quotas` : limites par utilisateur (autres systèmes)
- Pas de manipulation directe des fichiers : commandes d'administration des quotas

Activation des quotas (1)

- Une option doit être définie pour chaque système de fichiers :
  - `quota` (SunOS, HP-UX)
  - `rq` (Solarix 2, IRIX)
  - `userquota`, `groupquota` (OSF/1, BSD)
  - `usrquota`, `grpquota` (Linux)
- Activation : `quotaon [options] [fichier_spécial]`
  - `-a` Activation sur tous les systèmes de fichiers
  - `-v` Verbeux
  - `-u` Activation des quotas liés aux utilisateurs
  - `-g` Activation des quotas liés aux groupes

Activation des quotas (2)

— La commande `quotaon` est normalement appelée lors du démarrage (scripts d'initialisations)

  Les quotas sont alors activés sur les montages qui ont l'option `quota` dans `/etc/fstab`.

— Désactivation de quotas disque :

  `quotaoff [options] [fichier_spécial]`

  — `-a` Désactivation sur tous les systèmes de fichiers

Vérification des quotas

- En cas de crash, les fichiers de définition des quotas peuvent être corrompus
- Vérification des fichiers de quotas :

  `quotacheck [options] [système_fichiers]`

  - `-a` Vérification sur tous les systèmes de fichiers
  - `-v` Verbeux
  - `-u` Vérification des quotas liés aux utilisateurs
  - `-g` Vérification des quotas liés aux groupes
  - `-p` Vérifications en parallèle
- Normalement exécuté automatiquement au démarrage avant l'activation des quotas

Affectation de limites

- Lancement d'un éditeur pour modifier les limites liées à un utilisateur ou à un groupe : `edquota`
- Options :
  - `-u` Edition des limites d'un utilisateur
  - `-g` Edition des limites liées à un groupe

```
Quotas for user dugenou:
/dev/hdb2: blocks in use: 16, limits (soft = 5000, hard = 6000)
           inodes in use: 11, limits (soft = 100, hard = 110)
/dev/sda1: blocks in use: 0, limits (soft = 0, hard = 0)
           inodes in use: 0, limits (soft = 0, hard = 0)
```

- Copie de limites :

```
edquota -p prototype utilisateur
edquota -p prototype -g groupe
```

- Positionnement de limites :

```
setquota -u utilisateur 10000
```

Affichage des limites

— Affichage des limites et occupations : `repquota`

```
*** Report for user quotas on /dev/hdb2 (/home)
Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
root      --      19       0       0              2     0     0
bin       --       3       0       0              3     0     0
news      --   31691       0       0          11846     0     0
card      -- 111498       0       0           8673     0     0
dugenou   --      16    5000    6000             11   100   110
melanie   --       6    5000    6000              7   100   110
```

— Affichage des limites et de l'occupation d'un utilisateur : `quota`

```
           Disk quotas for user dugenou (uid 1004):
Filesystem blocks quota limit grace  files quota limit   grace
/dev/hdb2     16  5000  6000           11   100   110
```

GESTION DES UTILISATEURS

The Linux philosophy is 'Laugh in the face of danger'. Oops. Wrong One. 'Do it yourself'. Yes, that's it.

Linus Torvalds

Les utilisateurs

- Tout utilisateur est caractérisé par :
  - un nom
  - un numéro d'utilisateur
  - un numéro de groupe
  - un mot de passe
  - un shell
- Les utilisateurs sont définis dans `/etc/passwd` (et `/etc/shadow`)
- Les groupes d'utilisateurs sont définis dans `/etc/group`

Fichier /etc/passwd

- Liste des utilisateurs
- Chaque ligne contient :
  - le nom (login)
  - le mot de passe chiffré
  - le numéro d'utilisateur (uid)
  - le numéro de groupe d'utilisateurs (gid)
  - le nom complet (champ "GECOS")
  - le répertoire d'accueil
  - le shell
- Exemples :

```
root:20xI7leSjX1sY:0:0:Le chef:/:/bin/sh
card:fFi332cQDb7Gw:1001:10:Remy Card:/users/card:/bin/csh
```

Utilisateurs spéciaux

- root : administrateur (uid = 0)
- daemon : utilisateur fictif des démons
- bin : propriétaire de `/bin` et de `/usr/bin`
- sys : utilisateur système (System V)
- adm : propriétaire des fichiers de comptabilité
- uucp : utilisateur pour les connexions UUCP
- lp : utilisateur administrateur de l'impression

Groupes d'utilisateurs

- Un utilisateur appartient à :
  - un groupe primaire
  - plusieurs groupes secondaires
- Les groupes associés à un utilisateur sont utilisés pour les contrôles d'accès
- Changement de groupe courant :
  - automatique sous BSD et System V récent
  - `newgrp` sous les vieux System V

Fichier /etc/group

- Définition des groupes et des utilisateurs associés
- Chaque ligne contient :
  - le nom du groupe
  - le mot de passe chiffré (utilisé par `newgrp`)
  - le numéro du groupe
  - la liste des utilisateurs du groupe
- Exemples :

```
wheel::0:root
staff::10:root,card
admin::101:card
```

Le fichier 'shadow'

- Le fichier `/etc/passwd` est en lecture pour tous
- Le mot de passe chiffré n'est pas déchiffrable ...
- ... mais une attaque brutale à base de dictionnaires peut aboutir (exemple : Crack)
- Sous certains systèmes (principalement System V), la liste des utilisateurs est décomposée en deux fichiers :
  - `/etc/passwd` (sans mots de passe) lisible par tous
  - `/etc/shadow` (avec mots de passe) lisible par 'root' uniquement
- Le mot de passe peut être un caractère (e.g., `:x:` reflétant une redirection vers un autre moyen d'authentification, ou `:*:` empêchant l'authentification).

Fichier /etc/shadow

- Chaque ligne contient :
  - le nom de l'utilisateur
  - le mot de passe chiffré
  - la date du dernier changement de mot de passe
  - le nombre minimum de jours entre deux changements du mot de passe
  - le nombre maximum de jours de validité du mot de passe
  - le nombre de jours avant l'expiration du mot de passe à partir duquel l'utilisateur est averti
  - le nombre de jours pendant lesquel le compte peut être inutilisé
  - la date d'expiration du compte
  - un champ 'réservé'

Fichier /etc/master.passwd (FreeBSD)

- Chaque ligne contient :
  - le nom de l'utilisateur
  - le mot de passe chiffré
  - le numéro d'utilisateur (uid)
  - le numéro de groupe d'utilisateurs (gid)
  - la classe d'utilisateur
  - la date du dernier changement de mot de passe
  - la date d'expiration du compte
  - le nom complet de l'utilisateur
  - le répertoire d'accueil
  - le shell
- Les paramètres ci-dessus concernant la validité du compte et du mot de passe ont des valeurs par défaut dans `/etc/login.defs`

Gestion des utilisateurs (1)

- Édition du fichier `/etc/passwd` : `vipw`
- Changement de mot de passe :
  `passwd [options] [utilisateur]`
  - `-f` : changement du nom complet
  - `-s` : changement du shell (doit être listé dans `/etc/shells`)
- Changement de nom complet et de shell : `chfn`, `chsh`
- Modification de `/etc/passwd` et `/etc/shadow` : `passmgmt`
  - `-d` *`utilisateur`* : suppression
  - `-a` *`utilisateur`* : création
  - `-m` *`utilisateur`* : modification
  - options de `-a` et `-m` : `-c`*`nom`*, `-h`*`répertoire`*, `-u`*`uid`*, `-g`*`gid`*, `-s`*`shell`*, `-l`*`login`*
- Changement de paramètres de l'utilisateur (Linux) : `usermod`

Gestion des utilisateurs (2)

- Vérification de `/etc/passwd` : `pwck`
- Vérification de `/etc/group` : `grpck`
- Conversion du fichier `/etc/shadow` :

  `pwconv`

  `pwunconv`
- Création des fichiers hachés :

  `mkpasswd`

  `pwd_mkdb`

Création d'un utilisateur

- Ajout dans `/etc/passwd` (et dans `/etc/group` éventuellement)
- Enregistrement du mot de passe : `passwd utilisateur`
- Création du répertoire :

```
mkdir répertoire
chown utilisateur répertoire
chgrp groupe répertoire
```

- Création des fichiers d'initialisation
- Sous Linux : la commande `adduser` (ou `useradd`, c'est pareil) fait tout cela.

  Exemple :

```
adduser -u 12345 -s /bin/bash -c "Frederic Gilbert" -p amMcXKoJqL7S.
        -d /home/gilbert -m -g staff gilbert
```

  Le répertoire est créé à l'image d'un répertoire `/etc/skel`

Suppression d'un utilisateur

- Invalidation du compte :
  - remplacement du mot de passe chiffré par '*' ou '**No Login**'
  - remplacement du shell par `/bin/false`
- Suppression effective :
  - suppression du répertoire d'accueil
  - suppression de tous les fichiers de l'utilisateur :
    - mailbox
    - crontab
    - etc ...
  - suppression dans `/etc/passwd` (et dans `/etc/group` éventuellement)
- Sous Linux, la commande `userdel` fait tout ça ...

LE NOYAU

People disagree with me. I just ignore them.

Linus Torvalds, regarding the use of C++ for the Linux kernel.

Le noyau, composition et fabrication

— Cœur du système : séquenceur, gestion de la mémoire, etc.
  - Code principal
  - Drivers : fonctionnalités (VFS, TCP-IP, ...), ou matériel (SCSI, ethernet, ...)
    · “built-in”, intégrés au code du noyau
    · modules, chargement dynamique, config éventuelle

— Fabrication
  - Pourquoi ?
    · pour ajouter un pilote (périphérique, ...)
    · pour supprimer les pilotes inutiles (tuning en taille)
    · pour adapter aux composants (tuning en performance)
  - Trois grands types de mode de configuration :
    · BSD (SunOS, HP-UX, *BSD), System V et Linux

Le noyau, composition et fabrication

— Fabrication
  - Grandes étapes :
    · configuration (choix des options)
    · choix des composants du source (lié au choix et aux dépendances)
    · compilation puis installation du noyau et des modules
    · si nécessaire, reparamétrage du boot (loader)

— Ajout d'un driver
  - Version binaire (déjà compilée) : pour une version donnée seulement (noyau standard des distributions)
  - Version source :
    · compilation du driver avec les "includes" du noyau préalablement compilé
    · ou recompilation du noyau en intégrant les sources du driver, parties built-in et/ou partie module.

Configuration d'un noyau BSD

- Fichier de configuration :
  - description du matériel
  - sélection d'options
- Création d'une nouvelle configuration :

```
config fichier_configuration
```

- Compilation du noyau :

```
make depend
make
```

- Installation
- Redémarrage

Fichier de configuration BSD

- Suite de déclarations :
  - `machine type`
  - `cpu type`
  - `ident nom_noyau`
  - `maxusers nombre_d_utilisateurs`
  - `options option`
  - `config racine_et_swap`
  - `controller`
  - `disk`
  - `tape`
  - `device`
  - `pseudo-device`

Configuration d'un noyau Linux (2.6.*)

— Configuration :

```
cd /usr/src/linux; make config
```

ou

```
cd /usr/src/linux; make xconfig
```

ou

```
cd /usr/src/linux; make menuconfig
```

— Compilation du noyau :

`make` ou

`make bzImage` + `make modules`

— Installation :

```
make install
make modules_install
```

— Pour compiler un driver, il est possible de générer uniquement les fichiers à inclure :

`make prepare-all`, `make modules_prepare`

Exemple (make config)

```
# make config
...
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
Set version information on all symbols for modules (CONFIG_MODVERSIONS) [Y/n
Kernel module loader (CONFIG_KMOD) [Y/n/?]
*
* General setup
*
Networking support (CONFIG_NET) [Y/n/?]
PCI support (CONFIG_PCI) [Y/n/?]
...
```

4000 lignes/questions ! . . .

Démarrage d'un noyau Linux

- Noyau chargé par un programme externe
- Généralement LILO (LInux LOader)
- Installation d'un nouveau noyau :
  - copie de l'image (`arch/i386/boot/zImage` ou `arch/i386/boot/bzImage`)
  - configuration de `lilo` : édition de `/etc/lilo.conf`
  - exécution de `lilo` pour prendre les changements en compte

Démarrage du système par GRUB

— GRand Unified Bootloader. Version ré-écrite : GRUB2.

— Lecture du fichier de configuration au démarrage (pas d'étape d'installation comme Lilo).

— Fichier `/boot/grub/grub.cfg`, généré automatiquement (commande `update-grub`)

— Possibilité de passage en mode prompt et de saisir les instructions de boot, et mode "`grub rescue`" en cas d'échec d'utilisation de `grub.cfg`

Exemple de fichier `grub.cfg`

```
default 0
timeout 5
#
foreground = ffffff
background = 000000
#
splashimage=(hd0,1)/boot/grub/leaf_splash.xpm.gz
#
title Linux 2.2.14
root (hd0,1)
kernel /boot/vmlinuz-2.2.14 root=/dev/sda2 quiet
initrd /boot/initrd-2.2.14.img
#
title Windows
root (hd1)
chainloader +1
```

Ajout de périphérique

- Pilotes de périphériques caractérisés par :
  - un type (caractère ou bloc)
  - un numéro majeur

    Le numéro majeur est utilisé comme indice dans une table interne du noyau
  - un numéro mineur

    Le numéro mineur est utilisé pour distinguer un périphérique d'un type (numéro majeur) donné.
- Ajout d'un pilote :
  - intégration du pilote dans une table du noyau
  - modification du fichier de configuration
  - regénération d'un noyau

Fichiers de configuration

- Solaris 2 : `/usr/kernel/drv/*conf`, `/usr/kernel/drv/*`
- HP-UX 9 : `/etc/master`, `/etc/conf/dfile`
- HP-UX 10 : `/usr/conf/master.d/*`, `/stand/system`
- SunOS : `/sys/sunX/conf/NOYAU`, `/sys/sunX/conf/files*`
- OSF/1 : `/sys/conf/NOYAU`, `/sys/conf/files*`
- FreeBSD : `/sys/i386/conf/NOYAU`, `/sys/i386/conf/files*`

Intégration dans BSD (1)

— Ajout d'une entrée dans `files.`*`NOYAU`* :

```
local/pilote.o   optional périph      device-driver
```

— Placement des objets dans `/sys` :

```
mkdir /sys/local
cp pilote.o /sys/local/pilote.o
```

— Intégration dans une table : édition de conf.c

  — table des périphériques en mode caractère : `cdevsw`

  — table des périphériques en mode bloc : `bdevsw`

Intégration dans BSD (2)

— Exemple :

```
extern int drv_open(), drv_close(), drv_read();
...
struct cdevsw cdevsw[] =
{
   ...
   {    drv_open, drv_close, drv_read, nodev,
        nodev,    nodev,     nodev,    0,
        nodev,    0,         0,
   },
```

— Modification du fichier de configuration :

```
device-driver   driver
```

— Reconstruction du noyau

Intégration dans HP-UX

— Définition des pilotes dans `/etc/master` ou dans `/usr/conf/master.d`:

```
périph  pilote    type masque    bloc carac
...
périph  libpilote.a
...
```

— Ajout du nom du périphérique dans le fichier de configuration (`dfile` ou `/stand/system`):

```
périph
```

— Reconstruction d'un noyau

Intégration dans Solaris 2

- Pas besoin de reconfigurer le noyau
- Ajout dynamique de pilote dans le noyau
- Ajout : `add_drv`
- Suppression : `rem_drv`
- Chargement de module : `modload`
- Suppression de module : `modunload`
- Liste des modules : `modinfo`

Fichiers spéciaux

- Création des fichiers spéciaux correspondant au(x) périphérique(s) dans le répertoire `/dev`
- `mknod fichier type majeur mineur`
- Scripts de création :
  - `/dev/MAKEDEV`
  - `/dev/MAKEDEV.local`
- Exemple :

```
cd /dev
./MAKEDEV pty
```

- Solaris :

  `drvconfig` + `devlinks` + `disks` ou `tapes` + éventuellement `ucblinks` ou boot après un `touch /reconfigure`.
- Linux récents (noyau 2.6) :

  `udev`, configuration dans `/etc/udev.d/rules`

Udev

- Gestionnaire de pilotes
- Gère les périphériques de /dev
- Espace utilisateur
- Chargement dynamique

Fichiers spéciaux

— Extrait de `ls -l /dev` (Linux) :

```
crw-rw----+  1 root audio      14,   4 2010-11-22 16:37 /dev/audio
crw-------   1 root root        5,   1 2010-11-22 16:37 console
crw-r-----   1 root root       13,  32 2010-11-22 17:37 mouse0
crw-rw-rw-   1 root root        1,   3 2010-11-22 17:37 null

brw-rw----   1 root disk        8,   0 2010-11-22 17:37 sda
brw-rw----   1 root disk        8,   1 2010-11-22 17:37 sda1
.........   ...........       ..............................
brw-rw----   1 root disk        8,  16 2010-11-22 16:57 sdb
brw-rw----   1 root disk        8,  17 2010-11-22 16:57 sdb1

crw--w----   1 root root        4,   0 2010-11-22 17:37 tty0
crw-------   1 root root        4,   1 2010-11-22 16:37 tty1
crw-rw----   1 root dialout   166,   0 2010-11-22 16:37 ttyACM0
```

Fichiers spéciaux

— Ecrire dans la carte son :

```
#include <stdio.h>

void main(void)
{
    int t;
    for(t=0;; t++)
        putchar(
                ((t*9 \& t >> 4) |
                (t*5 \& t>>7) |
                (t*3 \& t/0x400))
                -1
               );
}


$ gcc a.c;  ./a.out > /dev/dsp1
```

Modules

- Module : sous-système chargé dynamiquement en mémoire
- Pas contenu de manière statique dans le noyau
- Supportés par :
  - SunOS
  - Solaris 2
  - IRIX
  - *BSD
  - Linux
- Deux types de chargements :
  - manuel
  - à la demande

Gestion des modules (1)

- Solaris 2 :
  - chargement : `modload`
  - suppression : `modunload`
  - liste : `modinfo`
- SunOS :
  - chargement : `modload`
  - suppression : `modunload`
  - liste : `modstat`
- IRIX : `ml`

Gestion des modules (2)

- *BSD :
  - chargement : `kldload`
  - suppression : `kldunload`
  - liste : `kldstat`
- Linux :
  - modules situés dans `/lib/modules/`*`version`*
  - chargement : `insmod`
  - suppression : `rmmod`
  - liste : `lsmod`, ou `cat /proc/modules`
  - dépendances : `depmod`, `modprobe`

    fichier `/lib/modules/`*`version`*`/modules.dep`
  - chargement à la demande : `kmod`

Installation d'outils (applications, système, librairies ...)

- A partir des sources : fichier *outil-version*`.tar.gz`
  - `./configure` : GNU autoconf, prend en compte les caractéristique de la machine
  - `make` : fabrique les binaires
  - `make install` : les installe au bon endroit dans le système.
- A l'aide d'un système de packaging
  - Sun, DEC : fichiers *outil-version*`.pkg`
    contient un fichier `.tar.gz` des binaires + des fichiers de description et des scripts, commande `pkgadd`
  - Linux (RedHat) : fichiers RPM, *outil-version*`.rpm`
    contenu équivalent, gestion des dépendances, commande `rpm`
  - Linux (Debian) : fichiers PKG, *outil-version*`.deb`
    contenu équivalent, gestion des dépendances, commande `dpkg`
  - Avec gestion automatique des dépendances : `yum` (RedHat), `apt-get` (Debian, Ubuntu, Xandros, ...).

La commande RPM

- Information : `rpm -q`
  - `-qa` : liste tous les RPMs installés
  - `-ql glibc` : liste de tous les fichiers issus du RPM (installé) donné en argument
  - `-qi glibc` : informations (version, licence, ...) sur le RPM (installé) donné en argument
  - `-qf /bin/ls` : donne le RPM d'où est issu le fichier
  - `-qpl glibc-2.3.3-21mdk.i586.rpm` : donne la liste des fichiers pouvant être installés par le *fichier* RPM donné en argument
- Installation : `rpm -i [options] produit-1.2.rpm`
- Mise à jour (upgrade) : `rpm -U [options] produit-1.2.rpm`
- Désinstallation : `rpm -E [options] produit`
- `--test` : simule l'opération
- `--nodeps` : ignore les conflits de dépendance
- `--force` : ignore les autres erreurs
- `--noscripts` : n'exécute pas les scripts de pré/post-(dés)installation.

Les paquets Debian

- Information :
  - `dpkg -l` : liste tous les paquets installés
  - `dpkg -L produit` : liste de tous les fichiers issus du paquet (installé)
  - `dpkg -s produit` : informations sur le paquet (installé)
  - `dpkg -S /bin/ls` : donne le paquet d'où est issu le fichier
  - `dpkg -c produit-1.2.deb` : donne le contenu du paquet
- Installation : `dpkg -i [options] produit-1.2.deb` ou le plus souvent `apt-get install produit` (ou source)
- Désinstallation : `dpkg -r [options] produit` ou `apt-get remove produit`
- Autres opérations spécifiques :
  - `apt-get upgrade` : met à jours tous les paquets pour lesquels une nouvelle version est disponible
  - `apt-get update` : met à jour la base des paquets disponibles dans les sources définies dans `/etc/apt/sources.list`

DÉMARRAGE D'UNIX

If you still don't like it, that's ok : that's why I'm boss. I simply know better than you do.

Linus Torvalds

Démarrage d'Unix

- Exécution d'un chargeur primaire
- Chargement et exécution d'un chargeur secondaire
- Chargement du noyau
- Exécution du noyau
  - détection et initialisation du matériel
  - lancement des processus système
    - `swapper` (0), `pagedaemon` (2) sous BSD
    - `sched` (0) sous System V
  - Exécution du processus `init` (processus numéro 1)

Rôle de init

- Exécution des scripts d'initialisation
- Gestion des connexions sur terminaux
- Ancètre de tous les processus
- Adoption des processus orphelins
- Deux types de programmes `init` :
  - BSD
  - System V

init BSD

- Deux niveaux d'exécution :
  - mono-utilisateur
  - multi-utilisateurs
- Lors du démarrage, exécution de :
  - `/etc/rc.boot` (SunOS)
  - `/etc/rc`
  - `/etc/rc.single` (SunOS en mode mono-utilisateur)
  - `/etc/rc.local`
- Gestion des connexions sur terminaux :
  - `/etc/ttys`
  - `/etc/ttytab`

init System V

- Configuration dans `/etc/inittab` :

  `label:niveaux:action:commande`
- Action :
  - `respawn` : relancé par init après terminaison
  - `wait` : lancement et attente
  - `once` : lancement une seule fois
  - `boot` : lancement lors de la première lecture de `/etc/inittab`
  - `bootwait` : idem et attente
  - `off`
  - `initdefault` : niveau d'exécution par défaut
  - `sysinit` : lancement avant l'accès à la console
  - `powerfail` : coupure d'alimentation

Niveaux de init System V

- S, s : mono-utilisateur
- 0 : arrêt
- 1 : mono-utilisateur, administration système
- 2 : multi-utilisateurs, réseau non configuré
- 3 : multi-utilisateurs, réseau configuré
- 4 : non utilisé
- 5 : idem 3 + serveur X (arrêt sur systèmes plus anciens)
- 6 : arrêt et redémarrage

Exemple de fichier /etc/inittab

```
is:3:initdefault:
p3:s1235:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/console 2>&1
s0:0:wait:/sbin/rc0           >/dev/console 2>&1
s1:1:wait:/usr/sbin/shutdown -y -iS -g 0 >/dev/console 2>&1
s2:23:wait:/sbin/rc2          >/dev/console 2>&1
s3:3:wait:/sbin/rc3           >/dev/console 2>&1
s5:5:wait:/sbin/rc5           >/dev/console 2>&1
s6:6:wait:/sbin/rc6           >/dev/console 2>&1
fw:0:wait:/sbin/uadmin 2 0    >/dev/console 2>&1
of:5:wait:/sbin/uadmin 2 6    >/dev/console 2>&1
rb:6:wait:/sbin/uadmin 2 1    >/dev/console 2>&1
co:235:respawn:/usr/lib/saf/ttymon -g -h -p ...
```

Rôle des scripts de démarrage

- Positionnement du nom de la machine
- Vérification des systèmes de fichiers (`fsck`)
- Montage des systèmes de fichiers
- Configuration des interfaces réseau
- Activation du swap
- Sauvegarde des sessions des éditeurs
- Démarrage des 'démons'

Scripts de démarrage

- Sous BSD :
  - `/etc/rc`
  - `/etc/rc.local`
- Sous System V :
  - définis par `/etc/inittab`
  - fréquemment situés dans le répertoire `/etc/init.d` ou `/sbin/init.d`
  - liens depuis les répertoires `/etc/rc`*N*`.d` ou `/sbin/rc`*N*`.d`

Scripts de démarrage System V (1)

— Les scripts `/etc/rc`*N* ou `/sbin/rc`*N* exécutent les scripts situés dans `/etc/rc`*N*`.d` en séquence

— Exécution des scripts K* lors de la sortie d'un niveau (avec le paramètre 'stop') :

```
for f in /etc/rcN.d/K*; do
   if [ -s $f ]; then
        sh $f stop
   fi
done
```

— Puis exécution des scripts S* lors de l'entrée dans un niveau (avec le paramètre 'start') :

```
for f in /etc/rcN.d/S*; do
   if [ -s $f ]; then
        sh $f start
   fi
done
```

Scripts de démarrage System V (2)

- Exemple : lancement de `xntp` (eXtended Network Time Protocol) en niveau 3 et 5
- Dans `/etc/rc.d/init.d`, script de référence `xntpd`
- Dans `/etc/rc.d/rc`*N*`.d`, liens symboliques pour le lancement et l'arrêt :
  - Démarrage en niveau 3 :

    `/etc/rc.d/rc3.d/S80xntpd` → `../init.d/xntpd`
  - Démarrage en niveau 5 :

    `/etc/rc.d/rc5.d/S80xntpd` → `../init.d/xntpd`
  - Arrêt en niveau 0 :

    `/etc/rc.d/rc0.d/K20xntpd` → `../init.d/xntpd`
  - Arrêt en niveau 6 :

    `/etc/rc.d/rc6.d/K20xntpd` → `../init.d/xntpd`
- Démarrage manuel :
  - `/etc/init.d/xntpd start / stop`
  - `service xntpd start / stop`

Ajout de tâches spécifiques au démarrage

- Sous BSD, ajout dans `/etc/rc.local` :

```
if [ -f /usr/local/sbin/serveur ]; then
    /usr/local/sbin/serveur
   echo 'serveur started'
fi
```

- FreeBSD gère également les répertoires :
  - `/usr/local/etc/rc.d`
  - `/usr/X11R6/etc/rc.d`
- Sous System V :
  - ajout dans un script existant
  - création d'un nouveau script et modification de `/etc/inittab`
  - création d'un script dans `/etc/init.d` et liens dans `/etc/rc`*N*`.d`
- Sous Linux :
  - Programme `chkconfig`, qui positionne automatiquement les liens dans `/etc/rc`*N*`.d`, avec un numéro d'ordre pris dans le script.
  - Programme `update-rc.d`, pour Debian etc.

Configuration des services

- Certains systèmes regroupent la configuration des services lancés au démarrage dans un ou plusieurs fichiers
- Exemples :
  - FreeBSD : `/etc/rc.conf`, `/etc/defaults/rc.conf`
  - HP-UX : `/etc/rc.config.d/*`
  - Linux (Red Hat) : `/etc/sysconfig/*`
  - Linux (Debian) : `/etc/default/*`

Configuration des services - exemples

— FreeBSD : `/etc/rc.conf`

```
hostname="atlas.ens.uvsq.fr"
ifconfig_fxp0="inet 193.51.26.1  netmask 255.255.255.0"
defaultrouter="193.51.26.254"
syslogd_enable="YES"
inetd_enable="YES"
inetd_flags="-l"
named_enable="YES"
named_flags="-b /etc/namedb/named.conf"
...
```

— RedHat Linux : `/etc/sysconfig/network`

```
NETWORKING=yes
HOSTNAME=bigdaddy.csi.uvsq.fr
DOMAINNAME=csi.uvsq.fr
GATEWAY=193.51.26.254
GATEWAYDEV=eth0
```

Démarrage + configuration (Linux)

— Script de démarrage `/etc/init.d/xntpd` (*très* simplifié ...)

```
#!/bin/sh

# Provides:          xntp
# Required-Start:    network
# Required-Stop:     network

# chkconfig 35 80 20

. /etc/sysconfig/xntp

case $1 in
  start)
    /usr/sbin/xntpd -h $NTPSERVER
  stop)
    killall xntp
esac
```

— Script de configuration `/etc/sysconfig/xntpd`

```
NTPSERVER=coucou.uvsq.fr
```

Initramfs

- Le problème
  - Beaucoup d'inconnus pour une distribution
  - Quel système de fichiers ? Quel RAID ?
  - Comment monter un système de fichiers exotique ?
  - Implémenter tous les besoins dans le noyau : explosion de la taille et des combinaisons
  - Les distributions grand public ne prennent pas tout en charge ...

Initramfs

- Objectif
  - Initramfs s'intercale entre le noyau et le système d'init
  - Monte un système de fichiers racine
  - Bien plus facile à modifier en espace utilisateur
- Sous le capot
  - Archive compressé au format cpio
  - Mini système Linux chargé en mémoire
  - Contient une arborescence de fichiers
  - ... et les modules noyaux nécessaire

Systemd

- Principes
  - Remplacement progressif du démon init, au grand dam de certains
  - Meilleure gestion des dépendances
  - Chargement parallèle
  - Support des snapshots et leurs restaurations
  - Projet démarré en 2010
- Démocratisation
  - Fedora, Debian...

Systemd

- Compatible avec les scripts d'init SysV, peut remplacer sysvinit
- Utilisation de D-Bus pour démarrer les services
- Utilise cgroups
- Exemple de script, nestor.service :

```
[Unit]
Description=nestor Service

[Service]
ExecStart=/opt/nestor_rpi_engine/server_nestor/bin/server_nestor 8000
Restart=on-abort

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

- myenv.conf :

```
[Service]
Environment="SERVER_NESTOR_ROOT_DIR=/opt/nestor_rpi_engine/server_nesto
```

D-Bus et cgroups

- D-Bus
  - Communication inter processus
  - Peut démarrer un service suite à un évènement
  - Devenu commun suite à l'adoption de systemd
  - Utilisé par Pidgin, Nautilus, ...
- Cgroups
  - Fonctionnalité du noyau
  - Limite l'utilisation des ressources (cpu, mémoire, disque dur, ...)

Arrêt du système

- Commandes particulières :
  - arrêt des processus utilisateur
  - arrêt des démons
  - réécriture du buffer cache
  - démontage des systèmes de fichiers
  - arrêt ou redémarrage

Arrêt du système sous BSD

- Arrêt brutal : `halt`
- Redémarrage brutal : `reboot`
- Arrêt ou redémarrage propre : `shutdown`
  - `+minutes` ou `now`
  - `-h` pour arréter
  - `-r` pour redémarrer
  - par défaut : passage en mode mono-utilisateur
- Redémarrage rapide : `shutdown -f`, `fasthalt`, `fastboot`
- Passage brutal en mode mono-utilisateur : `kill -TERM 1`

Arrêt du système sous System V

- Changement de niveau : `telinit` *`niveau`* ou (brutal) `init` *`niveau`*
- Arrêt brutal : `telinit 0`
- Redémarrage brutal : `telinit 6`
- Arrêt ou redémarrage propre : `shutdown`
  - `-g`*`secondes`* (*`secondes`* sous HP-UX)
  - `-i0` (`-h` sous HP-UX) pour arrêter
  - `-i6` (`-r` sous HP-UX) pour redémarrer
  - `-iS` pour passer en mode mono-utilisateur
  - `-y` pour éviter une demande de confirmation

LA GESTION DES TERMINAUX

See, you not only have to be a good coder to create a system like Linux, you have to be a sneaky bastard too.

Linus Torvalds

Gestion des terminaux

- Deux types de terminaux :
  - terminaux connectés directement au système
  - terminaux virtuels, utilisés par X-Window et les connexions par réseau
- Terminaux connectés :
  - activation d'un processus de connexion
  - mise à disponibilité des informations de contrôle
- Terminaux virtuels :
  - mise à disponibilité des informations de contrôle

Détail d'une connexion

- Lecture du nom d'utilisateur par `getty`
- Exécution du programme `login` par `getty`
- Lecture du mot de passe et validation par `login`
- Vérification du terminal de login dans `/etc/securetty` en cas de login root
- Vérification de la présence du fichier `/etc/nologin`
- Affichage de `/etc/motd` par `login` et blocage des logins le cas échéant
- Positionnement de la variable d'environnement `TERM` et exécution du shell par `login`
- Exécution des fichiers d'initialisation par le shell

Fichiers de configuration

Système	**Activation** (où)	**Type de terminal** (pour chaque `/dev/ttyXX`)	**Paramètres** (version simplifiée du termcap, pour getty etc.)	**Gestionnaire** (programme)
HP-UX	inittab	ttytype	gettydefs	getty
IRIX	inittab	ttytype	gettydefs	getty
OSF/1	inittab	inittab	gettydefs	getty
Linux	inittab	ttytype	-	mingetty
*BSD	ttys	ttys	gettytab	getty
SunOS	ttytab	ttytab	gettytab	getty
Solaris	_sactab	_sactab	_pmtab	ttymon

/etc/ttytab et /etc/ttys

- Systèmes basés sur 4.3BSD
- Description des terminaux directement connectés :
  - port
  - type du terminal
  - programme à exécuter
- Syntaxe de chaque ligne :

```
port    programme type_terminal on|off [secure]
```

- Exemple (`/etc/ttytab` sous SunOS) :

```
console '/usr/etc/getty std.9600'  sun     on secure
ttya    '/usr/etc/getty std.19200' vt100   on
ttyb    '/usr/etc/getty std.9600'  unknown off
```

- Après un changement de configuration : `kill -HUP 1`

/etc/gettytab

- Définition des informations de contrôle des ports
- Association de noms symboliques à la configuration des ports
- Syntaxe similaire à `printcap` et `termcap`
- Exemple :

```
default:\
   :lm=\r\n%h login\72 :sp#9600:

2|std.9600|9600-baud:\
   :sp#9600:
h|std.19200|19200-baud:\
   :sp#19200:
```

/etc/inittab

- Systèmes basés sur System V
- Le programme `getty` peut être lancé par `init`
- Exemple :

```
co:234:respawn:/etc/getty console console
t1:234:respawn:/etc/getty ttyS1 19200
t2:234:off:/etc/getty ttyS2 9600
```

/etc/ttytype

— Fichier décrivant le type des terminaux connectés

— Syntaxe de chaque ligne :

```
type_terminal  port
```

— Exemple :

```
wyse    console
dialup  ttyi1
dialup  ttyi2
vt320   ttyi2
```

/etc/gettydefs

- Rôle indentique à `gettytab` ...
- ... mais syntaxe différente
- Syntaxe :

```
label# initialisation# terminaison# message# suivant
```

- Exemple :

```
console# B9600 HUPCL # B9600 SANE #login: #console
19200# B19200 HUPCL # B19200 SANE #login: #9600
9600# B9600 HUPCL # B9600 SANE #login: #4800
9600# B9600 HUPCL # B9600 SANE #login: #4800
```

- Après modification :

```
getty -c gettydefs
```

Terminaux sous Solaris 2 (1)

— Gestion absolument différente de celle de tous les autres systèmes !

— 'Service Access Facility'

— Lancement du démon dans `/etc/inittab` :

```
sc:234:respawn:/usr/lib/saf/sac -t 300
```

— Lancement du gestionnaire de la console dans `/etc/inittab` :

```
co:234:respawn:/usr/lib/saf/ttymon -g -h \
     -p "`uname -n` console login: " -T sun \
     -d /dev/console -l console -m ldterm,ttcompat
```

Terminaux sous Solaris 2 (2)

- Plusieurs commandes d'administration :
  - `sacadm` : ajout, suppression, activation, désactivation de gestionnaires de ports
  - `pmadm` : configuration de gestionnaires de ports
  - `ttyadm` : configuration des ports série
  - `sttydefs` : création et modification d'entrées dans `/etc/ttydefs`
- Fichier `/etc/ttydefs` : similaire à `gettydefs`

```
9600f:9600 crtscts hupcl:9600 crtscts hupcl::9600f
38400:38400 hupcl:38400 hupcl::19200
19200f:19200 hupcl:19200 hupcl::9600
9600:9600 hupcl:9600 hupcl::4800
```

Terminaux sous Linux

— 8 consoles virtuelles :
  — 6 consoles texte

    création par `init` :

```
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty --noclear tty2
3:2345:respawn:/sbin/mingetty --noclear tty3
4:2345:respawn:/sbin/mingetty --noclear tty4
5:2345:respawn:/sbin/mingetty --noclear tty5
6:2345:respawn:/sbin/mingetty --noclear tty6
```

    accessibles par {Ctrl}+{F1 ... F6} depuis une autre console texte, ou {Ctrl}+{Alt}+{F1 ... F6} depuis une console X

  — 2 consoles X

    Crées lors du lancement de X

    accessibles par {Ctrl}+{F7 ... F8} depuis une console texte, ou {Ctrl}+{Alt}+{F7 ... F8} depuis une autre console X

Paramètres des terminaux

- Deux bases de données définissent les paramètres des terminaux :
  - `/etc/termcap` sous BSD
  - `/usr/lib/terminfo` (ou `/usr/share/terminfo` ou `/usr/share/lib/terminfo`) sous System V
- Ces bases définissent :
  - les paramètres des terminaux (nombre de lignes, de colonnes, ...)
  - les caractères de contrôle

/etc/termcap

— Suite d'entrées

— Syntaxe d'une entrée :

```
nom1|nom2...:paramètres
```

— Syntaxe similaire à `printcap`

— Exemple :

```
d0|vt100|dec vt100:\
    :co#80:li#24:ho=\E[H:\
    :ku=\EOA:kd=\EOB:
```

terminfo

- Ensemble de fichiers binaires décrivant des terminaux
- Chaque entrée est un fichier situé dans le répertoire de la base (≈ `/usr/share/terminfo`)
- Compilation d'un fichier de description : `tic`
- Décompilation d'une entrée compilée : `infocmp`
- Exemple de description (`infocmp /usr/share/terminfo/v/vt100`) :

```
vt100|dec vt100,
   cols#80, lines#24, home=\E[H,
   kcuul=\EOA, kdcudl=\EOB
```

Caractéristiques du terminal

- Un utilisateur peut utiliser `stty` pour modifier les caractéristiques de son terminal
- Syntaxe : `stty option [valeur] ...`
- Principales options :
  - `-a` : affiche les paramètres courants
  - *N* : vitesse de la ligne
  - `rows` *N* : nombre de colonnes
  - `lines` *N* : nombre de lignes
  - `erase` *C* : caractère d'effacement de caractère
  - `intr` *C* : caractère d'interruption
  - `susp` *C* : caractère de suspension
  - `oddp`, `evenp` : parité
  - `sane` : réinitialisation des paramètres (voir aussi commande `reset`)

X, configuration (1)

— Sur PC : XFree 86, ou maintenant Xorg (potentiellement replacé dans le futur par wayland)

— Fichier de configuration : `/etc/XF86config`, `/etc/xorg.conf`

— Paramètres de la carte graphique :

```
Section "Device"
        Identifier "Matrox Millennium G400"
        Driver "mga"
        BoardName "Unknown"
EndSection
```

Signification des principaux paramètres :

- `Identifier` : clef pour identifier la carte utilisée pour l'affichage
- `Driver` : identifie le driver utilisé, ici `/lib/modules/2.4.16/kernel/drivers/char/drm/mga.o` et `/usr/X11R6/lib/modules/drivers/mga_drv.o`

X, configuration (2)

— Paramètres de l'écran :

```
Section "Monitor"
        Identifier "COMPAQ TFT7000"
        VendorName "Unknown"
        ModelName  "Unknown"
        HorizSync 31.0 - 80.0
        VertRefresh 58 - 85
        Option "dpms"
EndSection
```

Signification des principaux paramètres :

- `Identifier` : clef pour identifier l'écran utilisé pour l'affichage
- `HorizSync` : fréquences admissibles pour le balayage horizontal, en kHz.
- `VertRefresh` : fréquences admissibles pour le balayage vertical, en Hz.

Ces deux informations sont à trouver dans la documentation constructeur, ou sur le web, ou par l'expérience. . .

X, configuration (3)

— Paramètres de l'affichage (association de la carte graphique et de l'écran) :

```
Section "Screen"
        Identifier "Screen0"
        Device "Matrox Millennium G400"
        Monitor "COMPAQ TFT7000"
        DefaultDepth 24
        Subsection "Display"
                Depth 24
                Modes "1280x1024" "1152x864" "1024x768" "800x600"
        EndSubSection
EndSection
....
Section "ServerLayout"
        ....
        Screen      0  "Screen0" 0 0
```

- `Identifier` : clé pour identifier l'affichage utilisé composé de l'association de la carte graphique ("`Device`") et de l'écran ("`Monitor`")
- `DefaultDepth` : nombre de plans par défaut pour cet affichage (8, ou maintenant le plus souvent 24 bits).
- `Depth` + `Modes` : résolutions admissibles pour cet affichage (il est possible de passer d'une résolution à l'autre en faisant {`Ctrl`}+{`Alt`}+{`+`} et {`Ctrl`}+{`Alt`}+{`-`})

X, lancement

— Lancement du serveur

Lancement par la commande `xinit`.

Souvent encapsulé : `startx`, `x11`

Syntaxe : `xinit $clientargs -- $serverargs`

— `$clientargs` : les paramètres sont lus dans le fichier `~/.xinitrc`, ou dans `/etc/X11/xinit/xinitrc` en son absence ;

— `$serverargs` : les paramètres sont lus dans le fichier `~/.xserverrc`, ou dans `/etc/X11/xinit/xserverrc` en son absence.

— Lancement des clients

lancement directement ou via fichier `xinitrc`, soit à partir de fichiers `~/.Xclients` ou `/etc/X11/xinit/Xclients`. Par exemple :

— un xterm en mode console (`xterm -C`)

— lecture des ressources X : `xrdb ~/.Xdefaults`

— mapping clavier : `xmodmap ~/.Xmodmap`

— `xclock`, `xbiff`, `xeyes`...

— Lancement d'un Window Manager (`fvwm`, `kde`),...

RÉSEAU ET SERVICES RÉSEAU

If you want to travel around the world and be invited to speak at a lot of different places, just write a Unix operating system.

Linus Torvalds

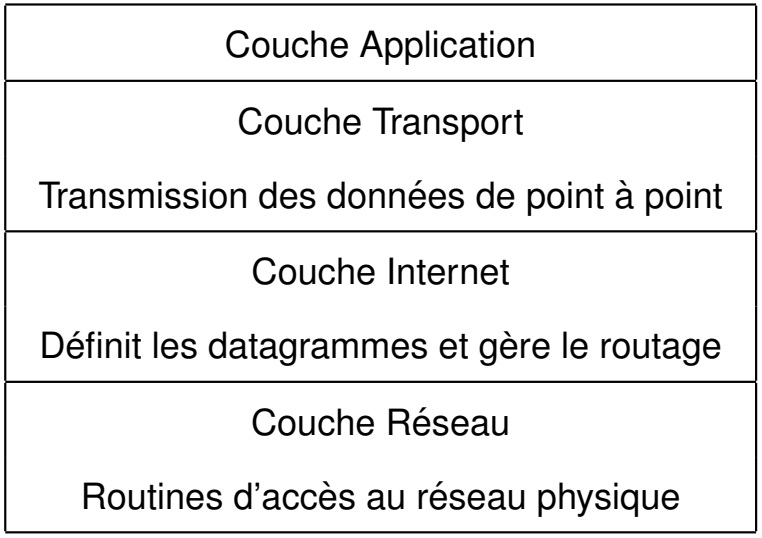
Configuration réseau

- Unix inclut le support du protocole TCP/IP
- La configuration réseau consiste à :
  - configurer les interfaces réseau ;
  - configurer le routage ;
  - configurer les services réseau fournis par le système ;
  - configurer les services réseau utilisés par le système.

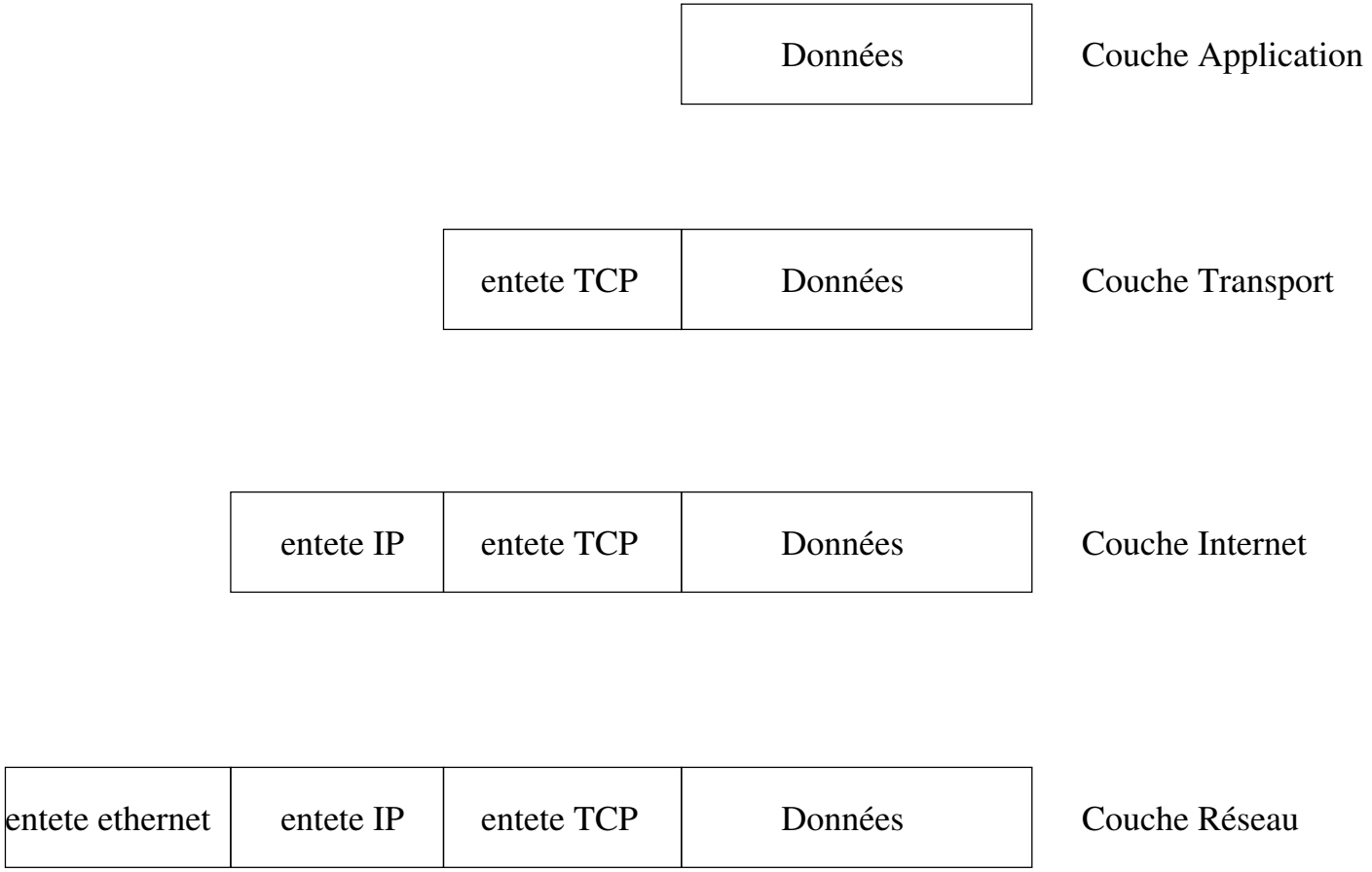
Modèle de référence OSI

Couche Application
Couche Présentation Standardisation de la représentation des données
Couche Session Gestion des sessions entre applications
Couche Transport Détection et correction des erreurs
Couche Réseau Gestion des connexions sur le réseau
Couche Liaison Transmission fiable des données sur le lien physique
Couche Physique Définit les caractéristiques du réseau physique

Architecture de TCP/IP



Encapsulation



Classes d'adresses IP (1)

- Format des adresses IP :
  - 4 octets ;
  - numéro de réseau et numéro de machine
- Quatre classes d'adresses IP
- Classes A : premier bit à `0` ; 8 bits pour le numéro de réseau, 24 bits pour le numéro de machine ;
- Classes B : deux premiers bits à `1 0` ; 16 bits pour le numéro de réseau, 16 bits pour le numéro de machine ;
- Classes C : trois premiers bits à `1 1 0` ; 24 bits pour le numéro de réseau, 8 bits pour le numéro de machine ;
- Classes D (adresses *multicast*) : quatre premiers bits à `1 1 1 0` ;
- Classes E (réservées) : cinq premiers bits à `1 1 1 1 0`.

Classes d'adresses IP (2)

- Réseau 0 : route par défaut
- Réseaux 1 à 126 : classes A (16777214 machines maximum)
- Réseau 127 : adresse de *loopback*
- Réseaux 128 à 191 : classes B (65534 machines maximum)
- Réseaux 192 à 223 : classes C (254 machines maximum)
- Numéros de machines réservés : `0` : adresse du réseau, `255` : adresse de diffusion
- Exemples :
  - `26.104.0.19` : machine `104.0.19` dans le réseau `26`
  - `132.227.60.30` : machine `60.30` dans le réseau `132.227`
  - `193.51.26.14` : machine `14` dans le réseau `193.51.26`

Notion de sous-réseau

- La structure standard d'une adresse IP peut être modifiée localement
- Utilisation d'une partie de l'adresse de machine comme numéro de sous-réseau
- Création de sous-réseau pour résoudre des problèmes :
  - topologiques : utilisation de plusieurs réseaux physiques pour la même classe ;
  - organisationnels : administration déléguée.
- Création de sous-réseau en appliquant un masque de réseau :
  - si un bit est à `1`, le bit correspondant dans l'adresse fait partie de l'adresse de réseau ;
  - si un bit est à `0`, le bit correspondant dans l'adresse fait partie de l'adresse de machine.

Masque de sous-réseau

- Le masque de sous-réseau définit les bits à prendre en compte dans l'adresse du réseau
- Son interprétation est locale
- Il peut être exprimé sous forme de :
  - une suite de bits ;
  - quatre octets ;
  - un nombre de bits consécutifs.
- Masques standards :
  - classes A : `255.0.0.0`
  - classes B : `255.255.0.0`
  - classes C : `255.255.255.0`

Exemples

- `132.227.60.30/24` (masque `255.255.255.0`) :
  - réseau `132.227`
  - sous-réseau `60`
  - machine `30`
- `134.157.0.129/25` (masque `255.255.255.128`) :
  - réseau `134.157`
  - sous-réseau `1`
  - machine `1`
- `193.51.24.74/27` (masque `255.255.255.224`) :
  - réseau `193.51.24`
  - sous-réseau `2`
  - machine `10`

Réseaux privés

- Le RFC (Request For Comment) 1597 définit plusieurs réseaux privés
- Un réseau privé peut être utilisé à l'intérieur d'une organisation ...
- ... mais il n'est pas accessible de l'extérieur
- Réseaux privés définis :
  - classe A : `10.0.0.0`
  - classe B : `172.16.0.0`
  - classes C : `192.168.0.0`

NAT (Network Address Translation)

— Correspondance adresses IP internes / externes

— Ralentit la diminution des adresses IPv4 disponibles

— Configuration avec iptable :

```
iptables -F INPUT ; iptables -P INPUT ACCEPT

iptables -F OUTPUT ; iptables -P OUTPUT ACCEPT

iptables -F FORWARD ; iptables -P FORWARD ACCEPT

iptables -t nat -F PREROUTING

iptables -t nat -A PREROUTING -d 195.115.19.35/32 -j DNAT –to-destination 172.16.0.1/32
```

Interfaces réseau

- Une adresse IP peut être affectée à chaque interface réseau
- Nom des interfaces réseau :
  - Solaris : `le0`
  - HP-UX : `lan0`
  - OSF/1 : `ln0`, `nu0`
  - Linux : `eth0`
  - FreeBSD : nom spécifique à la cartes (exemples : `fxp0`, `xl0`)
- Interfaces spécifiques :
  - Interface “loopback” : `lo0`
  - Point à point : `ppp0`
- Le noyau doit contenir :
  - le(s) pilote(s) de la(les) interface(s) réseau ;
  - le support des protocoles réseau

Configuration d'une interface réseau (1)

- `ifconfig` *interface* `[options]`
- Options :
  - `[inet]` *adresse*
  - `netmask` *masque*
  - `broadcast` *adresse*
  - `up`
  - `down`
- Exemple :

```
ifconfig fxp0 inet 193.51.24.1 netmask 255.255.255.224 \
broadcast 193.51.24.31
```

Configuration d'une interface réseau (2)

- Les interfaces réseau sont généralement configurées lors du démarrage du système par les scripts d'initialisation
- Les scripts utilisent des fichiers de configuration :
  - Linux Red Hat :
    `/etc/sysconfig/network`
    `/etc/sysconfig/network-scripts/ifcfg-*`
  - Linux Debian : `/etc/network/interfaces`
  - FreeBSD : `/etc/rc.conf`
  - SunOS, Solaris : `/etc/hostname.`*interface*
  - HP-UX : `/etc/rc.config.d/netconf`

Exemples (1)

- Linux Red Hat :
  - `/etc/sysconfig/network`

```
NETWORKING=yes
FORWARD_IPV4=false
HOSTNAME=bigdaddy.csi.uvsq.fr
DOMAINNAME=csi.uvsq.fr
...
```

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

```
DEVICE=eth0
IPADDR=193.51.26.14
NETMASK=255.255.255.0
NETWORK=193.51.26.0
BROADCAST=193.51.26.255
GATEWAY=193.51.33.254
ONBOOT=yes
PROTO=static
```

ou

```
PROTO=dhcp
```

Exemples (2)

— Linux Debian :

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 193.51.26.14
    network 193.51.26.0
    netmask 255.255.255.0
    broadcast 193.51.26.255
    gateway 193.51.33.254
```

ou

```
auto eth0
iface eth0 inet dhcp
```

Exemples (3)

- FreeBSD :
  - /etc/defaults/rc.conf

```
network_interfaces="lo0"          # List of network interfaces
ifconfig_lo0="inet 127.0.0.1"     # loopback device configuration.
```

  - /etc/rc.conf

```
network_interfaces="fxp0 lo0"     # List of network interfaces
ifconfig_fxp0="inet 193.51.24.1  netmask 255.255.255.224"
```

Exemples (4)

— HP-UX :

`/etc/rc.config.d/netconf`

```
HOSTNAME="romuald.isty-info.uvsq.fr"
OPERATING_SYSTEM=HP-UX
LOOPBACK_ADDRESS=127.0.0.1
...
INTERFACE_NAME[0]=lan0
IP_ADDRESS[0]=193.51.33.1
SUBNET_MASK[0]=255.255.255.0
BROADCAST_ADDRESS[0]=""
LANCONFIG_ARGS[0]="ether"
DHCP_ENABLE[0]=0
```

État des interfaces réseau (1)

`ifconfig` *interface*

Exemples :

— FreeBSD :

```
$ /sbin/ifconfig -a
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 193.51.24.1 netmask 0xfffffe0 broadcast 193.51.24.31
        ether 00:a0:c9:ee:76:58
        media: autoselect (100baseTX <full-duplex>) status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet 127.0.0.1 netmask 0xff000000
```

— HP-UX :

```
$ /usr/sbin/ifconfig lan0
lan0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
        inet 193.51.33.1 netmask ffffff00 broadcast 193.51.33.255
```

État des interfaces réseau (2)

```
netstat -i [options]
```

Exemple :

```
$ netstat -in
Name Mtu   Network       Address              Ipkts Ierrs  Opkts Oerrs  Coll
fxp0 1500  <Link>      00.a0.c9.ee.76.58 599269     1 614046     0     0
fxp0 1500  193.51.24/27 193.51.24.1      599269     1 614046     0     0
lo0  16384 <Link>                         46738     0  46738     0     0
lo0  16384 127           127.0.0.1        46738     0  46738     0     0
```

Test de connectivité

- La commande `ping` peut être utilisée pour tester la connectivité
- Elle utilise le protocole ICMP (Internet Control Message Protocol) pour :
  - envoyer une requête “echo”
  - recevoir la réponse
- Exemple :

```
$ ping soleil.uvsq.fr
PING soleil.uvsq.fr (193.51.24.1): 56 data bytes
64 bytes from 193.51.24.1: icmp_seq=0 ttl=254 time=0.638 ms
64 bytes from 193.51.24.1: icmp_seq=1 ttl=254 time=0.634 ms
^C
--- soleil.uvsq.fr ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.524/0.557/0.603/0.034 ms
```

- Aussi :
  - `ping -c 1` : une fois seulement (pour scripts)
  - `ping -f` : mode flood, pour “tester” une liaison réseau
  - `ping -b 193.51.0.0` : mode broadcast

Adresses MAC

- Au niveau de la couche Réseau, les communications sont assurées en utilisant les adresses MAC
- L'adresse MAC correspondant à une adresse IP est obtenue par le protocole ARP (Address Resolution Protocol) :
  - le système qui veut dialoguer avec un autre diffuse une requête "qui a cette adresse IP ?"
  - la machine concernée lui répond
- Les correspondances entre adresses IP et MAC sont maintenues dans un cache

Manipulation du cache ARP

— Liste : `arp -a [options]`

```
$ /usr/sbin/arp -an
 (193.51.33.32) at 0:0:a7:2:56:ee ether
 (193.51.33.2) at 8:0:9:c4:12:fc ether
 (193.51.33.5) at 8:0:9:e:41:8a ether
 (193.51.33.41) at 0:0:a7:3:78:24 ether
 (193.51.33.42) at 0:0:a7:3:78:41 ether
 (193.51.33.106) at (incomplete)
...
```

— Suppression : `arp -d` *nom/adresse-IP*

— Ajout :

`arp -s` *nom/adresse-IP adresse-MAC* `[option]`

`arp -f` *nom-de-fichier*

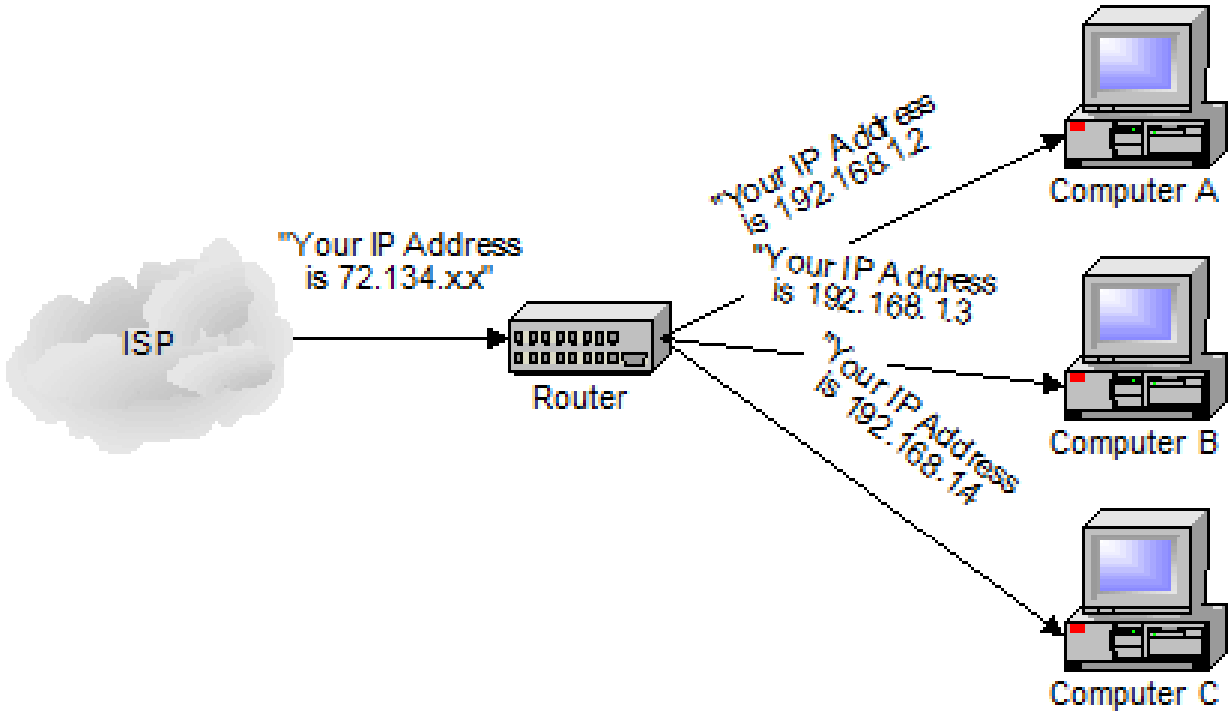
Quid du routeur

— 3 périphériques différents : routeur, hub et switch

— routeur : transfert des paquets via PLUSIEURS réseaux

— switch : transferts des paquets sur un même réseau, d'un port vers un autre

— hub : tout ce qui vient d'un port est transféré sur tous les autres



Routage IP

- Un système peut accéder directement aux machines connectées sur le même réseau (sous-réseau)
- Pour accéder aux machines situées sur un autre réseau, il doit disposer d'une table de routage
- Une table de routage est constituée d'entrées spécifiant :
  - l'adresse de destination
  - l'adresse du routeur à qui transmettre les paquets
- Plusieurs types de routages :
  - statique (`route`)
  - dynamique (`routed`, `gated`)

Table de routage

— La configuration d'une interface réseau crée une entrée permettant d'accéder au réseau local

— Les autres entrées doivent être configurées par l'administrateur

— Affichage de la table de routage : `netstat -r`

— Exemple simple :

```
$ netstat -rn
Routing tables
Destination   Gateway         Flags  Refs     Use  Interface  Pmtu
127.0.0.1     127.0.0.1       UH        0     119  lo0        4608
193.51.33.1   127.0.0.1       UH        5   87257  lo0        4608
default       193.51.33.254   UG       45 1917378  lan0       1500
193.51.33     193.51.33.1     U       110 3898293  lan0       1500
```

Exemple complexe

```
$ netstat -rn
Routing tables
Internet:
Destination     Gateway           Flags  Refs      Use    Netif Expire
default         193.51.24.30      UGSc     58  3409277    fxp0
127             127.0.0.1         URc       0        0    lo0
127.0.0.1       127.0.0.1         UH        1     4103    lo0
193.51.24/27    link#1            UC        0        0    fxp0
193.51.24.1     0:a0:c9:ee:76:58  UHLW      3  4567301    lo0
193.51.24.2     0:30:94:e2:be:a0  UHLW      2    16111    fxp0    521
...
193.51.24.64/27 193.51.24.30      UGc       0    12597    fxp0
193.51.25       193.51.24.2       UGc       2    14244    fxp0
193.51.26       193.51.24.3       UGc       2    15604    fxp0
193.51.27       193.51.24.30      UGc       0    20814    fxp0
193.51.28       193.51.24.30      UGc       1    18496    fxp0
193.51.29       193.51.24.9       UGc       0    10858    fxp0
```

Routage statique

— L'administrateur définit les entrées de la table de routage

— Ajout :

`route add` *`destination adresse metric`*

— Suppression :

`route delete` *`destination adresse`*

— Exemple :

```
# route add default 193.51.33.254 1
```

— Généralement, le routage par défaut est positionné par les scripts d'initialisation

— Rejet : possibilité d'une route pour rejeter les paquets venant d'une machine

`route add -host` *`adresse_ip`* `reject`

La route est alors affichée avec le flag `!H`.

Exemples

— Linux Red Hat : `/etc/sysconfig/network`

```
...
GATEWAY=193.51.26.254
GATEWAYDEV=eth0
...
```

— FreeBSD : `/etc/rc.conf`

```
...
defaultrouter="193.51.26.254" # Set to default gateway (or NO).
router_enable="NO"            # Set to YES to enable a routing daemon.
...
```

— HP-UX : `/etc/rc.config.d/netconf`

```
ROUTE_DESTINATION[0]=default
ROUTE_MASK[0]=""
ROUTE_GATEWAY[0]=193.51.33.254
ROUTE_COUNT[0]=1
```

Routage dynamique

— Dans un environnement complexe, la mise en œuvre du routage statique est souvent difficile

— La mise en place d'un mécanisme de routage dynamique permet de faciliter les mises à jour

— Chaque routeur diffuse la liste des réseaux sur lesquels il est connecté

— Chaque routeur met à jour sa table de routage à partir des informations reçues depuis les autres

— Démons de routage : `routed`, `gated`

Suivi du routage

— La commande `traceroute` permet de connaître le routage vers une destination

— Exemple :

```
$ traceroute ftp.lip6.fr
traceroute to nephtys.lip6.fr (195.83.118.1), 30 hops max, 20 byte pack
 1 r-isty-info.reseau.uvsq.fr (193.51.33.254)      3 ms    2 ms    1 ms
 2 r-uvsq.reseau.uvsq.fr (193.51.24.30)            2 ms    8 ms    6 ms
 3 195.83.240.221 (195.83.240.221)                14 ms   11 ms    8 ms
 4 boulogne1.rerif.ft.net (193.48.53.177)         13 ms   10 ms   14 ms
 5 stlambert1.rerif.ft.net (193.48.53.137)        12 ms   10 ms   21 ms
 6 stamand1.rerif.ft.net (193.48.53.101)          17 ms   11 ms   17 ms
 7 nio-i.cssi.renater.fr (193.51.206.145)         30 ms   28 ms   22 ms
 8 nio-n1.cssi.renater.fr (193.51.206.9)          26 ms   22 ms   36 ms
 9 jussieu.cssi.renater.fr (194.214.109.6)        29 ms   38 ms   17 ms
10 univ-jussieu.cssi.renater.fr (194.214.109.22)  14 ms   29 ms   18 ms
11 r-intercon.reseau.jussieu.fr (134.157.254.123) 14 ms   22 ms   17 ms
12 nephtys.lip6.fr (195.83.118.1)                 24 ms   28 ms   26 ms
```

Nommage des machines

- Les protocoles réseau utilisent des adresses IP pour désigner les machines
- Mais il est plus parlant d'utiliser des noms associés aux machines
- La correspondance entre adresses et noms peut être réalisée par :
  - `/etc/hosts` :

```
127.0.0.1       localhost
193.51.24.1     soleil.uvsq.fr soleil
193.51.24.5     lune.uvsq.fr lune
193.51.24.11    venus.uvsq.fr venus
193.51.24.15    pluton.uvsq.fr pluton
```

  - une base locale : Yellow Pages de Sun (`yp`), `netinfo`,...
  - le DNS

  Possibilités mentionnées dans le `/etc/nsswitch.conf`, par exemple :

```
hosts:files dns
```

Services réseau

- Les services réseau sont assurés par des serveurs (démons)
- Chaque serveur est en attente de connexion sur un port
- La liste des protocoles réseau est contenue dans `/etc/protocols`
- La liste des services (avec le(s) port(s) associé(s)) est contenue dans `/etc/services`
- Certains services sont lancés au démarrage du système (exemples : `named`, `nfsd`, `sendmail`)
- D'autres services sont lancés par le "serveur internet" `inetd`

/etc/protocols

```
ip      0    IP          # internet protocol v4, pseudo protocol number
icmp    1    ICMP        # internet control message protocol
igmp    2    IGMP        # internet group management protocol
ggp     3    GGP         # gateway-gateway protocol
tcp     6    TCP         # transmission control protocol
egp     8    EGP         # exterior gateway protocol
pup     12   PUP         # PARC universal packet protocol
udp     17   UDP         # user datagram protocol
hmp     20   HMP         # host monitoring protocol
xns-idp 22   XNS-IDP     # Xerox NS IDP
rdp     27   RDP         # reliable data protocol
iso-tp4 29   ISO-TP4     # ISO Transport Protocol Class 4
ipv6    41   IPv6        # Internet Protocol, version 6
esp     50   ESP         # IPSEC esp
ah      51   AH          # IPSEC ah
icmpv6  58   ICMPV6      # Internet Control Message Protocol version 6
iso-ip  80   ISO-IP      # ISO Internet Protocol
```

/etc/services

```
tcpmux      1/tcp                   # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp       sink null
discard     9/udp       sink null
systat      11/tcp      users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
...
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp      fspd
ssh         22/tcp                  # SSH Remote Login Protocol
ssh         22/udp                  # SSH Remote Login Protocol
telnet      23/tcp
...
```

Netcat

- Couteau suisse du TCP/IP
- Chat Client / Serveur

```
Serveur 1:
$ nc -lp 1337
Serveur 2:
$ telnet 127.0.0.1 1337
```

- Transfert de fichiers

```
Serveur 1:
$ nc -lp 1234 > monfichier.zip
Serveur 2:
$ nc -w 1 server1.example.com 1234 < monfichier.zip
```

Netcat (2)

— Scanneur de port

```
$ nc -v -w 1 monserveur.talbart.fr -z 1-8010
DNS fwd/rev mismatch: monserveur.ddns.net != 52.256.94.
talbart2.ddns.net [93.55.242.52] 8010 (?) : Connection
talbart2.ddns.net [93.55.242.52] 8004 (?) : Connection
...
talbart2.ddns.net [93.55.242.52] 8003 (?) : Connection
talbart2.ddns.net [93.55.242.52] 8002 (?) : Connection
talbart2.ddns.net [93.55.242.52] 8001 (?) open
talbart2.ddns.net [93.55.242.52] 8000 (?) open
talbart2.ddns.net [93.55.242.52] 7999 (?) : Connection
...
```

— Etc (création de backdoor)...

Nmap

— Scanneur de port (+ OS + uptime)

```
# nmap -O -v scanme.nmap.org

Starting Nmap ( http://nmap.org )
Nmap scan report for scanme.nmap.org (74.207.244.221)
Not shown: 994 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
646/tcp   filtered ldp
1720/tcp  filtered H.323/Q.931
9929/tcp  open     nping-echo
31337/tcp open     Elite
Device type: general purpose
```

```
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
OS details: Linux 2.6.39
Uptime guess: 1.674 days (since Fri Sep  9 12:03:04 2011)
Network Distance: 10 hops
TCP Sequence Prediction: Difficulty=205 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /usr/local/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.58 secon
           Raw packets sent: 1063 (47.432KB) | Rcvd: 103
```

Nouvelles commandes de remplacement

— arp : ip n (ip neighbor)

— ifconfig : ip a (ip addr), ip link, ip -s (ip -stats)

— netstat : ss, ip route (for netstat-r), ip -s link (for netstat -i), ip maddr (for netstat-g)

— route : ip r (ip route)

Nouvelles commandes de remplacement

```
portable-franck$ sudo ip rule
0:  from all lookup local
32766:  from all lookup main
32767:  from all lookup default

portable-franck$ sudo ip route list
default via 192.168.1.1 dev wlan0  proto static  metric 1024
169.254.0.0/16 dev wlan0  scope link  metric 1000
192.168.0.0/24 via 192.168.1.253 dev wlan0  proto dhcp  metric 10
192.168.1.0/24 dev wlan0  proto kernel  scope link  src 192.168.1.6

portable-franck$ sudo ip route list table main
default via 192.168.1.1 dev wlan0  proto static  metric 1024
169.254.0.0/16 dev wlan0  scope link  metric 1000
192.168.0.0/24 via 192.168.1.253 dev wlan0  proto dhcp  metric 10
192.168.1.0/24 dev wlan0  proto kernel  scope link  src 192.168.1.6
```

Le serveur inetd

- Rôle de `inetd`
  - `inetd` est démarré par les scripts d'initialisation
  - il lit le fichier `/etc/inetd.conf`
  - il se place en attente sur les ports spécifiés
  - lorsqu'une requête sur un port est reçue, `inetd` lance le serveur correspondant
- Format des lignes de `/etc/inetd.conf` :
  - nom de service
  - type de service (`stream` ou `dgram`)
  - nom de protocole (`tcp` ou `udp`)
  - attente ou non (`wait` ou `nowait`)
  - nom d'utilisateur
  - nom et arguments du serveur à lancer

Le serveur xinetd (1)

`xinetd` est le successeur de `inetd`, cumulant les fonctions de `inetd` (association du lancement d'un programme à une requète de service réseau), et de `tcp wrapper` (sécurisation par des autorisations).

— Configuration globale : fichier `/etc/xinetd.conf`, contenant des paramétrages communs à tous les services :

```
defaults
{
        instances                = 60
        log_type                 = SYSLOG authpriv
        log_on_success           = HOST PID
        log_on_failure           = HOST

        disable                 = yes
}

includedir /etc/xinetd.d
```

Le serveur xinet (2)

— Configuration de chaque service :
Répertoire `/etc/xinetd.d`, contenant un fichier de configuration par service ouvert, décrivant le programme à lancer et les paramètres de sécurité.
Exemple, `/etc/xinetd.d/telnet` :

```
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        only_from       = .uvsq.fr
        log_on_failure  += USERID
        disable         = no
}
```

Mise en place de NFS

- NFS : Network File System
- initialement développé par Sun
- implémenté sur la majorité des systèmes Unix
- Principe :
  - un serveur exporte une hiérarchie de fichiers
  - des clients accèdent à cette hiérarchie par une opération de montage
  - les accès aux fichiers sont transmis au serveur par le protocole NFS

Configuration d'un serveur NFS

- Plusieurs services :
  - `mountd` : serveur de montage
  - `nfsd` : serveur d'entrées/sorties
  - `lockd` : serveur de verrous
  - `statd` : serveur de surveillance
- Le fichier `/etc/exports` définit les hiérarchies exportées :
  `répertoire -[options]`
- La syntaxe de `/etc/exports` peut différer selon les systèmes
- Après modification de `/etc/exports` :
  - `exportfs -a`

    ou
  - `kill -HUP` *pid_de_mountd*
- Pour savoir ce qu'une machine exporte :
  `showmount -e` *machine*

Options d'exportation

- `ro` : exportation en lecture seule
- `rw=`*liste* : liste des clients autorisés à accéder en lecture/écriture
- `access=`*liste* : liste des clients autorisés
- `network=`*réseau* : accès autorisé à toutes les machines du réseau spécifié
- `alldirs` : tous les sous-répertoires de la hiérarchie sont exportés (montage possible)
- `anon=`*uid* : numéro d'utilisateur utilisé pour traiter les requêtes émanant d'un utilisateur non identifié (par défaut : `nobody`)
- `root=`*liste* : liste des clients autorisés à accéder aux fichiers avec un accès `root`
- `maproot=`*uid* : uid utilisé pour les accès effectués par `root`

Exemples

— HP-UX :

```
/users -anon=65534,async,access=athanase:paupiette:...,root=athanase
/public -anon=65534,async,access=athanase:paupiette:...,root=athanase
/var/mail -anon=65534,async,access=athanase:paupiette:...,root=athanase
```

— FreeBSD :

```
/usr    -maproot=root -alldirs -network=193.51.26
```

— Linux :

```
/public      athanase(rw,sync,no_root_squash) paupiette(ro,sync)
```

Configuration d'un client NFS

— L'accès à un répertoire distant est effectué via une opération de montage

— Une fois la hiérarchie montée, l'accès aux fichiers est transparent

— Montage manuel :

```
mount -t nfs -o rw,nosuid romuald.isty-info.uvsq.fr:/var/mail /var/mail
```

— Montage à chaque démarrage via mention dans `/etc/fstab` :

```
romuald.isty-info.uvsq.fr:/var/mail /var/mail nfs rw,nosuid 0 0
romuald.isty-info.uvsq.fr:/public /public nfs rw,nosuid 0 0
romuald.isty-info.uvsq.fr:/users /users nfs rw,nosuid 0 0
```

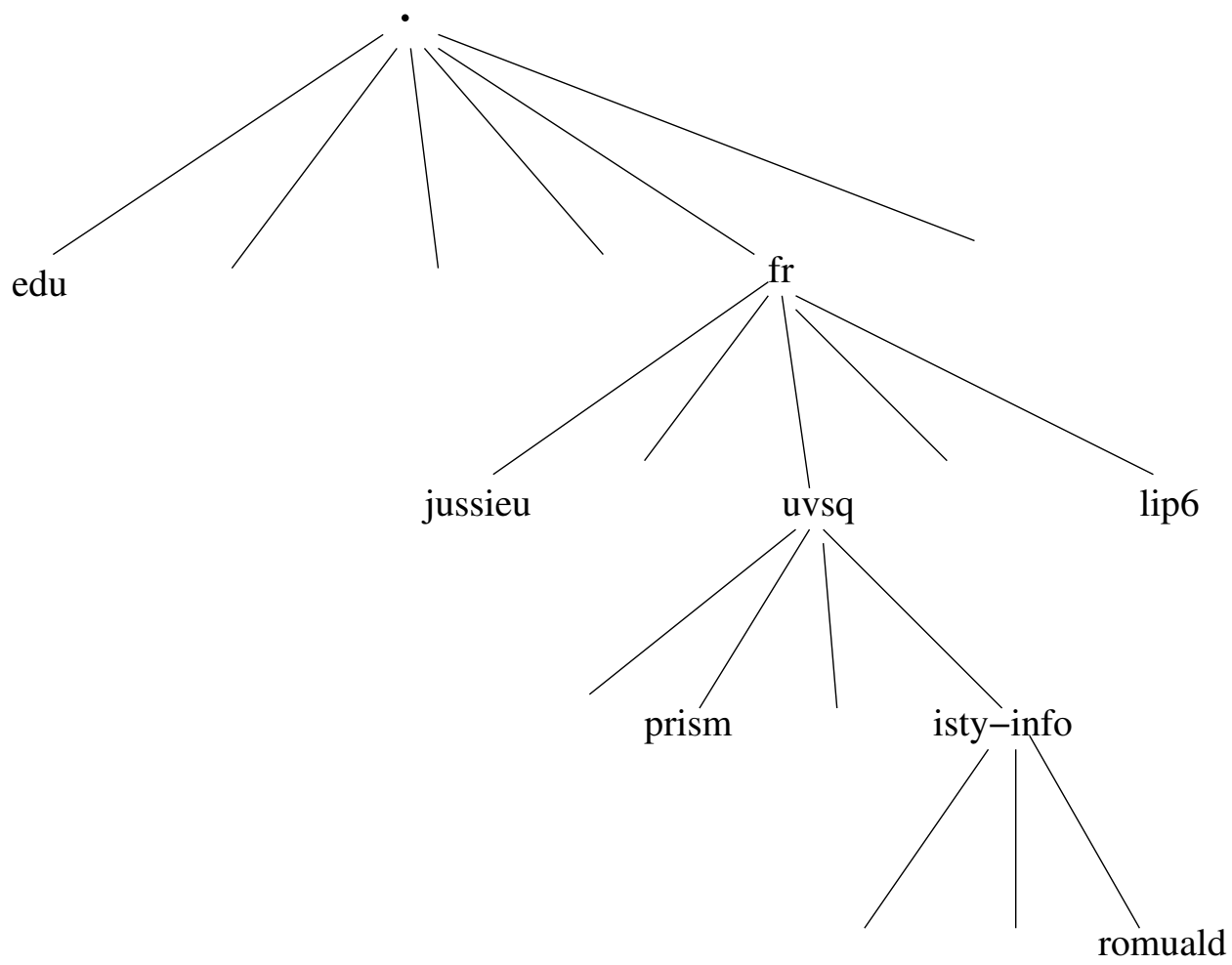
— Gestion de l'absence de réponse du serveur : simple renvoi d'erreur avec l'option `soft` (risques de corruption de donnée), blocage avec l'option `hard` (bloquant mais sûr, donc préférable).

— Montage automatique lors de l'accès au point de montage en utilisant `automount` (SunOS, ...), `autofs` (Linux, ...)

DNS - Principes

- Le DNS (Domain Name Server) est l'annuaire utilisé pour faire le lien entre des noms de machines et leurs adresses IP
- Le DNS utilise des noms qualifiés
- La base gérée par le DNS est :
  - répartie ;
  - hiérarchisée ;
  - avec une faible fréquence de changement ;
  - accessible en lecture seule.

DNS - Hiérarchie



Zones DNS

— Une zone représente un domaine (exemples : `fr`, `uvsq.fr`, `isty-info.uvsq.fr`)

— Une zone parente peut déléguer une zone fille à un ou plusieurs serveurs de noms

— Chaque zone est gérée par un serveur maître et plusieurs serveurs secondaires

— Le contenu de la zone n'est modifié que sur le serveur maître ; il est recopié sur les serveurs secondaires

— Tous les serveurs ont le même statut pour la consultation

— La recherche est hiérarchique

Recherche d'adresse

Recherche de l'adresse de `romuald.isty-info.uvsq.fr`

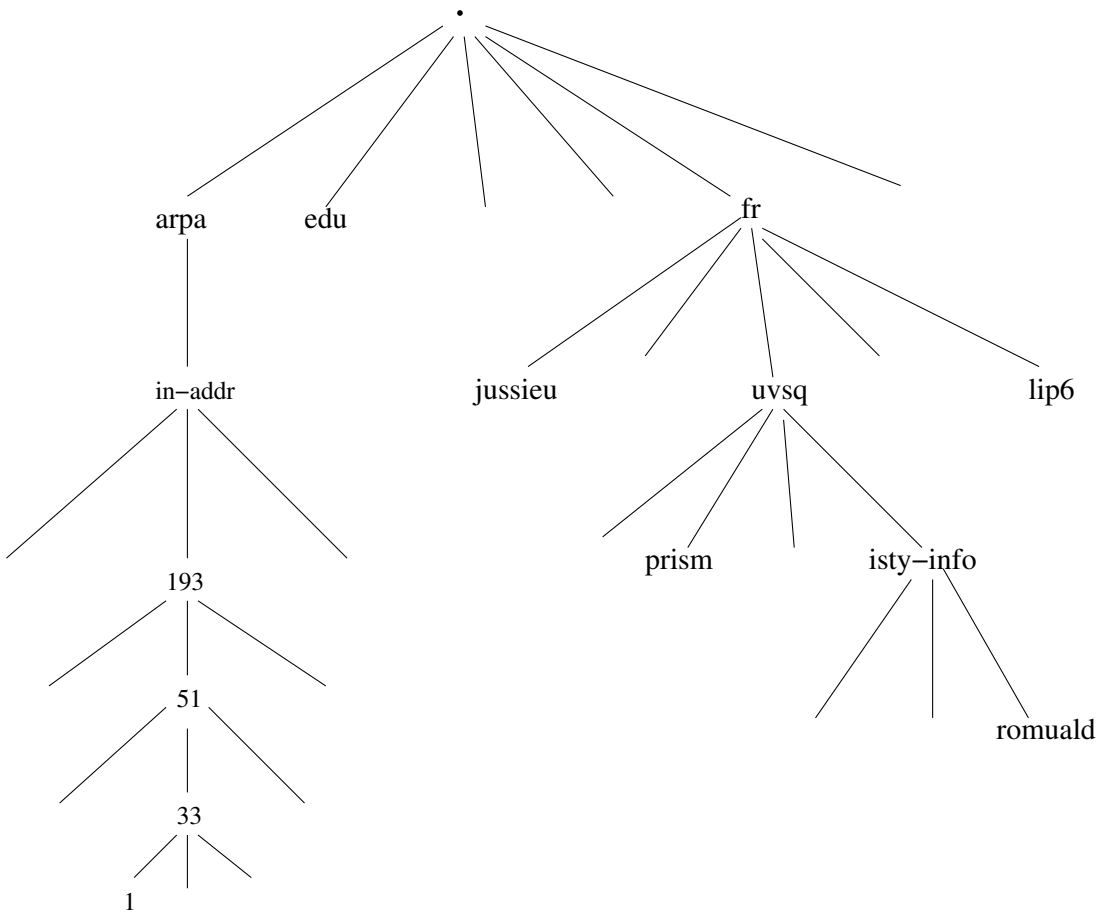
1. Demande aux serveurs de `.` les adresses des serveurs de noms de `fr`
2. Demande aux serveurs de `fr` les adresses des serveurs de noms de `uvsq.fr`
3. Demande aux serveurs de `uvsq.fr` les adresses des serveurs de noms de `isty-info.uvsq.fr`
4. Demande aux serveurs de `isty-info.uvsq.fr` l'adresse de `romuald.isty-info.uvsq.fr`

— Pour diminuer le nombre de requêtes, les serveurs gardent un cache des résultats antérieurs

Types d'enregistrements

- Le DNS permet d'associer des enregistrements à des noms de machines ou de domaines
- Principaux types d'enregistrements :
  - `A` : adresse IP (Authority)
  - `AAAA` : adresse IPv6
  - `CNAME` : nom canonique pour un alias (Canonical Name)
  - `HINFO` : informations sur le système (Host Info), peu utilisé
  - `MX` : relais de courrier électronique (Mail eXchange)
  - `NS` : serveur de noms (Name Server), pour déléguer une zone
  - `PTR` : pointeur (vers un autre nom)
  - `WKS` : services fournis (Well Known Services), obsolète
  - `RP` : e-mail du responsable de la zone (Responsible Person)
  - `SOA` : description de la zone (Start Of Authority), voir plus loin...

Résolution d'adresse en nom



Utilisation du DNS

- Le DNS est bâti selon un schéma client-serveur
- Le client fait partie de la bibliothèque C (`gethostbyname`, `gethostbyaddr`)
- Il s'adresse aux serveurs spécifiés par `/etc/resolv.conf`
- Exemple :

```
domain csi.uvsq.fr
search csi.uvsq.fr ens.uvsq.fr uvsq.fr
nameserver 193.51.24.1 # soleil.uvsq.fr
nameserver 193.51.26.1 # atlas.ens.uvsq.fr
nameserver 193.51.25.1 # guillotin.prism.uvsq.fr
```

Interrogation du DNS

— Plusieurs commandes : `nslookup` (obsolète), `host`, `dig`

— Exemple `nslookup` :

```
$ nslookup
Default Server:  soleil.uvsq.fr
Address:  193.51.24.1

> romuald.isty-info.uvsq.fr

Name:    romuald.isty-info.uvsq.fr
Address:  193.51.33.1

> set q=ptr
> 1.33.51.193.in-addr.arpa

1.33.51.193.in-addr.arpa        name = romuald.isty-info.uvsq.fr
```

Interrogation du DNS (2)

— Exemple `dig` :

```
$ dig inria.fr

; <<>> DiG 9.3.0 <<>> inria.fr
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64986
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;inria.fr.                      IN      A

;; AUTHORITY SECTION:
inria.fr.               7200    IN      SOA     dns.inria.fr. \
            hostmaster.sophia.inria.fr. \
            2006111400 21600 3600 3600000 7200
```

Interrogation du DNS (3)

— Exemple `dig` pour rechercher un enregistrement particulier :

```
$ dig inria.fr -t MX

; <<>> DiG 9.3.0 <<>> inria.fr
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64986
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;inria.fr.                      IN      A

;; ANSWER SECTION:
inria.fr.               55492   IN      MX      50 concorde.inria.fr.
inria.fr.               55492   IN      MX      50 discorde.inria.fr.
```

Mise en place d'un serveur de noms

- La mise en place d'un serveur de noms consiste à configurer et à activer le serveur `named`
- Pour être serveur primaire d'une zone, il faut obtenir la délégation dans la zone de niveau supérieur
- Un serveur peut être :
  - uniquement cache ;
  - primaire et/ou secondaire.
- Configuration de `named` :
  - `/etc/named.boot` : `named` version 4
  - `/etc/named.conf` : `named` version 8

    (version actuelle 9.4)

Serveur de noms cache

— Un serveur cache transmet les requêtes à un autre serveur

— Le résultat des requêtes est sauvegardé dans le cache des deux serveurs

```
options {
        directory "/etc/namedb";
//      forward only;
        forwarders {
                193.51.24.1;
        };
};
zone "." {
        type hint;
        file "named.root";
};
zone "0.0.127.IN-ADDR.ARPA" {
        type master;
        file "localhost.rev"; };
```

Liste des serveurs racine

— Un serveur de noms doit connaitre la liste des serveurs de la racine pour les contacter

— Sa configuration inclut cette liste (`named.root`, `root.cache`) :

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
...
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4

.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     128.9.0.107

.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
...
```

Serveur primaire et/ou secondaire

- Un serveur peut être primaire pour certaines zones et secondaires pour d'autres
- Chaque zone doit être définie dans `/etc/named.conf`
- Pour une zone primaire :
  - nom de zone
  - type (`master`)
  - nom de fichier
- Pour une zone secondaire :
  - nom de zone
  - type (`slave`)
  - nom de fichier
  - adresse IP du serveur primaire

Exemple (1)

```
options {
        directory "/local/named";
};

zone "." {
        type hint;
        file "root.cache";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "prim/local/localhost";
};
```

Exemple (2)

```
zone "uvsq.fr" {
        type master;
        file "prim/uvsq/uvsq";
};

zone "isty-info.uvsq.fr" {
        type master;
        file "prim/uvsq/isty-info";
};

zone "33.51.193.in-addr.arpa" {
        type master;
        file "prim/uvsq/33.isty-info";
};
```

Exemple (3)

```
zone "prism.uvsq.fr" {
        type slave;
        file "second/uvsq/prism";
        masters {
                193.51.25.1;
        };
};

zone "25.51.193.in-addr.arpa" {
        type slave;
        file "second/uvsq/25.prism";
        masters {
                193.51.25.1;
        };
};
```

Sécurité

— Sur un master :

`allow_transfers{193.51.25.2}`

On n'autorise en principe que les serveurs secondaires à faire des transferts de zone

— Sur un slave :

`allow_query{any}`

`allow_recursion{193.51.25.0/24}`

On autorise toutes les machines à faire une requête, mais seulement les machines du réseau "local" à faire une requête récursive, i.e. pour laquelle le serveur peut répercuter la demande sur les serveurs de la zone supérieure (sinon, c'est un "open DNS server").

Définition d'une zone

- *Start of Authority* (SOA) : paramètres de la zone
  - nom du serveur primaire
  - adresse électronique du contact
  - numéro de version (aussi appelé numéro de série)
  - délai de rafraîchissement
  - délai avant un nouvel essai
  - délai d'expiration
  - *Time To Live* minimum
- Enregistrements

Exemple (“`file "prim/uvsq/isty-info"`”)

```
@       IN      SOA     soleil.uvsq.fr.     hostmaster.soleil.uvsq.fr. (
                                200003162   ; Version
                                21600       ; Refresh (6h)
                                3600        ; Retry   (1h)
                                2592000     ; Expire (30j)
                                259200 )    ; Minimum TTL (3j)

; Serveurs primaire et secondaires
                IN      NS      soleil.uvsq.fr.
                IN      NS      atlas.ens.uvsq.fr
                IN      NS      guillotin.prism.uvsq.fr

; Relais de courrier pour la zone
                IN      MX      100 soleil.uvsq.fr.
                IN      MX      200 shiva.jussieu.fr.
```

Exemple (“`file "prim/uvsq/isty-info"`”)

```
; Designation des noeuds de la zone isty-info.uvsq.fr.
mailhost        IN      CNAME   romuald

; Les serveurs
romuald         IN      A       193.51.33.1
                IN      MX      100 soleil.uvsq.fr.
                IN      MX      200 shiva.jussieu.fr.
www             IN      CNAME   romuald

athanase        IN      A       193.51.33.2
                IN      MX      100 soleil.uvsq.fr.
                IN      MX      200 shiva.jussieu.fr.
...
```

Exemple (“`file "prim/uvsq/33.isty-info"`”)

```
@       IN      SOA     soleil.uvsq.fr.     hostmaster.soleil.uvsq.fr. (
                                200003162   ; Version
                                21600       ; Refresh (6h)
                                3600        ; Retry   (1h)
                                2592000     ; Expire (30j)
                                259200 )    ; Minimum TTL (3j)

; Serveurs primaire et secondaire
                IN      NS      soleil.uvsq.fr.
                IN      NS      atlas.ens.uvsq.fr
                IN      NS      guillotin.prism.uvsq.fr

; Designation des noeuds de la zone 33.51.193.in-addr.arpa.
0       IN      PTR     fr-uvsq-10.uvsq.fr.
; les serveurs
1       IN      PTR     romuald.isty-info.uvsq.fr.
2       IN      PTR     athanase.isty-info.uvsq.fr.
```

Attribution dynamique d'adresse IP

- Certains éléments d'un réseau n'ont pas d'adresse IP fixée
- Exemples :
  - terminaux X-Window
  - stations sans disques
  - appareils nomades (portables, mobiles, . . .)
- Plusieurs protocoles d'attribution d'adresse IP :
  - RARP
  - BOOTP
  - DHCP

BOOTP

- Le serveur `bootpd` peut être lancé :
  - au démarrage :

    ```
    bootptd -s
    ```
  - par `inetd` (ou `xinetd`) :

    ```
    bootps dgram  udp   wait  root  /usr/libexec/bootpd   bootpd
    ```
- Le fichier `/etc/bootptab` définit les paramètres de *boot* :
  - adresse MAC
  - adresse IP
  - masque de réseau
  - routeur
  - serveur(s) de noms
  - fichier à charger
  - etc

Exemple (1)

```
# Les valeurs par defaut
.default:                :sm=255.255.255.0:\
                         :gw=193.51.26.254:\
                         :ht=ethernet:\
                         :dn=ens.uvsq.fr:\
                         :ds=193.51.26.1, 193.51.24.1:\
                         :ts=ntp1.uvsq.fr, ntp2.uvsq.fr:\
                         :hn:
# Terminaux X Tektronics (salles de DEUG)
.tek:                    :tc=.default:\
                         :hd=/usr/local/boot:\
                         :bf=tekxp.new/boot/os.500:
# Terminaux X NCD de la salle 101
.ncd-xpl:                :tc=.default:\
                         :bf=Xncdxpl:\
                         :hd=/usr/local/boot/ncd/bin:\
                         :ht=ether
```

Exemple (2)

```
# Salle 203 Tx Tektronics a partir de 138
dijon:                  :tc=.tek:ha=08001108efff:
auxerre:                :tc=.tek:ha=08001108efea:
sens:                   :tc=.tek:ha=08001108ed4d:
avallon:                :tc=.tek:ha=08001108eff3:
beaune:                 :tc=.tek:ha=08001108ed4c:
nevers:                 :tc=.tek:ha=08001108efec:
chateau-chinon:         :tc=.tek:ha=08001108ed54:
macon:                  :tc=.tek:ha=08001108ed1f:
autun:                  :tc=.tek:ha=08001108f008:
louhans:                :tc=.tek:ha=08001108ed5b:
charolles:              :tc=.tek:ha=08001108ed22:
clamecy:                :tc=.tek:ha=08001108ed4f:
cosne:                  :tc=.tek:ha=08001108ed4a:
chalon-sur-saone:       :tc=.tek:ha=08001108ed58:
montbard:               :tc=.tek:ha=08001108ed2a:
tournus:                :tc=.tek:ha=08001108efe8:
```

DHCP

- *Dynamic Host Configuration Protocol*
- Compatible de manière ascendante avec BOOTP
- Affectation dynamique d'adresses :
  - des plages d'adresses peuvent être définies
  - les adresses sont affectées dans l'ordre des demandes
  - ou avec des adresses fixes en utilisant une table de correspondance avec les adresses MAC
- Gestion d'un délai de validité des adresses

- Lancement d'un démon `dhcpd`(`service dhcpd start`)
- Configuration dans `/etc/dhcpd.conf`
- Base de donnée des concessions dans `/var/lib/dhcp/dhcpd.leases`

DHCP, configuration

```
server-identifier dhcp.inria.fr;
allow unknown-clients;
option domain-name "inria.fr";
option smtp-server nez-perce.inria.fr, concorde.inria.fr;

shared-network INRIA-DHCP-NET {

  # reseau interne
  subnet 128.93.0.0 netmask 255.255.192.0 {
    option routers 128.93.1.100;
    option broadcast-address 128.93.63.255;
    option domain-name-servers 128.93.1.23, 128.93.1.9, 192.93.2.78;
    option netbios-name-servers 128.93.50.1, 128.93.50.2;
    pool {
        failover peer "dhcp-failover";
        range 128.93.62.1 128.93.62.254;
        deny unknown-clients;
        deny dynamic bootp clients;
    }
  }
```

DHCP, configuration (suite)

```
  # reseau invités, en fait les machine obtiennent une adresse privee
  subnet 10.10.10.0 netmask 255.255.255.0 {
    allow unknown-clients;
    option routers 10.10.10.254;
    option broadcast-address 10.10.10.255;
    option domain-name-servers 10.10.10.254;
    pool {
        deny known clients;
        range 10.10.10.1 10.10.10.253;
        default-lease-time 43200;
        max-lease-time 43200;
    }
  }
}

host client1 {
    fixed-address     client1.inria.fr;
    hardware ethernet 00:80:C8:87:09:E9;
    option dhcp-client-identifier 01:00:80:C8:87:09:E9;
}
host client2 {
.........
```

DHCP, négociation

— Le client DHCP émet, par broadcast sur le réseau local, une trame de découverte DHCP, “DHCP Discovery”

— Un serveur DHCP reçoit ce broadcast. S’il est capable de satisfaire la demande, il répond en émettant une trame “DHCP Offer”, incluant les paramètres requis par le client DHCP

— Le client reçoit l’offre, et peut décider d’accepter ou d’attendre éventuellement d’autres propositions de serveurs DHCP.
Il signifie son accord au serveur DHCP dont il retient l’offre, par une trame de broadcast “DHCP Request”

— Si le serveur sélectionné par le client DHCP est capable de satisfaire les options souhaitées, il acquitte la demande en émettant une trame “DHCP Ack” ; il lui confirme l’adresse IP et les paramètres associés, et enregistre le “Binding” dans sa base d’information.

X Display Manager (XDM)

- Gère les connexions depuis les serveurs X :
  - locaux
  - distants (terminaux X)
- Bannière de connexion :
  - authentification
  - exécution de fichiers de commandes
- `/usr/lib/X11/xdm/xdm-config`
  - définition des paramètres associés à chaque serveur X
- `/usr/lib/X11/xdm/Xservers` :
  - liste des serveurs X utilisant XDM
  - à modifier uniquement si le terminal X n'utilise pas XDMCP (XDM Control Protocol)

Configuration de XDM (1)

- Fichier `/usr/lib/X11/xdm/xdm-config`
- Paramètres de configuration :
  - globaux
  - spécifiques à chaque serveur X
- Paramètres spécifiques à chaque serveur X
  - `DisplayManager.`*serveur*`.`*paramètre* : `valeur`
  - `resources` : fichier de ressources à charger par `xrdb`
  - `setup` : programme exécuté avant l'authentification (sous `root`)
  - `startup` : programme exécuté après l'authentification (sous `root`)
  - `session` : programme exécuté après l'authentification
  - `reset` : programme exécuté à la fin de la session (sous `root`)

Configuration de XDM (2)

— Exemple :

```
DisplayManager.accessFile:  /usr/lib/X11/xdm/Xaccess
DisplayManager.servers:     /usr/lib/X11/xdm/Xservers
DisplayManager.errorLogFile:/usr/lib/X11/xdm/xdm-errors
DisplayManager.pidFile:     /usr/lib/X11/xdm/xdm-pid
DisplayManager*resources:   /usr/lib/X11/xdm/Xresources
DisplayManager.TX_0.setup:  /usr/lib/X11/xdm/Xsetup_TX
DisplayManager*startup:     /usr/lib/X11/xdm/Xstartup
DisplayManager*session:     /usr/lib/X11/xdm/Xsession
```

Ressources de XDM

- Définies dans le fichier `Xresources`
- Influent sur le comportement de `xdm`
- Exemple :

```
xlogin*login.translations: #override \
   Ctrl<Key>R: abort-display()\n \
   <Key>F1: set-session-argument(failsafe)
finish-field()\n \
   <Key>Return: set-session-argument() finish-field()\n
xlogin*borderWidth: 3
xlogin*greeting: 'Bienvenue sur le serveur'
xlogin*namePrompt: 'Serveur Login : '
```

Programme d'initialisation

- Défini par le champ `setup`
- Exécuté :
  - après la ré-initialisation du serveur X
  - avant l'affichage de la fenêtre d'authentification
- Exécuté avec les droits de `root` (attention aux trous de sécurité !)
- Ne peut pas recevoir d'entrée du clavier
- Exemple :

```
#!/bin/sh
xconsole -geometry 480x130-0-0 -notify -verbose &
```

Initialisation après connexion

- Programme défini par le champ `startup`
- Exécuté après l'authentification, avec les droits de `root`
- XDM s'interrompt si son code de retour est non nul
- Exemple :

```
#!/bin/sh
if [ -f /etc/nologin ]; then
   xmessage -file /etc/nologin -timeout 30 -center
   exit 1
fi
sessreg -a -l $DISPLAY -x /usr/lib/X11/xdm/Xservers \
        $LOGNAME
exit 0
```

Lancement de la session

- Programme défini par le champ `session`
- Exécuté après l'authentification, avec les droits de l'utilisateur
- Sa fin provoque la terminaison de XDM
- Ce programme, ou script, doit lancer les clients X initiaux
- Généralement :
  - traitement du cas 'failsafe'
  - exécution d'un script spécifique à l'utilisateur

Exemple

```
#!/bin/sh
if [ $# -eq 1 ]; then
   if [ $1 = failsafe ]; then
        exec xterm
   fi
fi
startup=$HOME/.xsession
resources=$HOME/.Xresources
if [ -x '$startup' ]; then
   exec '$startup'
else
   [ -f '$resources' ] && xrdb -load '$resources'
   xterm &
   xman &
   exec twm
fi
```

Programme de ré-initialisation

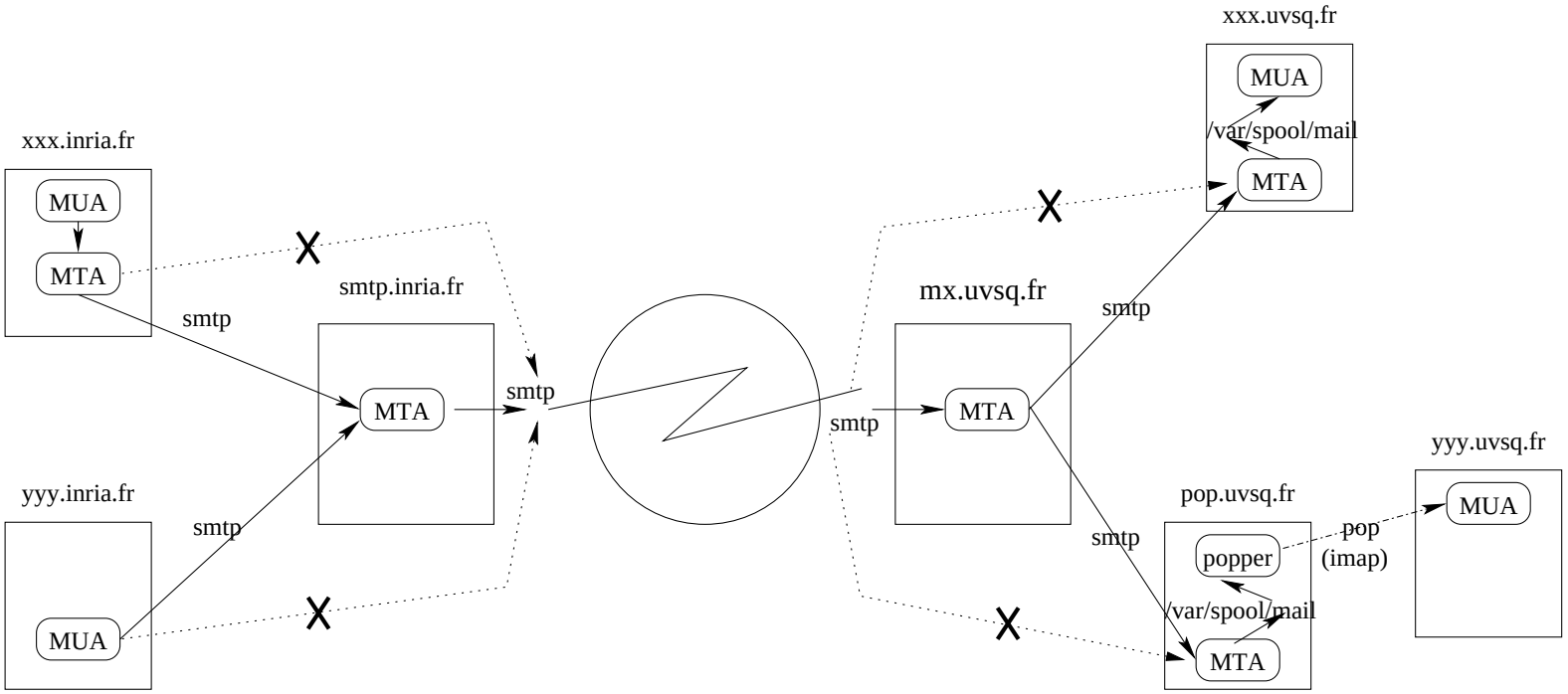
- Défini par le champ `reset`
- Exécuté à la fin de la session, avec les droits de `root`
- Exemple :

```
#!/bin/sh
sessreg -d -l $DISPLAY -x /usr/lib/X11/xdm/Xservers \
        $LOGNAME
```

Courrier électronique

- Trois types de programmes :
  - MTA (*Mail Transport Agent*) : acheminement du mail entre machines (exemples : `sendmail`, `postfix`, `exim`)
  - LDA (*Local Delivery Agent*) : dépose du mail dans la boîte aux lettres de l'utilisateur (exemples : `sendmail`, `mail.local`, `procmail`)
  - MUA (*Mail User Agent*) : lecture du courrier local et envoi via un MTA (exemples : `elm`, `mail`, `mutt` ; Mozilla Thunderbird, Outlook, ...)
- Accès au courrier :
  - accès direct à un fichier contenant le courrier (par ex. `/var/mail`)
  - accès par réseau : protocoles POP, IMAP, etc.
- Transmission de courrier entre machines : protocole SMTP (et SSMTP, ...)
- Entrée et sortie d'un site :
  - MX : désigné par le DNS comme machine à utiliser comme MTA en entrée du domaine
  - passerelle : machine utilisée par toutes les autres comme MTA pour sortir

Courrier électronique



sendmail

- MTA écrit à Berkeley, puis inclus dans la plupart des systèmes Unix
- Rôle de `sendmail` :
  - routage du courrier entre MUA et programmes de livraison (locale ou non)
  - réception et livraison de courriers reçus depuis le réseau
  - gestion d'alias de courrier, permettant de créer des listes de diffusion
- Configuration très puissante (`/etc/sendmail.cf`)...
- ...mais illisible
- Utilisation de kits de haut niveau :
  - macros M4
  - kit Jussieu

POP et IMAP

- POP (Post Office Protocol)
  - Permet d'accéder à sa boite aux lettres sur un serveur distant
  - Port spécifique (110, et 995 avec le mode sécurisé POPS)
  - Avec une authentification (nom + mot de passe)
  - Les messages peuvent rester sur le serveur (option "leave on server"), mais pas trop longtemps
  - Outils : serveurs `pop3d` (vieux), `qpopper`, `dovecot`, client `fetchmail`
- IMAP (Internet Message Access Protocol)
  - Permet d'accéder à sa boite aux lettres sur un serveur distant
  - Port spécifique (143, et 993 avec le mode sécurisé IMAPS)
  - Avec une authentification (nom + mot de passe)
  - Les messages restent sur le serveur ainsi que les folders (sous-répertoires dans `/var/spool/mail`)
  - Outils : serveurs `wu-imap` (Washington University, limité), `Cyrus Imap`, `dovecot`

Serveur Web : Apache

— Le serveur Web sous Unix

— Deux versions (Apache 1 et Apache 2), très différentes

— Lancement :

  - `/etc/init.d/httpd {stop|start}`

— Configuration :

  - `/etc/httpd/conf/http.conf` (ou `http2.conf`)
  - Plusieurs sites : plusieurs fichiers de configuration, lancement de `httpd -f <site>.conf`

— Localisation des données :

  - En standard, dans `/var/www/html`
  - `/var/www/html/hello-world.html` vu à `http://localhost/hello-world.html`

Serveur Web : Apache, configuration

`/etc/httpd/conf/http2.conf` (extraits) :

`ServerType standalone`

Le serveur s'executera seul, sans recourir au super-serveur `xinetd`.

`ServerRoot /etc/httpd`

Répertoire de configuration (conf globale et confs spécifiques).

`PidFile /var/run/httpd.pid`

Fichier stockant le PID de httpd (premier process)

`DocumentRoot /var/www/html`

Racine des documents (`index.html` etc.) pour ce site.

`Port 80`

Port sur lequel apache écoute, 80 par défaut (peut être 8080,...)

`User apache`, `Group apache`

Utilisateur et groupe propriétaire des process httpd.

Serveur Web : Apache, configuration

`UserDir public_html`

Répertoire par défaut pour le site personnel des utilisateurs (`~joe/public_html/index.html` vu à l'URL `http://www.mydomain.fr/~joe/`).

`DirectoryIndex index.html index.php index.htm`

Les fichiers pris comme point d'entrée dans un répertoire, dans l'ordre

`AccessFileName .htaccess`

Le fichier précisant les conditions d'accès (.htaccess est le nom par défaut).

Et d'autres :

temps de timeout, durée des sessions, nombre max de connexions, nombre min et max de processus, ...

Et l'inclusion des configurations spécifiques :

```
Include /etc/httpd/conf.d/*.conf
```

Serveur Web : Apache + PHP/MySQL

— Configuration PHP :
Dans `/etc/httpd/conf.d/`, fichier `70_mod_php.conf` :

```
LoadModule php4_module    extramodules/mod_php4.so
....
AddType application/x-httpd-php .php
```

— “Lancement” PHP :

`/var/www/html/hello-world.php`

vu à l’URL `http://localhost/hello-world.php`

— Configuration MySQL :
Dans `/etc/httpd/conf.d/`, fichier `12_mod_auth_mysql.conf` :

```
....
LoadModule mysql_auth_module    extramodules/mod_auth_mysql.so
....
```

Serveur Web : Apache + PHP/MySQL

— Lancement : `/etc/init.d/mysql {stop|start}`
— Initialisation :

```
# mysql_install_db
2 utilisateurs: root@localhost, @localhost
$ mysql
mysql> show databases;
mysql> use mysql;
mysql> show tables;
mysql> describe user;
mysql> select  host, user, password from user;
# mysqladmin -u root password xxxxx
# mysql
  access denied for user root@localhost
# mysql -u root -p mysql
  Enter password : xxxxx
  Welcome to the MySQL monitor
  ......
  mysql>  select  host, user, password from user;
```

Serveur Web : Apache, sécurisation

— httpd.conf :

```
<Directory />
order deny, allow
deny from all
Options None
AllowOverride None
</Directory>

<Directory /var/www/html>
Options Indexes Includes FollowSymLinks
AllowOverride All # on autorise l'"écrasement" par les .htaccess
order allow,deny
allow from all
</Directory>
```

— .htaccess :

```
Options -FollowSymLinks -Indexes
deny from 123.45.67.0/255.255.255.0
```

Serveur Web : Apache, sécurisation

— Liste des options :

`All | None` : toutes / aucune option(s) permise(s)

`ExecCGI` : exécution de scripts autorisée (CGI : Common Getaway Interface – perl, python, php . . .)

`FollowSymLinks` : le serveur suivra les liens symboliques

`Includes` : permet l'utilisation de CSS (Cascading Style Sheets)

`IncludesNOEXEC` : idem sauf les directives #exec et #include

`Indexes` : autorise l'affichage du contenu d'un répertoire

`AllowOverride` : prise en compte du fichier .htaccess

Serveur Web : Apache, authentification

— Création d'un fichier de mots de passe cryptés

```
# cd /var/www/html
# htpasswd -c .priv_passwd admin
  (saisie d'un mot de passe)
# htpasswd .priv_passwd webmaster
# htpasswd .priv_passwd joe
# ...
```

— Utilisation pour authentification : dans le `.htaccess`

```
AuthUserFile /var/www/html/.priv_passwd
AuthGroupFile /dev/null
AuthName "Acces prive"
AuthType Basic

<limit GET>
require valid-user
</limit>
```

En pratique, `.priv_passwd` dans un répertoire séparé regroupant les fichiers de mots de passe

Serveur Web : Apache, authentification

— Authentification indirecte (LDAP, Kerberos, . . .) :

```
AuthLDAPEnabled on
AuthType Basic
AuthName "LDAP Restricted Directory"
AuthLDAPUrl ldap://ldapserver.mydomain.fr:389/ou=people,\
          dc=mydomain,dc=fr?login?sub?(uid=*)
<limit GET>
require valid-user
</limit>
```

Un petit mot sur Nginx et Lighttpd

- Nginx
  - Développement commencé en 2002
  - Requêtes découpés en micro taches
  - Excellentes performances et faible empreinte mémoire
  - A utiliser pour les services nécessitant un fort trafic
- Lighttpd
  - Léger et flexible
  - Rapide
  - Faible empreinte mémoire

SÉCURITÉ ET CRYPTOGRAPHIE

I think people can generally trust me, but they can trust me exactly because they know they don't have to.

Linus Torvalds

Objectifs théoriques

- authentifier les utilisateurs, gérer leurs autorisations
- assurer la confidentialité et l'intégrité des données et des communications
- assurer la disponibilité des services

On distingue :

- Sûreté : protection contre les actions non intentionnelles
- Sécurité : protection contre les actions intentionnelles malveillantes

Sécurité réseau

- Une machine accessible par réseau est plus exposée aux atteintes à la sécurité
- Plusieurs types d'attaques :
  - atteinte à la confidentialité
  - disponibilité des données
  - intégrité des données
- Quelques pistes :
  - authentification des utilisateurs
  - sécurité des commandes "r"
  - surveillance automatisée
  - contrôles d'accès

Authentification des utilisateurs

- Exiger de bons mots de passe :
  - `passwd+`, `npasswd`
  - `Crack`
  - Utilisation de générateurs de mots de passe
- Utiliser les fichiers shadow :
  - mots de passe chiffrés non lisibles
  - possibilité d'expiration de comptes
- Installer un système de mot de passe à usage unique (OTP, One Time Password).

  Exemple : OPIE (*One-time Password In Everything*), `skey`
- Utiliser un service d'authentification dédié (Kerberos, LDAP, LDAP+Kerberos)

Les PAMs

— **Objectif** : les PAM (Pluggable Authentication Module), moyen générique d'authentifier des utilisateurs, indépendamment de chaque application. Uniformisation ou changement global (Kerberos, LDAP,...).

— **Configuration** : fichier `/etc/pam.conf` (Solaris, ...), fichiers dans `/etc/pam.d` (Linux,...)

Une ligne = 3 champs :

- *Type de module* : `auth` pour l'authentification, `account` pour l'accès au compte, `password` pour la gestion des mots de passe, et `session` pour l'ouverture d'une session ;
- *Drapeau de contrôle* : `required` si la réussite est requise, `requisite` pour arrêt immédiat en cas d'échec, `optional` pour action sans condition, et `sufficient` pour acceptation immédiate.
- *Nom du module* : librairie partagée, dans `/lib/security`.

Les PAMs

— **Quelques PAM à connaître**

— *pam_unix* : authentification unix standard

— *pam_pwdb* : module de base équivalent à pam_unix.

— *pam_nologin* : permet de désactiver les comptes si le fichier `/etc/nologin` existe

— *pam_securetty* : pour `root`, vérifie que le terminal utilisé est listé dans `/etc/securetty`

— *pam_console* : permet de spécifier les droits d'accès à la console, en utilisant le fichier `/etc/security/console.perms`

— *pam_cracklib* : permet de vérifier, en particulier au moment du changement, qu'un mot de passe n'est pas dans un dictionnaire et éventuellement qu'il vérifie certains critères (longueur,...).

Les PAMs

— **Exemple** : la PAM login en mode Kerberos

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_nologin.so
auth       sufficient   /lib/security/pam_unix.so
auth       sufficient   /lib/security/pam_krb5afs.so try_first_pass \
                        tokens debug
auth       required     /lib/security/pam_pwdb.so shadow nullok
account    required     /lib/security/pam_pwdb.so
password   required     /lib/security/pam_cracklib.so
password   required     /lib/security/pam_pwdb.so shadow nullok \
                        use_authtok
session    optional     /lib/security/pam_console.so
session    optional     /lib/security/pam_krb5afs.so
session    required     /lib/security/pam_pwdb.so
```

Sécurité des commandes ”r”

- Suppression des entrées “dangereuses” dans `/etc/inetd.conf`
  - services non utilisés
  - services peu sécurisés
- Vérification des fichiers d’équivalence
  - `/etc/hosts.equiv`
  - `.rhosts`
- Utilisation de `ssh`
  - remplacement des commandes standard (`rcp`, `rlogin`, `rsh`)
  - versions sécurisées : authentification et chiffrement
  - transmission chiffrée du protocole X

Surveillance automatisée

- Logiciels de surveillance :
  - `tcpdump` et sa surcouche graphique Ethereal, remplacée par WireShark
  - Tripwire
  - COPS
  - SATAN
  - NESSUS
- mais aussi, petits outils standards :
  - comptabilité
  - `find`

Contrôles d'accès

- Limitation de l'accès aux services
  - TCP wrappers
  - filtres sur les routeurs
- Mise en place d'une machine coupe-feu :
  - serveur de noms pour l'extérieur
  - routage des courriers électroniques
  - services proxy

TCP Wrappers

— Activé par `inetd`, au travers de `/etc/inetd.conf`

```
telnet  stream tcp nowait root    /usr/sbin/tcpd in.telnetd
```

— Ce qui est interdit : `/etc/hosts.deny`

```
in.telnetd:     *.pirates.org
```

ou mieux :

```
ALL:            ALL
```

voire (syntaxe Linux) :

```
ALL:            ALL: spawn (echo "tentative d'intrusion sur %d par %u@%
                     | mail -s tcpd root)&
```

— Ce qui est permis : `/etc/hosts.allow`

```
in.telnetd:     *.isty-info.uvsq.fr
```

— Compatibilité ascendante avec `xinetd` qui prend en compte ces deux fichiers

Autres pistes

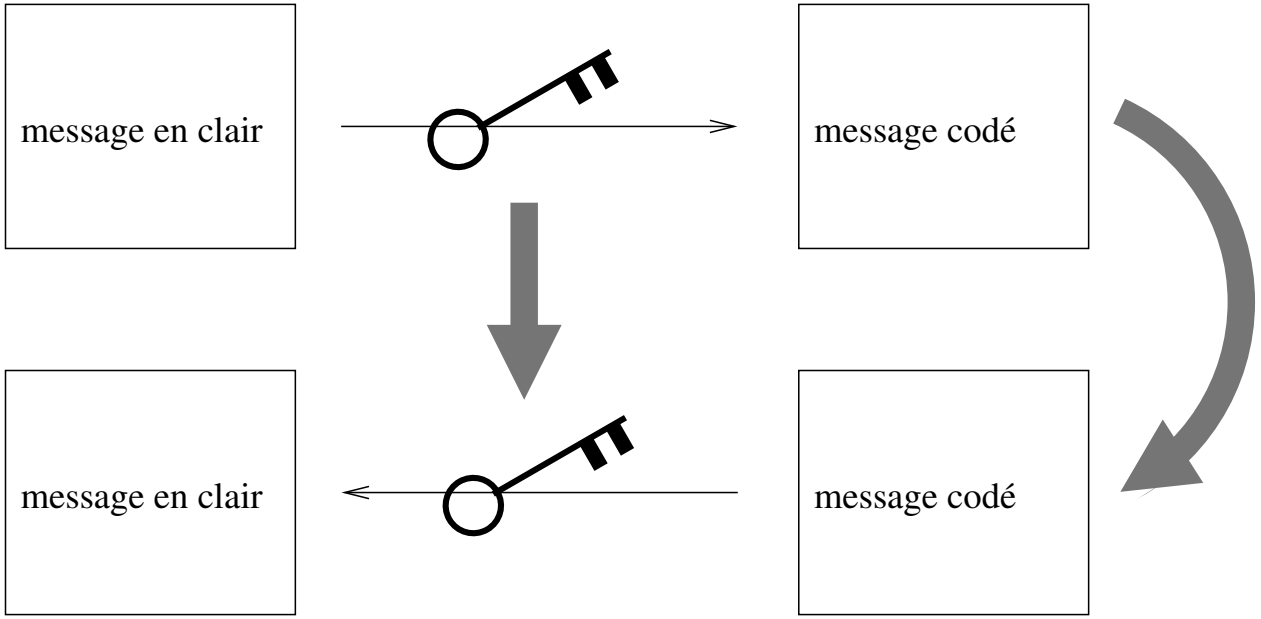
- Bannir des IP
  - fail2ban
  - SSH, HTTP, FTP...
- Firewall :
  - iptables
- Man in the middle :
  - Echange de clés via moyen de confiance
  - Authentification avec mot de passe
- SSL / TLS

Cryptographie et certificats

- Cryptographie
- Signature électronique
- Certificats électroniques
  - Infrastructures de gestion de clé
  - Législation sur la signature électronique

Le cryptage symétrique

— utilisation d'une clé secrète de chiffrement partagée par l'expéditeur et le destinataire pour chiffrer un message



Le cryptage symétrique

— Les algorithmes

Algorithme	longueur de la clef
DES	56 bits
Triple DES	128, 156 bits
IDEA	128 bits
RC4	variable
Blowfish	1-448 bits
AES	128, 192, 256 bits

— Inconvénients

- Premiers codes avec des clefs courtes (implémentations cablées)
- Nécessité d'un canal sûr pour transmettre la clef
- Nécessite d'une clef différente par destinataire

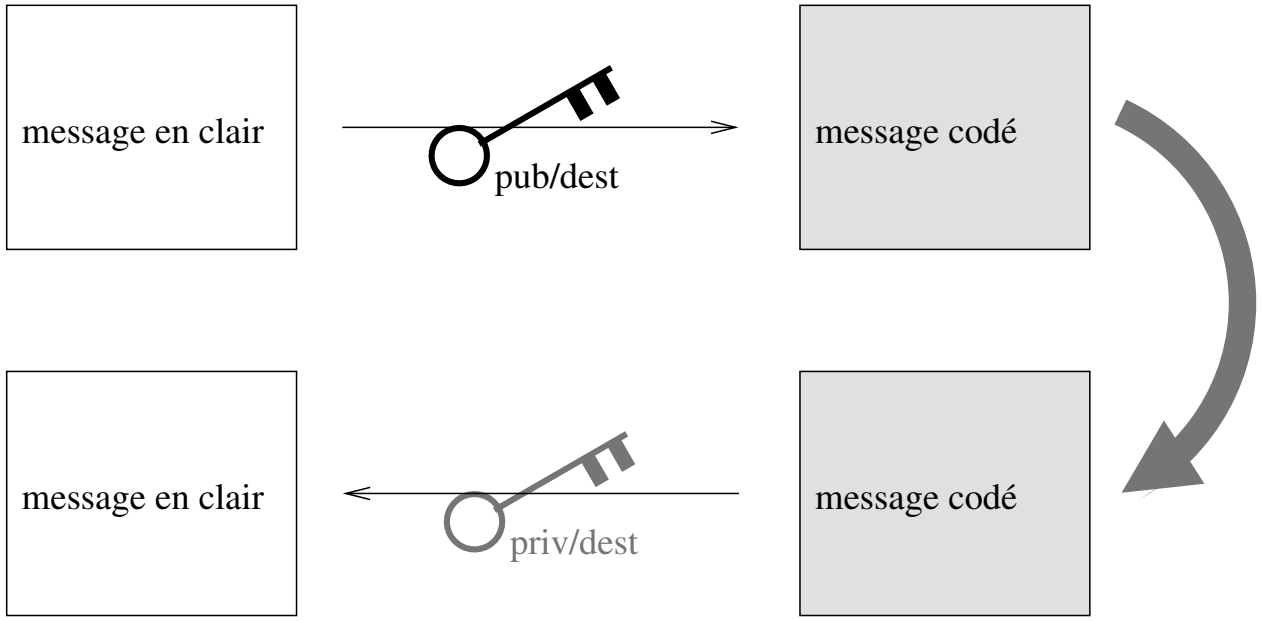
La cryptographie à clef publique : cryptage asymétrique

— Histoire
  - Diffie et Hellman, 1976
  - Le plus utilisé : RSA, 1977

— Principe
  - Création par une personne d'un couple clef publique / clef privée, nombres liés par une opération mathématique
  - Tout message encrypté avec la clef publique, ne peut être décrypté qu'avec la clef privée
  - Et réciproquement : tout message encrypté avec la clef privée, ne peut être décrypté qu'avec la clef publique

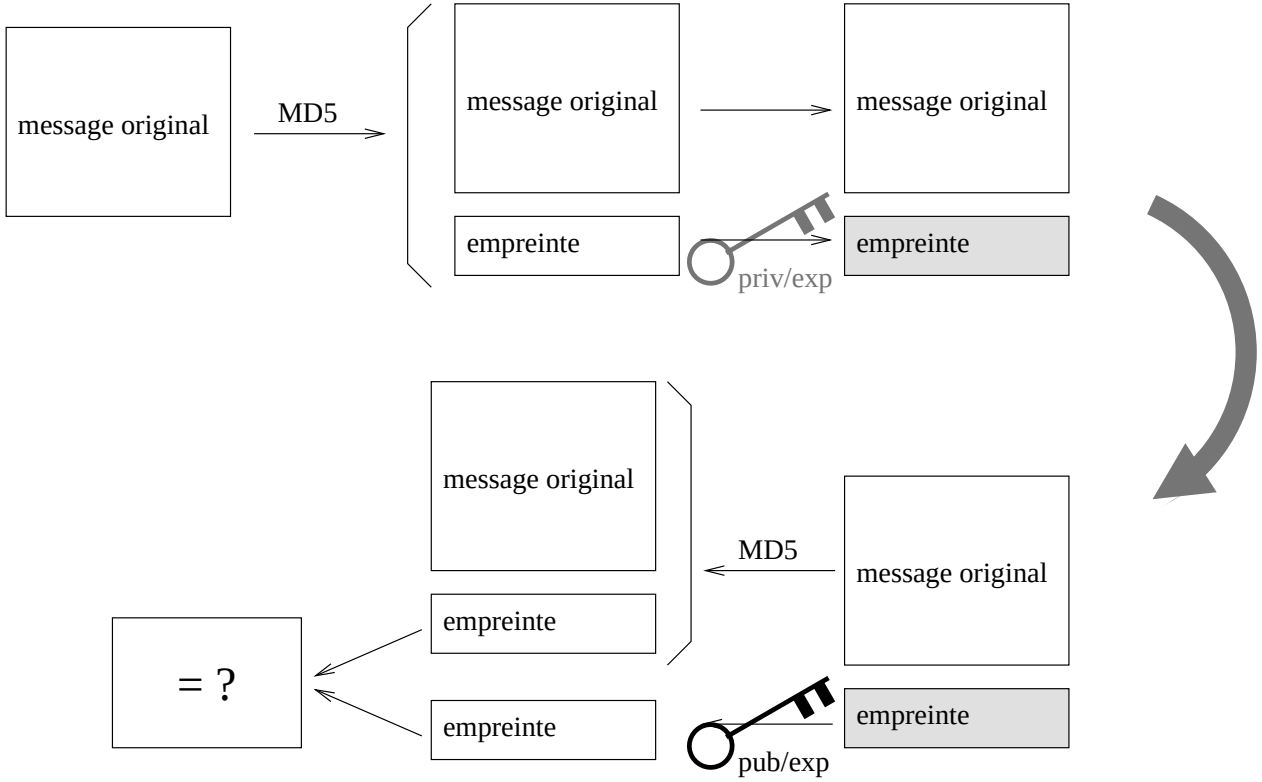
La cryptographie à clef publique : cryptage

— Un expéditeur encrypte un message avec la clef publique du destinataire, qui est le seul à pouvoir le décrypter avec sa clef privée



La cryptographie à clef publique : signature

— Un expéditeur calcule une empreinte de son message (MD5, etc.) et l'encrypte avec sa clef privée ; le destinataire doit utiliser la clef publique de l'expéditeur pour décrypter l'empreinte, et compare l'empreinte du message initial avec le résultat.



La cryptographie à clef publique

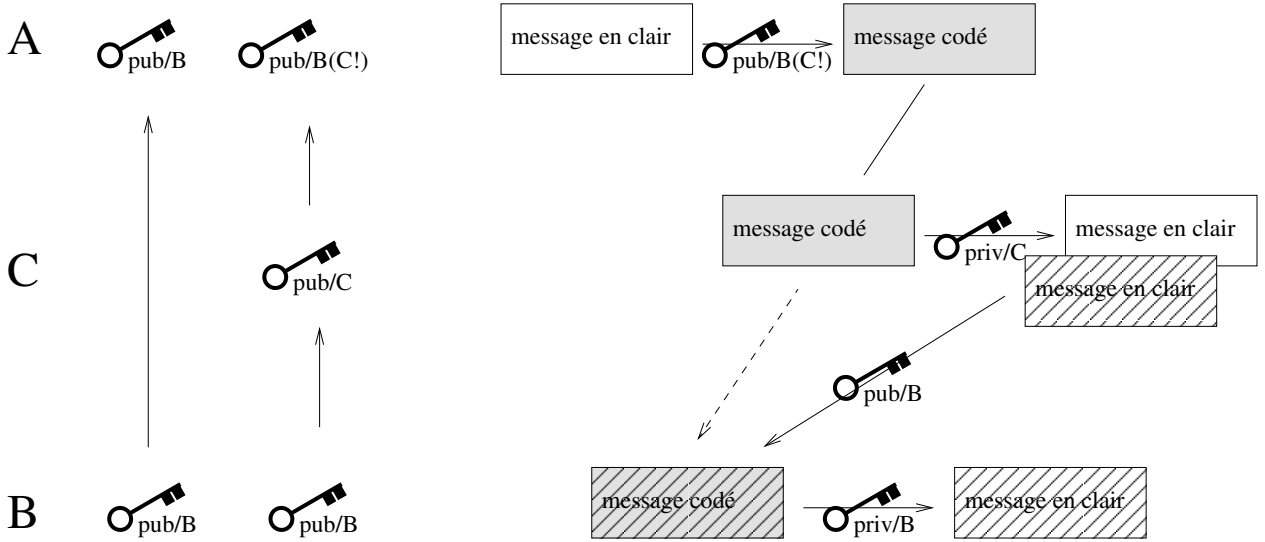
— Cas des messages vers plusieurs destinataires
  - Cryptage : pour éviter de crypter *n* fois le message (temps cpu), on l'encrypte avec une clef de session, et c'est cette clef (plus courte) que l'on encrypte avec la clef publique de chaque destinataire.
  - Signature : rien de changé (empreinte cryptée une seule fois indépendamment du nombre de destinataires).

— Cryptage et signature
  - un expéditeur encrypte un message avec la clef publique du destinataire
  - l'expéditeur calcule ensuite une empreinte du résultat et la crypte avec sa clef privée
  - le destinataire s'assure de l'origine du message en comparant son empreinte avec celle résultant du décryptage de celle envoyée, à l'aide de la clef publique de l'expéditeur
  - il peut lire le message en clair en le décryptant avec sa propre clef privée.

La cryptographie à clef publique

— Le point noir : comment être sûr de l'origine d'une clef publique ?

— Exemple d'exploitation, the "man in the middle attack" : C transmet sa clef publique à A, ce dernier pensant recevoir celle de B . . .



Certificats électroniques

— Utilité : certifier qu'une clef publique est bien celle d'une personne identifiée

— Contient les informations suivantes :

- Clef publique
- Nom du propriétaire de la clef (peut être une personne, mais aussi une machine, un logiciel)
- La durée de validité du certificat
- D'autres informations (attributs)

— Certifié par une **Autorité de Certification**, qui certifie que dans un certificat le nom du titulaire est bien celui du propriétaire de la clé publique

— Cette certification prend la forme d'une signature du certificat avec la clef privée de l'AC.

Cette signature sera donc vérifiée en décodant le certificat (une empreinte) avec la clef publique de l'AC, qui doit être mise à disposition... dans le certificat de l'AC.

Certificats électroniques : utilisation

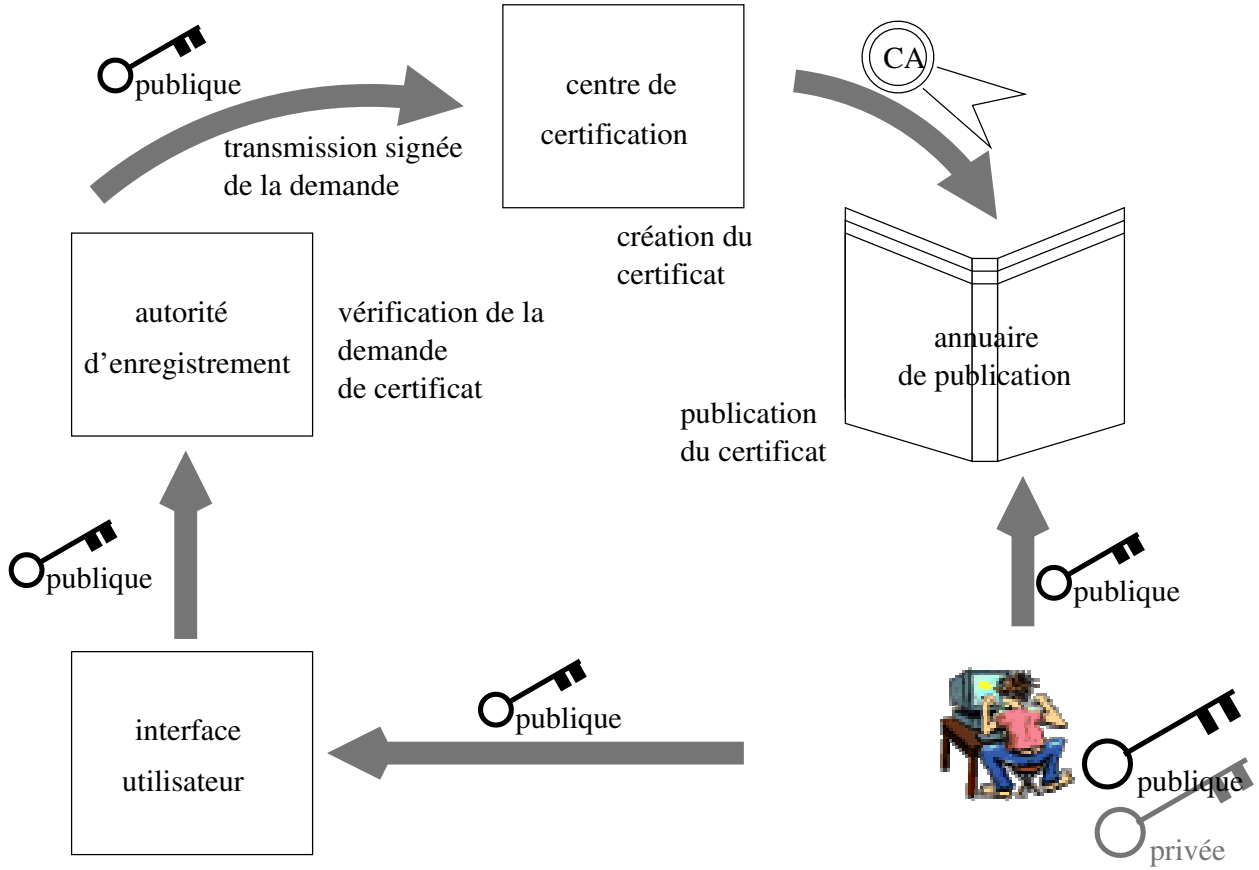
— Le certificat est un fichier (format X509), stocké dans l'espace de données de l'utilisateur, et lu par le navigateur / MUA

— Une AC est validée après chargement par ce navigateur / MUA du certificat de l'AC.

— Chiffrement / signature et séquestre des clefs

- chiffrement : un document chiffré doit pouvoir être déchiffré, même en cas de perte de la clé privée permettant de déchiffrer $\rightarrow$ il faut un service de recouvrement des clés privées, i.e. que les clés privées associées aux certificats soient conservées par un tiers.
- signature : la loi impose que la clé privée soit sous le contrôle exclusif du titulaire, cette clé privée ne peut donc être conservée par un tiers pour un service de recouvrement.

Certificats, création/gestion : l'Infrastructure de Gestion de Clefs



SSH

— SSH = Secure SHell

— Remplace les commandes en *r* (rlogin, rsh, ...), telnet, ftp, pour lesquelles l'authentification passe en clair (essayer `tcpdump -A`)

— Deux versions du protocole :

- Version 1 : surtout SSH 1.5 et 1.99
- Version 2 : open source (OpenSSH), plus sécurisée.

— Fonctionnalités :

- chiffrement fort (3DES, BlowFish)
- transfert X11 (X11 forwarding)
- transfert de port (tunnel SSH)
- authentification forte (par mot de passe transmis crypté, kerberos, jeu de clefs publique/privée au niveau machine et au niveau utilisateur)

SSH - architecture

— Binaires etc :

Partie utilisateur, requête de connexion : `/usr/bin/ssh`, `slogin`, `scp`, ...

Partie serveur, acceptation des connexions : `/usr/sbin/sshd`

Lancement : `service sshd start`

— Configuration

Serveur : `/etc/ssh/sshd_config`

Client : `/etc/ssh/ssh_config`

Clefs machine :

`/etc/ssh/ssh_host_key` +

`/etc/ssh/ssh_host_key.pub`

À comparer avec les clefs déjà connues :

`~/.ssh/known_hosts`

SSH - config serveur

— `/etc/ssh/sshd_config` (extraits)

```
Port 22
HostKey /etc/ssh/ssh_host_key
RandomSeed /etc/ssh/ssh_random_seed
PermitRootLogin no
X11Forwarding yes
RSAAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
UseLogin no
AllowHosts *.inria.fr
DenyHosts *.evil.org evil.org
KerberosAuthentication          no
AFSTokenPassing                 yes
```

SSH - config client

— `/etc/ssh/ssh_config` (extraits)

```
# This is ssh client systemwide configuration file.  This file provides
# defaults for users, and the values can be changed in per-user configu
# files or on the command line.
Host *
ForwardAgent yes
ForwardX11 yes
KerberosAuthentication no
KerberosTgtPassing yes
RSAAuthentication yes
PasswordAuthentication no
FallBackToRsh yes
UseRsh no
BatchMode no
IdentityFile ~/.ssh/id\_rsa
Port 22
Cipher idea
```

SSH - les clefs

— Génération des clefs :

```
$ ssh-keygen -t rsa1
```
(compatible SSH1)

```
$ ssh-keygen -t dsa
```
(SSH2)

Génère des fichiers :

`~/.ssh/id_rsa` (clef privée, protégée par les droits Unix et une passphrase)

`~/.ssh/id_rsa.pub` (clef publique)

— Connexions autorisées :

```
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

— Activation de la clef :

```
$ ssh-agent bash
$ ssh-add -i ~/.ssh/id_rsa
Need passphrase for /home/joe/.ssh/id_rsa
(joe@localhost).
Enter passphrase:
```

SSH - le tunnel SSH

— On raccorde un port standard sur un serveur, à un port temporaire sur le client, SSH faisant le lien (crypté) entre les deux.

— Exemple avec la connexion à un serveur POP :

```
$ ssh -N -f -L 2110:pop.outside.com:110 joe@ssh-server.outside.
$ ssh -N -f -L 2110:pop.inria.fr:110 joe@localhost
```

- `-N` : établissement d'un tunnel
- `-L` : syntaxe "port-local : serveur-distand : port-distant"
- `-f` : passe en background après connexion
- On connectera son MUA au port 2110 de localhost au lieu du port 110 de pop.inria.fr (exemple 2).

— Autre utilité : contourner un parfeu draconien . . .

SSH - le tunnel SSH inversé

— On peut inverser un tunnel SSH. Il n'est plus nécessaire de modifier la configuration du réseau distant.

— Exemple :

```
distant$ ssh -NR 22222:localhost:22 user@local
local$ ssh -p 22222 user@127.0.0.1
```

— Autossh permet de créer un script de démarrage pour que le tunnel soit recréé systématiquement.

Kerberos

— Un mécanisme d'authentification à clef secrète

— Développé au MIT. Version actuelle : Kerberos v.5.

— Gestion par un serveur d'authentification / gestion des clefs, le KDC (Key Distribution Center)

— Supporté par de nombreux systèmes : Unix, Unix/AFS, Windows (2000, XP, Server), ...

— Configuration sur le KDC : `/etc/krb5.conf`

Kerberos : authentification

— En s'identifiant sur un client, pour utiliser un serveur, un utilisateur saisit son nom et un mot de passe.
— Le client calcule (hash) une clef privée à partir du mot de passe.
— Le client envoie une demande de requète de service au KDC
— Le KDC envoie au client :
  - Une clef de session encodée avec la clef privée du client
  - Un "Ticket Granting Ticket" (TGT) encrypté avec la clef de session. Le TGT contient notamment l'identité de l'utilisateur et du client.

Kerberos : accès à un service

— Lorsque l'utilisateur requiert un service (exemple : NFS kerberisé), il envoie une requête au KDC, avec :
  - son TGT
  - son identité encryptée avec la clef de session

— Le KDC compare l'id du client avec celle dans le TGT, et envoie alors un ticket et une clef de session propres à la connexion client/serveur demandée, et encryptés avec la première clef de session.

— Le client contacte ensuite le serveur en envoyant le ticket et son identité encryptée avec la clef de session client/serveur.

— Le serveur confirme son identité en renvoyant un acquittement encrypté avec la clef de session, et accepte de fournir le service demandé au client.

— Pour mieux comprendre : lire le *Dialogue de Charon* (en anglais), `http://web.mit.edu/Kerberos/dialogue.html`

LDAP

— Annuaire d'organisation, qui reprend le modèle de X500

— Modèle client-serveur, port 389 (les clients incluent la plupart des MUA, etc.), avec notion de serveurs maître / esclaves avec réplication.
Les requêtes client peuvent inclure modification, suppression, etc.

— Interaction avec `sendmail`, les mécanismes d'authentification classiques (Kerberos,...)

— Support SSL/TLS, certificats,...

— Un enregistrement = un ensemble de champs de la forme `attribut=valeur`; les enregistrements sont organisés de manière arborescente.

— Chaque entrée est distinguée par un DN (Distinguished Name), par exemple : `cn=gilbert,o=inria`

— Requête : serveur + DN, par exemple : `ldap://ldap.inria.fr/o=inria,cn=gilbert`

LDAP

— Attributs courants :

- Attribut `objectClass`, donnant le rattachement à un schéma, `dcObject`, `organizationalUnit`, `person`, `organizationalPerson`, `inetOrgPerson`, `inetLocalMailRecipient`, ...
- `cn` : Canonical Name
- `o` : Organisation
- `c` : Country (pays)
- `uid`, `mail`, `userPassword`, ...

LDAP

— Import / export : format LDIF (Lightweight Directory Interchange Format)

```
dn: dc=inria,dc=fr
objectclass: dcObject
objectclass: organization
o: I.N.R.I.A.
dc: inria

dn: ou=people,dc=inria,dc=fr
objectclass: organisationalUnit
ou: people

dn: uid=gilbert,ou=people,dc=inria,dc=fr
objectclass: top
objectclass: person
uid: gilbert
cn: Frederic Gilbert
givenname: Frederic
sn: Gilbert
o: I.N.R.I.A.
userPassword: {SSHA}D3DT4BJyKicf+PJ1+eqkWMNRG/B28xt+
mail: frederic.gilbert@inria.fr
```

LDAP - Installation

— OpenLDAP, `/etc/init.d/ldap start`
— Configuration : `/etc/openldap/slapd.conf`

```
# inclusion du schema
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/misc.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
# specifications de la base
database ldbm
suffix "dc=metaparadigm,dc=com"
rootdn "cn=Manager,dc=metaparadigm,dc=com "
rootpw {crypt}mvRCcD3ajNmf2
directory /opt/openldap/var/openldap-ldbm
index objectClass eq
```

LDAP - Ajout et requêtes

— Ajout :

```
# ldapadd -D cn=Manager,dc=inria,dc=fr -W < init.ldif
Enter LDAP Password: xxxxx
adding new entry "dc=inria,dc=fr"
adding new entry "ou=people,dc=inria,dc=fr"
adding new entry "uid=gilbert,ou=people,dc=inria,dc=fr"
```

— Requêtes : filtres de recherche

- `(&(cn=Frederic Gilbert)(objectClass=posixAccount))`
- `(&(objectClass=inetOrgPerson)(!(o=Microsoft*)))`
- `(|(cn=Frederic*)(cn=Fred*))`
- Outil en ligne de commande, `ldapsearch` :

```
# ldapsearch -LLL -h ldap1-prd -b dc=inria,dc=fr \
  '(&(|(givenname=Frederic)(givenname=Fred))(objectClass=inetOrgPers
  cn mail
dn: uid=gilbert,ou=people,dc=inria,dc=fr
mail: frederic.gilbert@inria.fr
cn: Frederic Gilbert
```

LA JOURNALISATION (LES “LOGS”)

How should I know if it works ? That’s what beta testers are for. I only coded it.

Linus Torvalds

Journalisation

- De nombreux démons doivent transmettre une trace de leur exécution
- Les messages sont centralisés par un démon : `syslogd`
- Avantages :
  - une seule configuration
  - uniformité des messages
  - peu de code de trace dans chacun des démons
- Fichier de configuration : `/etc/syslog.conf`
- Chaque message est caractérisé par :
  - un type (ou “facilité”, *facility* en anglais)
  - une sévérité

Types de messages

— `kern` Messages du noyau

— `syslog` Messages de `syslogd` lui-même

— `mail` Messages du système de messagerie

— `lpr` Messages du système d'impression

— `auth` Messages d'authentification

— `daemon` Messages des démons

— `news` Messages du système de news

— `cron` Messages de `cron`

— `user` Messages des applications utilisateur

— `local0`, ..., `local7` Réservé pour utilisation locale

Sévérités

- `emerg` Crash imminent
- `alert` Erreur très grave
- `crit` Erreur grave
- `error` Erreur sans gravité
- `warning` Avertissements
- `notice` Messages normaux
- `info` Simples informations
- `debug` Messages de mise au point

Configuration de syslogd

- Chaque ligne de `/etc/syslog.conf` contient :
  - une liste de priorités (types de messages et sévérités)
  - une action à effectuer
- L'action peut spécifier :
  - un nom de fichier (`/...`)
  - une machine à qui transmettre le message (`@machine`)

    demande à ce que le syslog distant accepte les messages venant de machines distantes : lancement avec l'option `-r`
  - une liste d'utilisateurs
  - *pipe* nommé
- Pour tester : outil `logger`

  ```
  logger -p mail.info "message d'essai"
  ```
- Rotation des logs : `rotate-log`, `logrotate`

Exemple de configuration

```
*.err;kern.debug;auth.notice;mail.crit          /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
*.info;daemon.none;mail.none                    /var/log/messages
mail.debug                                      /var/log/mail.log
daemon.err                                      /var/log/daemon.err
daemon.info                                     /var/log/daemon.info
daemon.notice                                   /var/log/daemon.notice
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/cron/log
auth.*                                          /var/log/auth.log
*.alert                                         root
*.emerg                                         *
local2.*                                        /var/log/poppassd.log
local4.*                                        @soleil.uvsq.fr
*.emerg                                         |/etc/mail-to-root
```

Exemple de fichiers résultants (1)

/var/log/messages

```
Feb 14 16:10:00 atlas CRON[49500]: (root) CMD (/usr/libexec/atrun)
Feb 14 16:11:56 loiret printer: paper out
Feb 14 16:12:43 atlas sshd[49743]: connect from bigdaddy.csi.uvsq.fr
Feb 14 16:12:46 loiret printer: error cleared
Feb 14 16:15:00 atlas CRON[49819]: (root) CMD (/usr/libexec/atrun)
Feb 14 16:16:13 atlas mountd[151]: mount request from 193.51.26.61
for non existent path /usr/lib/X11/ncd
Feb 14 16:16:14 atlas mountd[151]: mount request denied from
193.51.26.61 for /
```

Exemples de fichiers résultants (2)

/var/log/daemon.info

```
Feb 14 16:18:54 atlas inetd[188]: tftp from 193.51.26.59
Feb 14 16:18:55 atlas mountd[151]: mount request denied from
193.51.26.59 for /
Feb 14 16:18:55 atlas inetd[188]: tftp from 193.51.26.59
Feb 14 16:18:55 atlas mountd[151]: mount request denied from
193.51.26.59 for /
Feb 14 16:18:55 atlas inetd[188]: tftp from 193.51.26.59
Feb 14 16:18:57 atlas inetd[188]: pop3 from 193.51.26.3
Feb 14 16:19:02 atlas inetd[188]: pop3 from 193.51.26.9
Feb 14 16:19:05 atlas inetd[188]: pop3 from 193.51.26.10
Feb 14 16:19:15 atlas last message repeated 3 times
```

Comptabilité utilisateur

- La comptabilité (en anglais, *accounting*) consiste à sauvegarder :
  - les traces de connexions
  - les commandes exécutées
- Plusieurs fichiers sont utilisés :
  - `utmp` : Connexions en cours
  - `wtmp` : Connexions ayant eu lieu
  - `acct`, `pacct` : Commandes exécutées

Comptabilité BSD

- Activation : `accton fichier`
- Examen :
  - `who` : Connexions en cours (lit `utmp`)
  - `last` : Connexions passées (lit `wtmp`)
  - `lastcomm` : Commandes exécutées
- Consommation CPU : `sa [options]`
  - `-m` : Synthèse par utilisateur
  - `-s` : Synthèse par commande + purge
- Connexions : `ac [options] [utilisateurs]`
  - `-p` : Synthèse par utilisateur
  - `-d` : Synthèse par jour

Exemples (1)

```
# last

root     ttyp1  atlas           Mon Feb 14 16:36   still logged in
fradang  ttyp1  193.51.25.3     Mon Feb 14 16:13 - 16:19 (00:05)
jmorea   ttyp0  jungle          Mon Feb 14 15:33   still logged in
vdc      ttyp0  193.51.25.115   Mon Feb 14 14:07 - 14:15 (00:08)
darje    ttyp2  bourgogne       Mon Feb 14 13:53 - 15:14 (01:21)
darje    ttyp2  bourgogne       Mon Feb 14 13:47 - 13:47 (00:00)
jmorea   ttyp1  jungle          Mon Feb 14 12:26 - 15:21 (02:55)
dntt     ttyp0  193.51.25.3     Mon Feb 14 11:54 - 13:54 (01:59)
dntt     ttyp0  193.51.25.3     Mon Feb 14 11:25 - 11:26 (00:01)
jburet   ftp    164.138.210.210 Sun Feb 13 23:13 - 23:33 (00:19)
amaran   ftp    193.51.26.17    Sun Feb 13 18:40 - 18:41 (00:00)
  ...
```

Exemples (2)

```
# lastcomm

sh       -F    fschlo      __          0.00 secs Mon Feb 14 16:32
sh       -     fschlo      __          0.00 secs Mon Feb 14 16:32
sh       -F    fschlo      __          0.00 secs Mon Feb 14 16:32
bash     -F    abidaud     ttyp5       0.00 secs Mon Feb 14 16:32
rm       -     abidaud     ttyp5       0.00 secs Mon Feb 14 16:32
ls       -     abidaud     ttyp5       0.00 secs Mon Feb 14 16:31
rshd     -S    root        __          0.02 secs Mon Feb 14 16:30
ksh      -S    cat         __          0.00 secs Mon Feb 14 16:30
pstat    -     cat         __          0.12 secs Mon Feb 14 16:30
rshd     -S    root        __          0.00 secs Mon Feb 14 16:30
xdm      -SF   lsavar       __          0.05 secs Mon Feb 14 12:35
bash     -X    lsavar      ttyp1       0.03 secs Mon Feb 14 12:39
  ...
```

(`S` = superuser, `F` = forked, `D` = décédé avec core, `X` = terminé par un signal)

Exemples (3)

```
# sa -m
root   77740306   5707.10cpu  113984412tio 23383018084k*sec
...
card     395414  17191.37cpu  179112687tio   320990375k*sec
...

# sa -s
 695190 6364383.82re  18439.48cp     34avio    382k
   2063   19440.61re  14199.00cp   8484avio      2k   ftp
   6039    2446.77re    473.90cp    547avio    518k   pwd_mkdb
  38562      86.21re    472.61cp      2avio    951k   cat
  38392    6683.32re    377.77cp      3avio    449k   dialog
    223     354.57re    314.74cp   1394avio     11k   rcp
...
```

Exemples (4)

```
# ac -p
        cty                    3.21
        card                  59.17
        sapin                  1.59
        cat                   23.13
        ftp                  955.11
        total               1042.20

# ac -d
  ...
Feb 10  total       109.55
Feb 11  total       140.15
Feb 12  total        71.94
Feb 13  total        39.11
Feb 14  total        37.76
```

Comptabilité System V (1)

- Utilitaires présents dans `/usr/lib/acct`
- Activation : `/usr/lib/acct/startup`
- Désactivation : `/usr/lib/acct/shutacct`
- Traitement : `/usr/lib/acct/runacct`
  - Doit être exécuté périodiquement (1 fois par jour via `cron`)
  - Traite les fichiers de comptabilité
  - Remet à zéro les fichiers de comptabilité
  - Produit des rapports synthétiques dans `/var/adm/acct`

Comptabilité System V (2)

- Utilisation des disques :
  `/usr/lib/acct/dodisk`
  - résultat dans `/var/adm/acct/nite/disktacct`
- Vérification des fichiers de comptabilité :
  `/usr/lib/acct/ckpacct`
  - doit être exécuté fréquemment (par `cron`)
  - vérification de la taille des fichiers
  - décomposition en plusieurs si besoin
  - suspension de la comptabilité si saturation de `/var`

Types de comptabilité

- Solaris 2 : System V
- HP-UX : System V
- IRIX : System V
- SunOS : System V + commandes BSD
- OSF/1 : System V (`/usr/sbin/acct`)
- FreeBSD : BSD
- Linux : BSD

LES SAUVEGARDES

Only wimps use tape backup : real men just upload their important stuff on ftp, and let the rest of the world mirror it.

Linus Torvalds

Sauvegardes

- Principe : sauvegarde périodique des fichiers modifiés
- Restaurations :
  - mise à jour du système
  - crash disque
  - erreur utilisateur
- Supports de sauvegarde :
  - disquettes
  - cartouches magnétiques
  - WORM
  - disques magnétiques, magnéto-optiques
  - CD, DVD (voir `dar`, `Partimage`)
  - Produits : EMC Networker, . . .(robots, gestion client/serveur).
  - Sur disques : baies RAID, NAS, etc.

Politique de sauvegarde

- Sauvegardes complètes
- Sauvegardes incrémentales :
  - sauvegarde des fichiers modifiés
  - la restauration peut nécessiter plusieurs cartouches
- Exemples de politiques de sauvegarde :
  - sauvegarde complète chaque jour
  - sauvegarde complète chaque semaine, incrémentale chaque jour
  - tours de Hanoï
- Sauvegardes sur disques : sauvegardes incrémentales au niveau bloc, possibilité de faire une sauvegarde totale puis un nombre illimité d'incrémentales.

Outils de sauvegarde

- `dump` et `restore` (`ufsdump` et `ufsrestore` sous Solaris 2)
  - rapide (interprétation de la structure du système de fichiers)
  - traitement de tous les types de fichiers
  - traitement des trous
  - sauvegarde de systèmes de fichiers complets (sauf sous SunOS, Solaris 2, et Linux)
  - format non normalisé
- `tar` et `cpio`
  - sauvegarde d'arborescences
  - formats normalisés

Fonctionnement de dump

- Quatre passes
  - recherche des fichiers à sauvegarder
  - recherche des répertoires à sauvegarder
  - sauvegarde des répertoires
  - sauvegarde des fichiers
- Format utilisé :
  - en-tête (description de la sauvegarde)
  - liste des i-nœuds sauvegardés
  - i-nœuds et contenus des répertoires
  - i-nœuds et contenus des fichiers

Fonctionnement de dump

- Syntaxe :

  `dump [options] fichier_spécial`

- Options :
  - `0`, `1`, `2`, ..., `9` Niveau de sauvegarde
  - `b blocage` Facteur de blocage
  - `s taille` Taille de la bande en pieds
  - `f fichier` Périphérique de sauvegarde
  - `d densité` Densité en BPI
  - `u` mise à jour du fichier `/etc/dumpdates`

Exemple

```
# dump 0ufB /dev/st0 120000 /dev/hda2

DUMP: Date of this level 0 dump: Mon Dec  9 00:37:55 1996
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/hda2 (/) to /dev/st0
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 95430 tape blocks.
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: 61.01% done, finished in 0:03
DUMP: DUMP: 95731 tape blocks
DUMP: level 0 dump on Mon Dec  9 00:37:55 1996
DUMP: DUMP IS DONE
```

Restauration

— Syntaxe :

```
restore [options] [fichiers]
```

— Options :

— `b blocage` Facteur de blocage

— `i` Restauration interactive

— `r` Restauration complète

— `x` Restauration des fichiers spécifiés

— `t` Liste des fichiers sauvegardés

— `f fichier` Périphérique de sauvegarde

— `v` Mode verbeux

Exemple (1)

```
# restore rvf /dev/st0

Verify tape and initialize maps
Tape block size is 32
Dump   date: Mon Dec  9 00:37:55 1996
Dumped from: the epoch
Level 0 dump of / on bbj:/dev/hda2
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Make node ./dev
Make node ./etc
  ...
```

Exemple (2)

```
Extract new leaves.
Check pointing the restore
Create symbolic link ./bin->/usr/bin
extract file ./usr/share/zoneinfo/posix/Asia/Ishigaki
...
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
```

Restauration interactive

- Commandes de pseudo-navigation dans l'archive
- Commandes
  - `help`
  - `add [fichiers]`
  - `cd répertoire`
  - `ls`
  - `delete [fichiers]`
  - `extract`
  - `quit`

Exemple (3)

```
# restore ivf /dev/st0
Verify tape and initialize maps
...
restore > ls
.:
    2 *./                11883  1/                   579  log
    2 *../               11893  2/                    11  lost+found/
   17  .Maelstrom-data   23846  3/                  8258  mnt/
  592  .Xauthority        4256  4/                 21737  msdos/
  593  .bash_history        12  bin                  522  opt
  594  .bashrc            1977  boot/              23970  proc/
  582  .gcalrc           23745  cdrom/             15812  sbin/
  595  .profile           3953  dev/                 633  src
  ...
```

Exemple (4)

```
restore > add etc
Make node ./etc
Make node ./etc/vga
restore > extract
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards towards the first.
Specify next volume #: 1
extract file ./etc/vga/libvga.config
Create symbolic link ./etc/utmp->/var/run/utmp
...
Add links
Set directory mode, owner, and times.
set owner/mode for '.'' [yn] n
```

Utilisation de tar

- `tar [options] [fichiers]`
- Options
  - `c` Création d'archive
  - `x` Extraction de fichier
  - `t` Liste des fichiers
  - `f fichier` Périphérique de sauvegarde
  - `b blocage` Facteur de blocage
  - `v` Mode verbeux
- Options de GNU-tar (Linux, BSD)
  - `Z`, `z` Compression (gzip), `j` (bzip2)
  - `M` Multi-volumes
  - `p` Préservation des droits Unix

Exemple (1)

— Création :

```
# tar cvf /dev/st0 /etc
tar: Removing leading / from absolute path names in the
archive
etc/
etc/mtab
etc/mail.rc
etc/group
etc/passwd
etc/HOSTNAME
...
```

Exemple (2)

— Liste :

```
# tar tvf /dev/st0
drwxr-xr-x root/wheel        0 Dec  9 00:25 1996 etc/
-rw-r--r-- root/wheel      204 Dec  9 00:25 1996 etc/mtab
-r--r--r-- bin/bin         102 Jun 18 23:42 1995 etc/mail.rc
-r--r--r-- bin/bin         383 May 12 17:47 1996 etc/group
-rw-r--r-- root/wheel      517 Jun 14 23:52 1996 etc/passwd
-r--r--r-- bin/bin           4 Oct  3 17:06 1993 etc/HOSTNAME
...
```

Exemple (3)

— Extraction :

```
# tar xvf /dev/st0
etc/
etc/mtab
etc/mail.rc
etc/group
etc/passwd
etc/HOSTNAME
etc/brc
...
```

Exemple (4)

— Dans un fichier :

```
# tar cvf /tmp/home.tar /home
home/
home/joe
home/joe/.cshrc
...
# tar xvf /tmp/home.tar
home/
home/joe
home/joe/.cshrc
...
```

— Dans un fichier compressé :

```
# tar zcvf /tmp/home.tar.gz /home
...
# tar zxvf /tmp/home.tar.gz
...
```

Exemple (5)

— Création et extraction simultanées :

```
tar cpf - . | (cd /ailleurs ; tar xpf -)
```

— Création et extraction simultanées à distance :

```
tar cpf - . | rsh otherhost "(cd /ailleurs ; tar xpf -)"
tar cpf - . | ssh -x otherhost "(cd /ailleurs ; tar xpf -)"
```

LES PROCESSUS PÉRIODIQUES

Intelligence is the ability to avoid doing work, yet getting the work done.

Linus Torvalds

Processus périodiques

- Exécution de processus :
  - de manière périodique
  - à date et heure fixes
- Deux mécanismes :
  - `cron`
  - `at`, `batch`
- Traitements périodiques :
  - `cron` est lancé au démarrage
  - il lance les processus définis dans les `crontab`
- Synchronisation après arrêt :
  - `anacron` : Anacronistic Cron (Linux)

Crontab

- Modèle BSD : un seul fichier
  - `/etc/crontab`
  - `/usr/lib/crontab`
- Modèle System V : une crontab par utilisateur
  - `/var/spool/cron/`
  - `/var/cron/`
  - manipulation par la commande `crontab`
- Modèle Linux (certaines distributions) :
  - scripts dans répertoires `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, `/etc/cron.monthly`.
  - entrées dans la crontab globale pré-positionnées pour exécuter ces scripts
- Liste des utilisateurs autorisés : `cron.allow`
- Liste des utilisateurs non autorisés : `cron.deny`

Format d'une crontab

— Format de chaque ligne :

1. minute (0 . . . 59)
2. heure (0 . . . 23)
3. jour du mois (1 . . . 31)
4. mois (1 . . . 12 ; jan . . . dec)
5. jour de la semaine (0, 1 . . . 7 ; sun, mon . . . sun)
6. nom d'utilisateur (si crontab globale)
7. commande à exécuter

— Valeurs :

  — * : toutes
  — v1,v2,v3,... : liste de valeurs
  — v1-v2 : intervalle de valeurs
  — */v1 : toutes les valeurs divisibles par v1 (récent)

Exemple

```
15 03 * * * find / -name .nfs\* -mtime +7
     -exec rm -f {} \; -o -fstype nfs -prune
05 04 * * 6 /usr/local/etc/newsyslog >/dev/null 2>&1
15 04 * * * find /var/preserve/ -mtime +7 -a
     -exec rm -f {} \;
00  * * * * /usr/lib/acct/ckpacct
00 07 * * 1-6 /usr/lib/acct/dodisk
15 07 * * 1-6 /usr/lib/acct/runacct
     2>/usr/adm/acct/nite/fd2log
30 07 01 * * /usr/lib/acct/monacct
00,20,40 * * * * /usr/local/etc/check_daemon rpc.mountd -n
00,15,30,45 * * * * /usr/local/etc/check_daemon in.named
01,16,31,46 * * * * /usr/local/etc/check_daemon
     /usr/local/bin/xntpd
```

Exécution à dates fixes

- Lancement :

  `at [options] heure [date] [commande]`

  `batch [options] [commande]`

  Par exemple :

  `at 17:45 12/31/2007`

  `at teatime tomorrow`

- File d'attente :

  `atq`

- Suppression dans la file d'attente

  `atrm numéro|-`

- Exécution des travaux :
  - intégré dans `cron` (exemple : SunOS)
  - exécuté par `atrun`, lui-même lancé par `cron` (exemple : *BSD, Linux)

L'IMPRESSION

I'm always right. This time I'm just even more right than usual.

Linus Torvalds

Gestion des imprimantes

- Deux modèles
- BSD :
  - `lpr`
  - `lpd`
  - `lpc`
- System V
  - `lp`
  - `lpsched`
  - `lpadmin`
  - `lpshut`
  - ...
- Des successeurs :
  - `lprng`
  - `cups`

Impression BSD

- Imprimantes gérées par le démon `lpd`
- Impression par la commande `lpr`
- Impression d'un fichier :
  - `lpr fichier`
  - transmission du fichier à `lpd` par `lpr`
  - Traitement par `lpd`
- Sélection de l'imprimante :
  - option de `lpr`
  - `-Pimprimante`

Fonctionnement de lpd

- Fichier de configuration : `/etc/printcap`
- Attente de requête d'impression
- Deux possibilités :
  - imprimante distante : transmission à une autre machine
  - imprimante locale : mise en file d'attente
- File d'attente locale :
  - fichier de contrôle (`cf*`)
  - fichier de données (`df*`)

Le fichier /etc/printcap

- Définition des imprimantes
- Chaque imprimante est définie par une suite de champs
- Principaux champs :
  - `sd=répertoire` : répertoire contenant la file d'attente
  - `lp=fichier_spécial` : périphérique de l'imprimante
  - `lf=fichier` : fichier contenant les traces des erreurs
  - `af=fichier` : fichier de comptabilité (pac)
  - `rm=machine`,
    `rp=imprimante` : localisation de l'imprimante
  - `if=programme`,
    `nf=programme`,
    `of=programme` : filtres

Exemples (1)

```
laser_pourrie:\
     :lp=/dev/ttya:sd=/var/spool/laser_pourrie:sh:\
     :lf=/var/log/laser_pourrie:\
     :af=/var/adm/laser_pourrie:\
     :br#9600:rw:fc#0000374:fs#0000003:xc#0:\
     :xs#0040040:mx#0:sf:sb:\
     :if=/usr/local/lib/filtre:

laser:\
     :lp=:rm=laser310:sd=/var/spool/laser:sh:\
     :lf=/var/log/laser:af=/var/adm/laser:
```

Exemples (2)

— Imprimante locale, raccordée au serveur srvimpr.uvsq.fr

```
# nom court de l'imprimante locale
l0929b:\
        # nom du répertoire de la file d'attente (sd=spool directory)
        :sd=/var/spool/lpd/l0929b:\
        # la taille maximum du fichier est illimitée (car 0)
        :mx#0:\
        # pas de page de séparation
        :sh:\
        # nom du fichier spécial pour printer locale
        :lp=/dev/lp0:\
        # nom du fichier de traitement du fichier
        :if=/var/spool/lpd/l0929b/filter:
```

Exemples (3)

— Imprimante distante

```
# nom court de l'imprimante REMOTE
l0929b:\
        # nom du répertoire de la file d'attente
        :sd=/var/spool/lpd/l0929b:\
        ..........................
        # nom du serveur d'impression  distant (rm=remote machine)
        :rm=srvimpr.uvsq.fr:\
        # nom de l'imprimante distante (rp=remote printer)
        :rp=l0929b:\
```

Gestion de la file d'attente

- Affichage : `lpq`
- Suppression : `lprm`
- Gestion par l'administrateur : `lpc`
  - gestion interactive
  - plusieurs commandes

La commande lpc (1)

- Commande de base : `help`
- Activation :
  - `enable imprimante|all`
  - `disable imprimante|all`
- Démarrage :
  - `start imprimante|all`
  - `stop imprimante|all`
- Activation et démarrage :
  - `up imprimante|all`
  - `down imprimante|all [message]`

La commande lpc (2)

- Arrêt de l'impression : `abort imprimante|all`
- Suppression de la file d'attente : `clean imprimante|all`
- Passage en tête :
  - `topq imprimante numéro`
  - `topq imprimante utilisateur`
- Etat : `status [imprimante]`

Impression System V

- Impression gérées par `lpsched`
- Impression par la commande `lp`
- Plusieurs commandes d'administration :
  - dans `/usr/lib`
  - dans `/usr/sbin` (Solaris 2)
- Deux types de destinations :
  - imprimantes
  - classes
  - paramètre de `lp : -d destination`

Configuration des imprimantes

- Une commande : `lpadmin`
- Options :
  - `-p`*`imprimante`* : imprimante à configurer
  - `-v`*`fichier_spécial`* : périphérique auquel est connecté l'imprimante
  - `-e`*`imprimante`* : copie de l'interface
  - `-m`*`modèle`* : copie de l'interface depuis le modèle spécifié
  - `-i`*`programme`* : spécification de l'interface
  - `-c`*`classe`* : classe
  - `-d`*`destination`* : sélection de l'imprimante par défaut
  - `-x`*`destination`* : suppression
  - `-r`*`classe`* : suppression de la classe

Gestion de la file d'attente

- Suppression :
  - `cancel numéro`
  - `cancel destination`
- Désactivation :
  - `reject [-rmessage] destination`
  - `disable [-rmessage] destination`
- Activation :
  - `accept destination`
  - `enable destination`
- Etat : `lpstat -pdestination`

CUPS (1)

— Common Unix Printing System, surtout commun à Linux

— Plutôt orienté “end user” : interface graphique, pour éviter la complexité du printcap

— Basé sur un ensemble de “locations” avec des droits spécifiques

— Fichier de configuration, `/etc/cupsd/cupsd.conf`

```
ServerName cupserver
ServerAdmin root@cupserver
AccessLog /var/log/cups/access_log
ErrorLog  /var/log/cups/error_log
LogLevel  info
MaxClients 100
BrowseAddress @IF(dc0)

<Location />
Order Deny,Allow
Deny From All
Allow From 192.168.0.*
</Location>
.........
```

CUPS (2)

— “Locations”, ou chemin d’accès pour différents types d’objets ou d’opérations :

`/`	toutes les opérations (statut des jobs, imprimantes,...)
`/admin`	toutes les opérations d’administration (ajout/suppression imprimante,....)
`/admin/conf`	à la configuration de CUPS (`cupsd.conf`,...)
`/classes`	toutes les classes
`/classes/name`	nom des classes
`/jobs`	les jobs
`/jobs/id`	leurs numéros d’identifiant
`/printers`	les imprimantes
`/printers/name`	leurs noms
`/printers/name.ppd`	les fichiers de description d’imprimantes

Types de systèmes d'impression

- Solaris 2 : System V
- HP-UX : System V
- IRIX : System V
- SunOS : BSD
- OSF/1 : BSD (+ commandes System V)
- *BSD : BSD
- Linux : BSD, CUPS

DÉPANNAGE, RÉSOLUTION DE PROBLÈMES

Talk is cheap. Show me the code.

Linus Torvalds

Pannes et dépannage : Écrasement du MBR

— *Qu'est-ce que c'est ?* le Master Boot Record, localisé dans les premiers secteurs du disque dur, conditionne le démarrage de la machine.
C'est notamment le siège de Lilo.

— *En quel cas est-il écrasé ?* lors d'un appel de `lilo` avec de mauvais paramètres, lors d'une réinstallation de Windows en double boot, ...

— *Comment réparer ? RedHat*

- booter sur un support externe (CD 1 de la RedHat en mode `linux rescue`)
- monter la partition `/` du disque de la machine après avoir créé un device approprié :

```
mknod hda
mkdir /mnt2
mount /dev/hda1 /mnt2
```

- lancer Lilo en indiquant que la "racine" à considérer est le disque interne :

```
lilo -v -r /mnt2
```

Pannes et dépannage : Écrasement du MBR

— *Comment réparer ? Mandrake*

- booter sur un support externe (CD 1 de la Mandrake en mode `linux rescue`, demander un shell)
- lancer la commande `drvinst` qui identifie les éléménts matériels, charge les drivers et crée les devices
- monter la partition `/` du disque de la machine :

```
mkdir /mnt2
mount /dev/hda1 /mnt2
```

- changer le répertoire `root` de la machine :

```
chroot /mnt2
```

- lancer Lilo qui va agir sur cette “racine” :

```
lilo -v
```

Pannes et dépannage : Blocs défectueux

— Blocs défectueux

Type d'erreur : une erreur peut être “soft” ou “hard”. Une erreur “soft” peut généralement être corrigée en ré-écrivant le bloc concerné :

— Identifier le numéro du bloc défectueux sur la partition, *n* (on peut utiliser les logs, ou le programme `badblocks`)

— Lire le contenu de ce bloc (on admet une taille de bloc de 1024 octets) :

```
dd if=/dev/hda1 of=/tmp/bloc.dat bs=1024 skip=n-1
count=1 conv=noerror
```

— Ré-écrire le contenu du bloc :

```
dd if=/tmp/bloc.dat of=/dev/hda1 bs=1024 seek=n-1
count=1 conv=noerror
```

— Super-bloc défectueux

Si `fsck` ne passe plus, executer `e2fsck -b 8193 /dev/hda3`

Éventuellement ré-essayer en augmentant le nombre de 8192 à chaque fois, ou en affichant les superblocs en faisant `mke2fs -n`.

Pannes et dépannage : Tracage des processus

— Tracer les processus

  — Sun : `truss`

  — Linux : `strace`

  La commande `strace` permet d'imprimer les appels système au fur et à mesure de l'exécution d'un programme.

  - Au lancement d'un programme :

    `strace -o /tmp/outfile -f ./`*`programme`*

  - Sur un processus existant :

    `strace -o /tmp/outfile -f -p 13657`

  L'option "`-f`" permet de tracer également les processus fils issus de l'appel système `fork()`.

— Visualiser les librairies partagées : `ldd -r `*`programme`*

Le répertoire /proc

— Le répertoire `/proc` contient une image, sous forme d'arborescence et de fichiers, de la mémoire noyau et de la mémoire utilisateur.

— C'est un répertoire virtuel (type `procfs`), point d'entrée vers des zones mémoire du noyau

— On y trouve notamment :

- un répertoire d'informations par processus en cours (programme exécuté, descripteurs ouverts,...)
- des informations sur l'état du système (paramétrages réseaux dans `net`, `modules` chargés, montages,...)
- des répertoires pour les périphériques de la machine, classés par type (bus `pci`, `scsi`,...).

— Utilisation en écriture :

- `echo 1 > /proc/sys/net/ipv4/ip_forward`

LE CLOUD COMPUTING

Le cloud computing

- Le concept remonte aux années 1950...
  - mainframes
  - ”the cloud”
  - Hébergeurs web
- Principes
  - Adaptation à la demande
  - Ouverture
  - Mutualisation et scalabilité
  - Paiement au prorata

Le cloud computing

- Différences avec un réseau informatique
  - Les tâches sont effectuées différemment
  - Notions d'élasticité
  - Pas d'investissements dans l'infrastructures
- Livraison de :
  - Logiciels
  - L'infrastucture
  - Le stockage
- Certains parlent de techniques marketing...
  - Richard Stallman

Le cloud computing

- Trois formes de cloud :
  - Cloud public
  - Cloud privé
  - Cloud hybride

Le cloud computing

- Services principaux
  - Iaas : infrastructure as a service
  - PaaS : platform as a service
  - SaaS : software as a service
- Autres
  - Data as a service
  - MPaaS : Business Process as a service
  - Daas : Desktop as a Service
  - NaaS : Network as a Service
  - STaaS : Storage as a Service

Le cloud computing

- Avantages
  - Solution économique
  - Evolutivité / élasticité
- Inconvénients
  - Quid de la sécurité du cloud ?
  - Dépendance sur la qualité du réseau
  - Complexité architecturale
  - Plus de garantie sur la confidentialité

Le cloud computing

- Principaux acteurs
  - Amazon
  - Citrix
  - Gandi
  - Google
  - IBM
  - Etc...

Le cloud computing

- Amazon Web Services (AWS)
  - Démarrage en 2006
  - 330 000 développeurs
- Grands clients
  - NASA
  - Netflix
  - CIA
- Architectures
  - Protocole HTTP
  - Architecture REST
  - Protocole SOAP
- Plusieurs services
  - Amazon Elastic Compute Cloud (EC2), Amazon Elastic Block Store (EBS), Amazon Simple Storage Service (S3), ...

Le cloud computing

- Amazon Elastic Compute Cloud (EC2)
  - Elasticité des serveurs
  - 40 000 serveurs
- Virtualisation Xen
  - Small Instance
  - Large Instance
  - Extra Large Instance
  - High-CPU Instance
  - High Memory Instance
  - ...