Introduction

INTRODUCTION

Contenu

- L'administrateur système
  - Le métier
  - Les pré-requis
  - Les aptitudes
- Matière abordée
- Présentation d'**Unix**
- Les sources d'informations

Le métier d'administrateur

- Au service des utilisateurs
- S'occupe des machines et offre
  - Disponibilité
  - Sécurité
  - Privacité
  - Efficacité
  - Confort

Le métier d'administrateur

- Pourquoi centraliser ?
  - Connaissance technique importante
  - Cohérence
  - Privacité
  - Les utilisateurs font leur métier
- De quoi s'occupe-t-il ?
  - S'occupe surtout du système
  - Mais aussi du matériel
  - Et des logiciels

Le métier d'administrateur

- En pratique, cela comprend :
  - Installer les machines et les systèmes
  - Ajouter et supprimer des utilisateurs
  - Ajouter et supprimer du matériel
  - Effectuer des sauvegardes
  - Installer de nouveaux logiciels
  - Vérifier le système

Le métier d'administrateur

Mais aussi :

- Effectuer les réparations diverses (matérielles et logicielles)
- Tenir à jour la documentation
- Evaluer et gérer la sécurité
- Aider les utilisateurs

Le métier d'administrateur

Il faut distinguer

- L'administration système
- L'administration des systèmes distribués
- L'administration réseau

Les pré-requis

- Etre un bon utilisateur
- Connaître les bases de la programmation sur le système
- Connaître un peu les rouages internes du système

Les aptitudes

- Gestion du stress
- Multi-tâches
- Souplesse horaire
- Bonne communication
- Ordre
- Ethique forte
- Serviabilité

SYSTÈMES ÉTUDIÉS

- Surtout **Unix**
- **Windows** : intégration avec **Linux**
- Pas les autres : **OS400** , l'OS des mainframes **IBM** ou celui du **Mac**

L'HISTOIRE D'UNIX

Pas de standard pour **Unix** . Existence de plusieurs **Unix** qui se ressemblent plus ou moins.

1969 Naissance dans les laboratoires **AT&T**

1976 Distribué gratuitement aux Universités

1977 Naissance de la version **BSD** de Berkeley (diverge de la version **SysV**)

1991 **Linux** créé par Linus Torvald

PRÉSENTATION DE SOLARIS

- Système d'exploitation de Sun Microsystems
- En fait, package incluant **SunOS**

  (un peu comme **Linux** est une distribution)
- **SunOS** 3 & 4 (**Solaris** 1) étaient très **BSD**
- **SunOS** 5 (**Solaris** 2 à 8) est très **SysV**

PRÉSENTATION DE LINUX

- Implémentation gratuite du système **Unix**
- Crée par L. Torvald à partir d'un travail de A. Tanembaum
- Désigne à proprement parler le noyau
- Une distribution **Linux** inclut un noyau et une série de logiciels ; le tout intégrés correctement

Les sources d'informations

Les livres

- "**Unix , Guide de l'administration**",

  Evi Nemeth, Garth Snyder, Scott Seebass and Trent R. Hein

Les sources d'informations

Le site internet

- http://users.skynet.be/Marco.Codutti/
  - Cette présentation
  - Mes notes
  - Des liens vers d'autres sources d'information

Les sources d'informations

L'aide locale

- Les **pages man**
- Les **pages info**
- Les **How-to**
- Les systèmes propriétaires

Les sources d'informations

Les sources Internet

- Les sites dédicacés
- Les **news**

Ces sources peuvent

- être dédiées à un OS
- ou à une distribution
- parler d'administration de plusieurs systèmes

Les pouvoirs du root

Les pouvoirs du root

Contenu

- Droits d'accès
- Le **super utilisateur**
- Choisir son mot de passe
- Devenir **root**
- Les pseudo-utilisateurs

INTRODUCTION

- Utilisateur : identifié par un nom (et un numéro)
- Chaque fichier et chaque processus appartient à **un** utilisateur
- L'utilisateur peut affiner les droits d'accès
- Le root a tous les droits

Droits d'accès

- Chaque fichier possède
  - un propriétaire
  - un groupe
- Les groupes sont définis dans **/etc/group**
- En fait, les utilisateurs et les groupes sont des nombres au niveau interne (**UID**, **GID**)
- Un processus est lancé avec les droits de la personne qui l'exécute
- Peut-être modifié avec le **setuid** bit.

LE SUPER UTILISATEUR

- Il s'agit de l'utilisateur avec le UID à 0
- Son nom est **root**
- Le seul à pouvoir se faire passer pour un autre
- Egalement le seul habilité à
  - créer des fichiers périphériques
  - modifier l'horloge système
  - manipuler les quotas
  - configurer le réseau
  - . . .

Choisir son mot de passe

- Ne pas le noter
- Il existe des programmes qui *découvrent* les mots de passe
- Choisir un bon mot de passe
  - Mélanger chiffre, lettres et ponctuations
  - Résumer une phrase
- Le changer régulièrement ou dès qu'un doute s'installe

DEVENIR ROOT

- On peut se connecter comme avec un autre compte
- Mieux : utiliser **su**
  - La personne doit d'abord s'identifier en tant que personne sans droit
  - Trace dans un journal
- Mieux encore : utiliser **sudo**
  - Contrôle plus fin de ce qui est permis

SUDO

- Permet d'exécuter une seule commande avec les pouvoirs du root
- On la retrouve de plus en plus
- Un fichier de configuration (**/etc /sudoers**)
- On y retrouve des entrées indiquant
  - Qui
  - Peut faire quoi
  - Sur quelle machine
  - Sous quel nom

SUDO

- Exemple

```
Host_Alias      DI = lit1 , lit2 , lit3
User_Alias      PARTTIMERS    = bostley , jwfox , crawl
User_Alias      FULLTIMERS    = millert , mikef , dowdy
Cmnd_Alias      DUMPS  = /usr/bin/mt, /usr/sbin/dump
Cmnd_Alias      KILL  = /usr/bin/kill
Cmnd_Alias      PRINTING   = /usr/bin/lprm
PARTTIMERS      DI = ALL
FULLTIMERS      ALL = NOPASSWD : ALL
operator        ALL = DUMPS , KILL , (hpadmin ) PRINTING
```

SUDO

- Avantages
  - Commandes enregistrées dans un journal
  - Déléguer des responsabilités à discrétion
  - Mot de passe root peu divulgué
  - Le fichier de configuration est centralisé
  - Reprent clairement qui peut faire quoi
  - On minimise la possiblité de laisser un session root ouverte

SUDO

- Inconvénients
  - Les utilisateurs avec un peu de responsabilité doivent veiller à la sécurité de leur propre compte
  - Soigner le configuration sinon c'est la porte ouverte aux failles (exemple : **sudo csh**)

PSEUDO-UTILISATEURS

- Le fichier **/etc /passwd** contient des utilisateurs qui ne sont pas des personnes physiques
- Reconnus par le système avec un statut particulier
- Citons **daemon**, **bin**, **nobody**

LES FICHIERS

LES FICHIERS

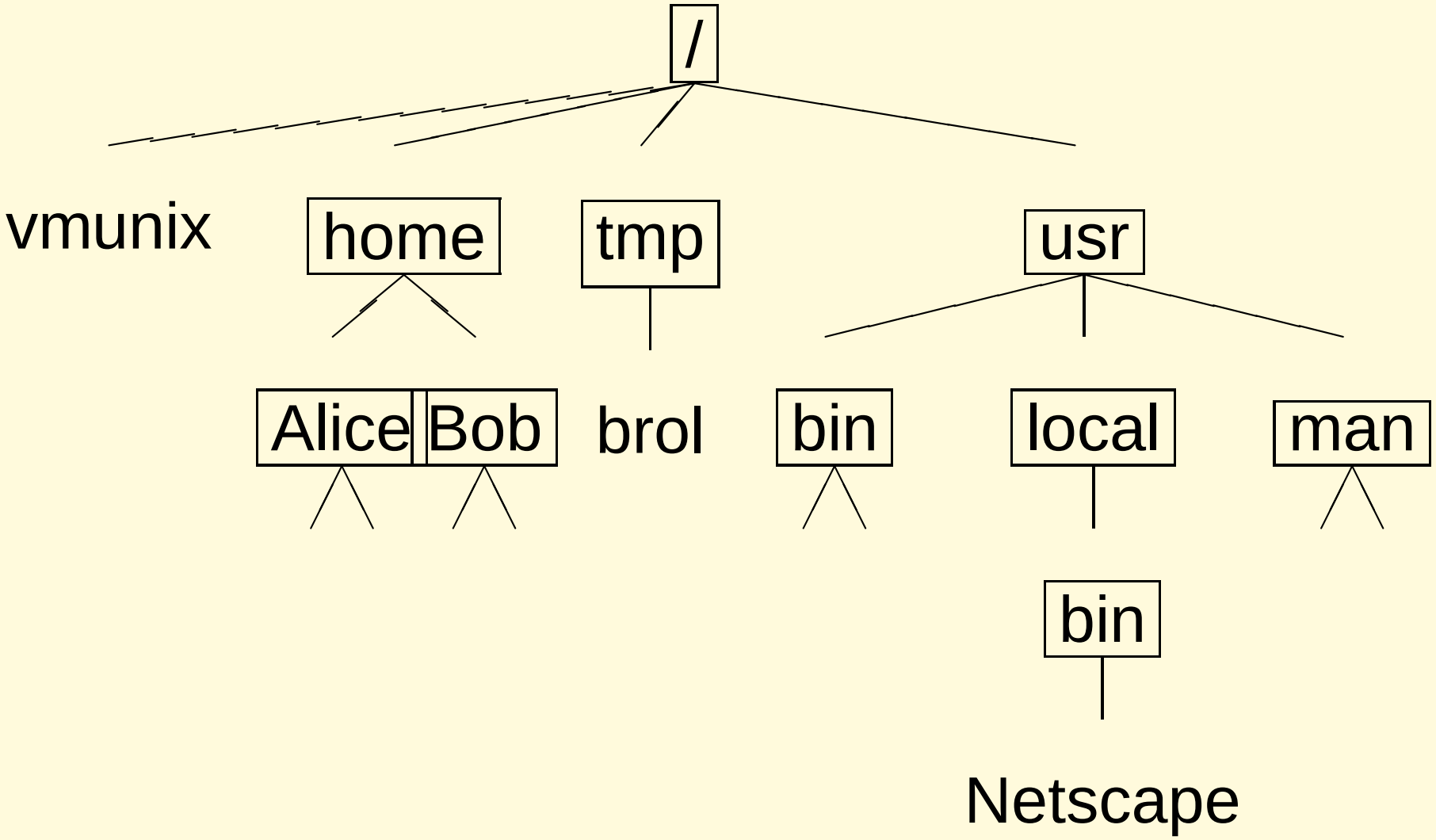
Contenu

- Structure / Partitions
- Occupation des disques
- Les types de fichiers
- Les permissions

INTRODUCTION

- Donne accès aux informations sur support permanents (fichiers)
- A été étendu à l'accès d'autres informations (pseudo-fichiers)
  - gestionnaires de périphériques
  - informations du noyau

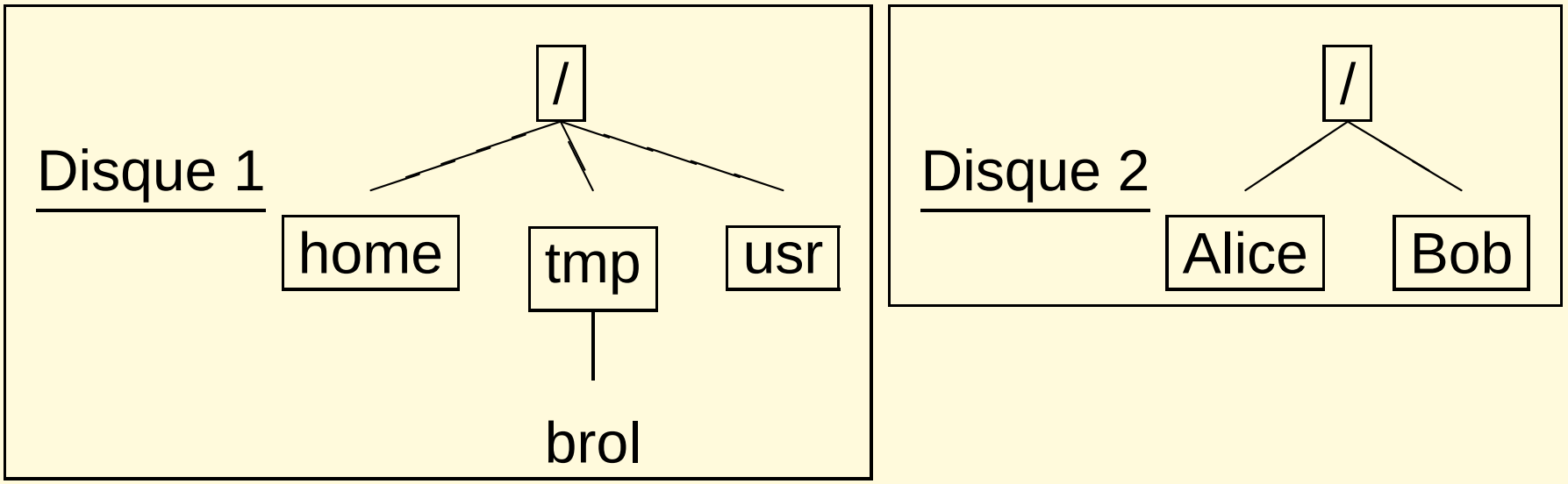
STRUCTURE

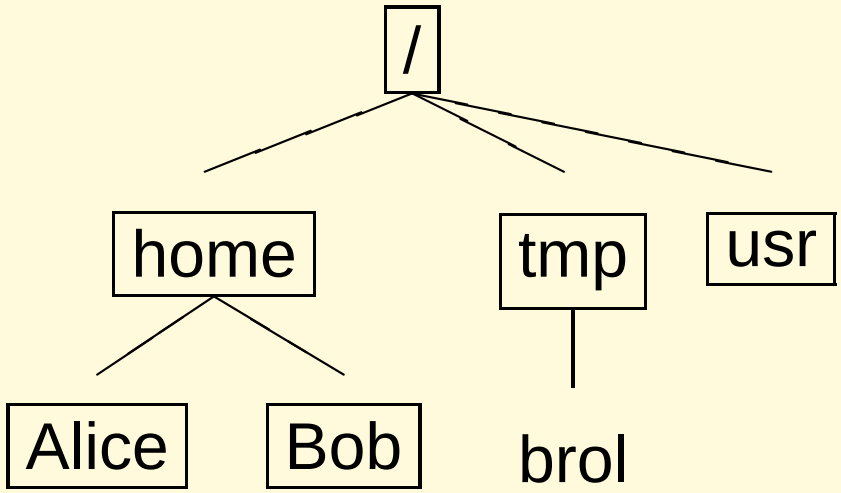

/
vmunix
home
tmp
usr
Alice
Bob
brol
bin
local
man
bin
Netscape

Les partitions

- Logiquement, un seul système de fichiers
- Physiquement, plusieurs systèmes qui sont **montés** ensembles

LES PARTITIONS





LES PARTITIONS

- La partition principale est celle contenant le noyau.
- Les autres peuvent être ajoutées à volonté à la hiérarchie (via **mount**) et enlevées (via **umount**)
- Un fichier (**/etc /fstab**) reprend la structure des partitions

LES PARTITIONS

- **mount**
  - On indique la partition et l'endroit où la monter
  - On peut donner des options (type de système de fichiers, lecture seule, . . .)
  - Exemple : `mount /dev/sd1c /home`
  - Sans paramètre, on obtient la liste des partitions montées

LES PARTITIONS

- Lors du démarrage, les systèmes de fichiers sont automatiquement montés
- Les informations sont reprises dans un fichier (**/etc /fstab** , **/etc /vfstab** ou encore **/etc /checklist**)
- Exemple : (sur **Linux**)

```
/dev /hda6  / ext3  defaults   1 1
/dev /hda3  /home  ext3  defaults   1 2
/dev /hda1  /mnt /nt  ntfs  (<-|)
  iocharset  =iso8859 -15, ro ,umask =0  0 0
/dev /hda7  swap  swap  defaults   0 0
```

LES PARTITIONS

- **umount**
  - On indique le dossier à démonter
  - Exemple : **umount /home**
  - Interdit si partition **occupée**
  - On peut le vérifier avec **fuser** ou **lsof**

LES PARTITIONS

- **fuser**
  - Permet d'identifier les processus utilisant un (des) fichier(s)
  - Existe de plus en plus mais avec des options différentes
  - Exemple : **fuser -mv /home** (extrait)

```
USER        PID  ACCESS   COMMAND
marco      2305  f.c ..   startkde
marco      2535  f.c ..   kalarmd
marco      2592  f.c ..   kile
marco      2594   ..c ..  bash
marco      2858  f....    xmms
```

Les partitions

- **lsof**
  - Identifie les fichiers utilisés par un processus
  - Existe sur certains systèmes (disponible gratuitement)
  - Exemple : **lsof -p 1** (extrait)

```
  FD     TYPE   DEVICE       SIZE      NODE   NAME
 cwd      DIR      3,6       4096         2  /
 rtd      DIR      3,6       4096         2  /
 txt      REG      3,6      31384    701828   /sbin /init
 mem      REG      3,6     539887    750724   /lib /ld -2.2.5.  so
 mem      REG      3,6    1167240    962882   /lib /i686 /libc.  so
  10u    FIFO      0,6                  733   /dev /initctl
```

LES PARTITIONS

- **df**
  - Informe sur l'espace utilisé sur les partitions
  - Options différentes sur **BSD** et **AT&T** .
  - Exemple : (**df -k** sur **AT&T**)

```
[marco@localhost      marco ] df
SysFichier    1K-blocs   Utilisé    Dispo .   Ut% Monté   sur
/dev /hda6     8594008   2465048   5692396   31%  /
/dev /hda3     5352592   2768880   2583712   52%  /home
/dev /hda5     8610040   3258644   4914032   40%  /mnt/ ml82
/dev /hda2     5379280   2063676   3315604   39%  /mnt/ windows
```

LES PARTITIONS

- **du**
  - Donne la taille des dossiers
  - Exemple :

```
root@lit1   :du  -s  /var /*
5000        /var /adm
790         /var /mail
345         /var /spool
1259        /var /tmp
```

  - Il faut avoir les permissions en lecture sur les dossiers

LES FICHIERS

Contenu

- Structure / Partitions
- Occupation des disques
- **Les types de fichiers**
- Les permissions
- L'organisation standard

Les types de fichiers

- Les dossiers
- Les fichiers ordinaires
- Les fichiers de périphériques
- Les tubes nommés et sockets
- Les liens symboliques
- ...

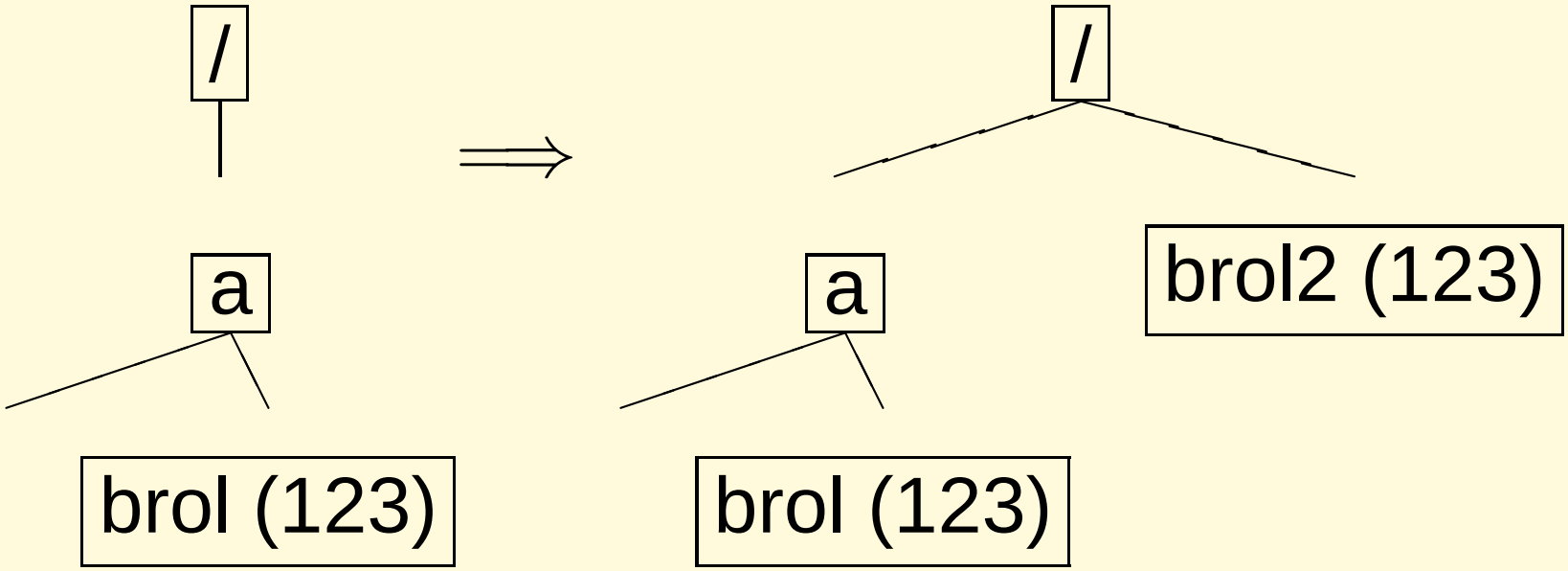
Les dossiers

- Fichier contenant une table associant un nom à un inode
- Exemple :

Nom	inode
.	123 *(directory courante)*
..	126 *(directory parent)*
vmunix	345
etc	12

LES LIENS

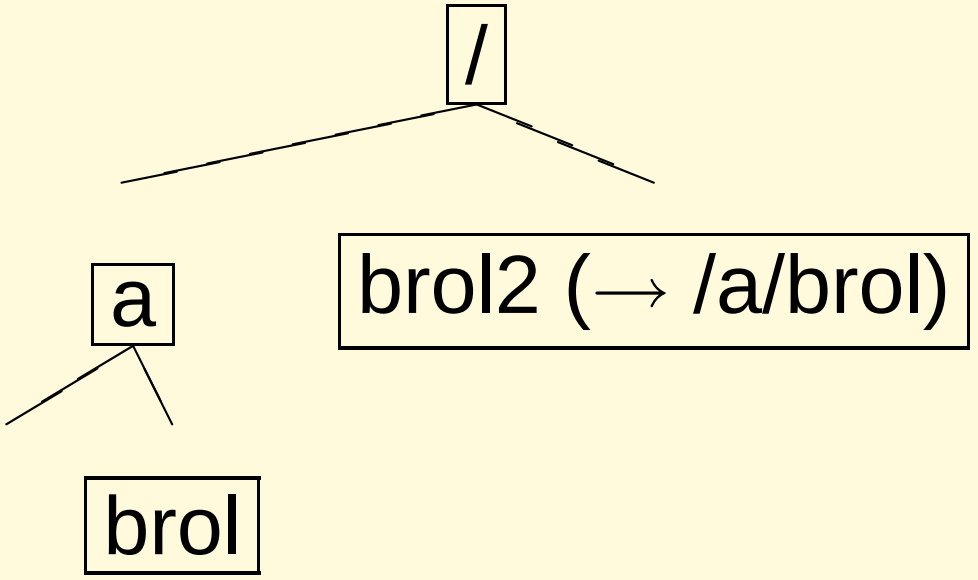
- Lien physique
- Exemple : `ln /a/brol /brol2`



- Le fichier n'existe qu'une fois avec 2 accès équivalents

Les liens

- Lien symbolique
- `ln -s /a/brol /brol2`



- On crée une **indirection**
- Attention à `cd /a; ln -s ./brol /brol2`

Les fichiers de périphériques

- Accès à un élément physique via un driver
- Tous dans **/dev**
- Numéro majeur + numéro mineur
- Exemple (extrait)

```
[marco@codec    tmp ]$ ll /dev /tty
crw -rw -rw -   1 root   root   5,  0 fév  2 22:41   /dev /tty
```

Contenu

- Structure / Partitions
- Occupation des disques
- Les types de fichiers
- **Les permissions**
- L'organisation standard

LES PERMISSIONS

- Chaque fichier appartient à un *propriétaire* et un *groupe*
- On lui associe des *permissions* représentées par 12 bits.

			proprio			*groupe*			*autres*		
setuid	setgid	sticky	r	w	x	r	w	x	r	w	x

Les permissions

Les effets de r, w et x

- Pour un fichier
  - **r** permet de lire le fichier
  - **w** permet d'écrire
  - **x** peut être exécuté

LES PERMISSIONS

Les effets de r, w et x

- Pour un dossier
  - **r** permet de voir le contenu
  - **w** permet d'écrire dans le dossier
  - **x** permet de traverser le dossier

LES PERMISSIONS

Exemples

brol	brol2	action
r		ls brol
x		cd brol
x	r	cat brol/brol2
wx		rm brol/brol2
x	w	vi brol/brol2
wx		touch brol/brol3
x	w	cat brol/brol3 >brol/brol2
wx		mv brol/brol3 brol/brol2

Les permissions

- Un **script** doit posséder à la fois le bit **x** et le bit **r**
- Un programme avec le **setuid** bit s'exécute avec les permissions du propriétaire du fichier contenant le programme
- Idem avec le **setgid** bit pour le groupe

LES PERMISSIONS

- **ls** affiche le contenu d'un système de fichiers

```
mcodutti@cso8    :ls  -lg
drwx ------   2  root   other      512   Jan  10   1996   Mail /
-rw-r--r --   1  root   other    42932   Dec  13  09:33   core
```

- La première colonne indique le type de fichiers et les permissions
- Comme type de fichier on a : d, l, c et b
- Les setuid, setgid et sticky apparaissent à la place du x.

LES PERMISSIONS

- **chmod** permet de modifier les permissions
- Syntaxe : **chmod perm fichier**
- où la permission peut être donnée
  - numériquement.

    Exemple : **chmod 4755 brol**
  - symboliquement. (syntaxe **who op perm**)

    Exemple : **chmod a+x brol**

Les permissions

- Modifier le propriétaire et le groupe
- `chown owner file`
- `chgrp group file`
- `chown owner :group file`

  (sur certains systèmes)
- Souvent réservé au **root** car dangereux

  (quotas par exemple)

Les permissions

- Par défaut,
  - Nouveau dossier : permission 777
  - Nouveau fichier : permission 666
- **umask** est une variable d'environnement/une commande qui contrôle les permissions des nouveaux fichiers/dossiers
- Spécifie les bits qui sont mis à 0 lors de la création
- Exemple : **umask 022**

L'ORGANISATION STANDARD DES FICHIERS

ORGANISATION STANDARD

- L'organisation des fichiers dans la hiérarchie a évolué au fil du temps
  - Evolution des besoins
  - Sensibilités différentes des responsables
- D'un système à l'autre il y a des différences
- Le **Filesystem Hierarchy Standard Group** tente d'imposer un standard
- C'est celui-là qu'on va expliquer

ORGANISATION STANDARD

Dichotomie définie par le FSHG

- Un fichier **partageable** est identique sur plusieurs machines de même architecture tournant le même OS
- Un fichier **variable** voit son contenu changer de façon non contrôlée

ORGANISATION STANDARD

- Dossiers typiques

	partageable	non partageable
statique	`/usr`	`/etc`
variable	`/var /mail`	`/var /lock`

ORGANISATION STANDARD

Le premier niveau

- Les dossiers non essentiels au démarrage devraient être dans une partition séparée

/bin	Exécutables essentiels au démarrage
/sbin	Idem pour commandes administrateurs
/boot	Nécessaire pour charger le noyau
/dev	Fichiers liés au périphériques
/etc	Confi guration propres à la machine

ORGANISATION STANDARD

/home	Fichiers des utilisateurs
/lib	Les librairies essentielles
/mnt	Point de montage temporaire
/opt	Logiciels non systèmes
/root	Les fichiers personnels du **root**
/tmp	Les fichiers temporaires
/usr	Section importante
/var	Section des fichiers variables

ORGANISATION STANDARD

Le dossier /usr

- Son rôle historique a un peu changé

/usr /bin	Exécutables non essentiels
/usr /include	Fichiers **include**
/usr /lib	Librairies non essentielles
/usr /local	Tout ce qui est local

Le dossier /usr

/usr /sbin	Les binaires réservés à l'administrateur mais pas nécessaires au démarrage
/usr /share	Tout ce qui est indépendant de l'architecture (mais pas de l'OS précis ni de sa version)
/usr /src	On y trouve tous les sources

LES UTILISATEURS

Les Utilisateurs

Contenu

- Le fichier des utilisateurs
- Le fichier des groupes
- Ajouter/supprimer un utilisateur
- Désactiver un utilisateur
- Le fichier **shadow**
- Les commandes
- Particularités FreeBSD

LE FICHIER DES UTILISATEURS

- Les utilisateurs sont repris dans

  /etc /passwd

- Lignes de la forme

```
login :passwd :uid :gid :GECOS :home :shell
```

- Exemple :

```
root :$1$og798BK :0:0: root :/root :/bin /bash
marco :$1$ErmZy8n :501:50: Codutti :/home /marco :/bin /bash
nobody :*:65534:65534: Nobody :/:/bin/sh
```

LE FICHIER DES GROUPES

- Les groupes sont repris dans **/etc /group**
- Lignes de la forme

```
group :passwd :gid :users
```

- Mot de passe plus guère utilisé (**newgrp**)
- Reprendre aussi le groupe principal

AJOUTER UN UTILISATEUR

- Avec un outil graphique (très lié à l'OS)
- Avec des commandes : **useradd**

  (ne fait pas tout)
- Manuellement

AJOUTER UN UTILISATEUR

- Ajouter un utilisateur manuellement
  - Ajouter une ligne au fichier **/etc /passwd**
  - Définir le mot de passe initial (**passwd**)
  - Adapter le fichier **/etc /group**
  - Créer le dossier personnel (**mkdir**)
  - Le remplir avec les fichiers de départ (**.bashrc** , **.profile** , . . .)
  - Ajuster les permissions (**chown**)
  - Vérifier !

SUPPRIMER UN UTILISATEUR

- Supprimer un utilisateur manuellement
  - Supprimer l'utilisateur dans **/etc /passwd** et **/etc /group**
  - Supprimer (sauvegarder) son dossier personnel
  - Supprimer (sauvegarder) ses autres fichiers
- Désactiver un utilisateur manuellement
  - Mettre un mot de passe impossible
  - Mettre un faux shell

LE FICHIER SHADOW

- Introduit pour pallier au problème de sécurité de **/etc /passwd**
  - Reprend le mot de passe
  - Lisible uniquement par root
  - Le champ dans **/etc /passwd** n'est plus utilisé
- On en a profité pour introduire un contrôle plus fin sur les mots de passe

LE FICHIER SHADOW

- Contient 9 champs
  - Nom utilisateur
  - Mot de passe crypté
  - Date dernière modification
  - Nb minimal de jours entre 2 modifications
  - Nb maximal de jours entre 2 modifications
  - Nb de jours pour prévenir de l'expiration
  - Inactivité avant expiration **du compte**
  - Date expiration du compte
  - Réservé

LE FICHIER SHADOW

- Les commandes associées sont
  - **chage -l user** permet de lire les informations
  - **chage ...** pour les modifier
  - **pwconv** et **pwunconv** permettent de passer d'un modèle à l'autre

LES COMMANDES

- **useradd** permet d'ajouter un utilisateur
- **userdel** supprime un utilisateur
- **usermod** modifie les propriétés d'un compte
- Plus ou moins complet en fonction du système
- Utilisation de **/etc /skel** pour initialiser le dossier

PARTICULARITÉ FREEBSD

- A adopté une nouvelle approche intégrée
- Tout est regroupé dans

  /etc /master .passwd
- Par souci de compatibilité, les fichiers

  /etc /passwd et **/etc /shadow** sont crées automatiquement

Les processus

Les processus

Contenu

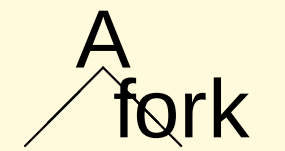
- Création
- Etats
- Les signaux
- Les priorités
- Le swap
- Les démons
- /proc

INTRODUCTION

- Un processus est caractérisé par
  - Du code
  - Des données
  - Des attributs
    - **PID**, **PPID**
    - **UID**, **GID**, **EUID**, **EGID**
    - Etat
    - Priorité

Les processus

- Création d'un processus



- A' reçoit un nouveau PID
- Son PPID est le PID de A
- L'accounting est réinitialisé

Les processus

- Les états d'un processus
  - CPU
  - Exécutable
  - Endormi
  - Zombie
  - Stoppé
  - Swappé

Les signaux

- Les signaux s'échangent entre processus
- 3 comportement possibles
  - ignoré (si possible)
  - routine de gestion prévue
  - action par défaut
- Pour envoyer un signal : **kill**
  - **kill -HUP 1**
  - **kill -9 123**
- Sur certains systèmes : **killall**

LES SIGNAUX

Num.	Nom	Description	Défaut	Ignorer ?	Gérer ?
1	HUP	Eveil	Terminer	Oui	Oui
2	INT	Interrompre (CTRL-C)	“	“	“
3	QUIT	Quitter	“	“	“
9	KILL	Tuer	“	Non	Non
15	TERM	Tuer	“	Oui	Oui
	STOP	Stop	Stop	Non	Non
	TSTP	Stop Clavier (CTRL-S)	“	Oui	Oui

Les priorités

- La **priorité** d'un processus dépend de
  - **classe** : système, real-time, time-sharing, interactif, . . .
  - **temps CPU** déjà utilisé
  - **nice** : paramètre modifiable par l'utilisateur
    - Plus c'est petit, plus la priorité est grande.
    - En **BSD** elle varie de -19 à 19.
    - En **AT&T** elle va de 0 à 39.
- Un utilisateur ne peut qu'augmenter le nice

Les processus

- Manipulation du paramètre **nice**
  - Pour donner une priorité quand on lance un processus, **nice valeur commande**
  - Sur certains systèmes, **renice**

Les processus

- Le **swap** (ou **mémoire virtuelle**)
  - une mémoire auxilliaire sur le disque dur.
  - Pour vérifier : `swap` ou `swapinfo` ou `free` ou `pstat -s`
- Les démons
  - processus du système tournant en tâche de fond et remplissant un rôle spécifique
  - ex : `sendmail` ou `in.rlogind`

Les processus

- La commande **ps** sous **BSD**

```
root@cso8   :ps   -aux   |  head
USER          PID  %CPU  %MEM     SZ   RSS  TT      S     START    TIME  COMMAND
tcouss       3319  93.5   5.0   6752  6288  pts /9  R  11:15:54    5:01  mapleV
mcodutti    28320   3.0   7.410928    9376  ?       S  17:17:10    2:04  xemacs
```

% CPU	Moyenne utilisation du CPU
% MEM	Pourcentage de mémoire RAM occupée
SIZE	Taille totale du processus
RSS	Taille RAM effectivement occupée (le reste est en swap)
TIME	temps total CPU attribué (en min :sec)

Les processus

- La commande **ps** sous **AT&T**

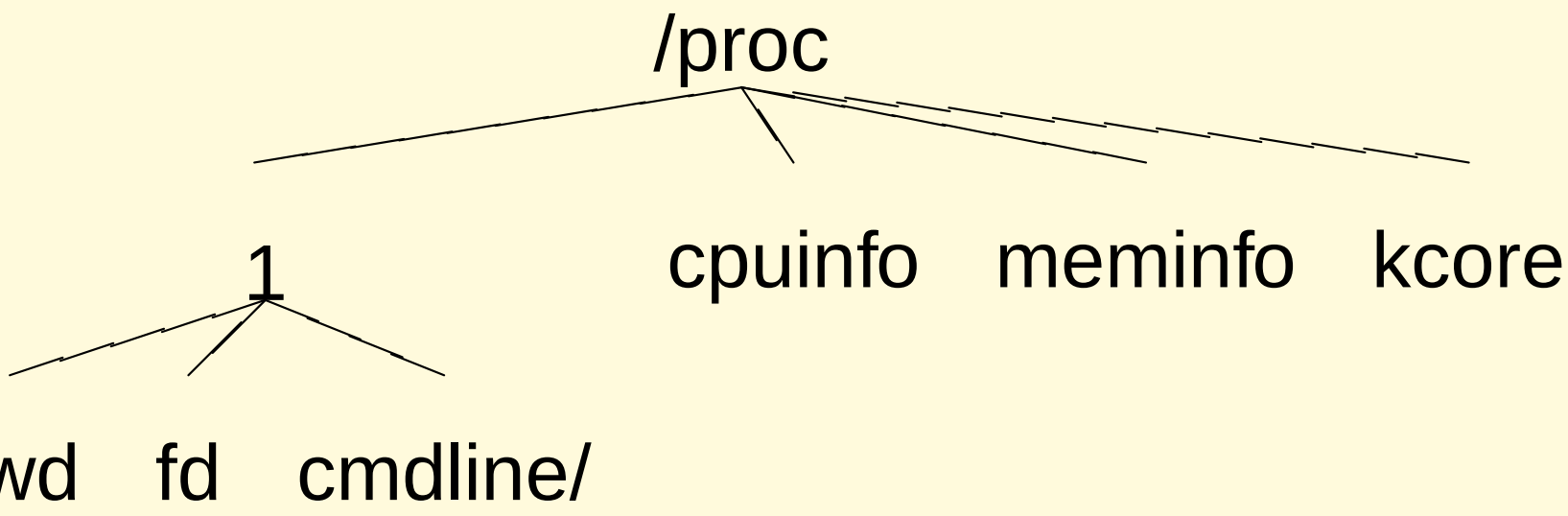
```
root@cso8   :/usr/bin/ps  -ef  | more
 UID  PID   PPID    C   STIME   TTY  TIME   CMD
root     0      0   0 Dec  22  ?    0:00   sched
root     1      0   0 Dec  22  ?    4:22   /etc/init   -
```

PPID	PID du parent
STIME	date de début du processus
C	lié à la priorité
TTY	terminal attaché au processus

Les processus

- La commande **top**
  - De plus en plus disponible
  - Présentation plein écran
  - Rafraichissement dynamique
  - Informations sur la mémoire
  - Informations sur la charge CPU
  - Accepte des commandes

Le dossier /proc



- *pseudo-directory*
- Sert de zone de dialogue avec le noyau
- Usage plus étendu que la consultation des processus

LE SHELL

Contenu

- Les scripts
- Les fichiers d'initialisation
- Les variables
- Les alias
- Les quotes
- Les redirections
- Expressions régulières

Contenu

- Historique
- Evaluation d'une commande
- Les paramètres
- Les expressions booléennes
- Les structures de contrôles
- Quelques commandes utiles

Le Shell

- Sert d'interface avec l'utilisateur
- Accepte des commandes et les lance
- Fonctionne aussi en mode **script**

sh Bourne Shell (début des années '70)

csh C-Shell (fin des années '70)

ksh Korn-Shell. **sh** + facilités du **csh**

zsh The Z-Shell. Une extension de ksh

bash Bourne Again SHell. Soutenu par GNU.

LES SCRIPTS

- **script** : suite de commandes (dans un fichier) que le shell va exécuter
- Pour lancer un script, il y a deux possibilités
  - `source script` (en **(t)csh** et **bash**)
  - `. script` (en **(k)sh** et **bash**)
  - Le rendre **exécutable** (permission **x**) et taper simplement son nom
  - La première ligne indique le programme associé (ex : `#!/ bin /sh`)

L'INITIALISATION

- Lit un script d'initialisation dans la **HOME**
- **sh**
  - `.profile` (au login)
- **(t)csh**
  - `.login` (au login)
  - `.(t)cshrc`
  - `.logout` (au logout)

L'INITIALISATION

- **bash**
  - **.bash_profile** ou **.bash_login** ou **.profile** (au login)
  - **.bashrc** (si ce n'est pas un shell associé à un login)
  - **.bash_logout** (au logout)
- Permet ainsi de configurer l'environnement
- Sur certains OS, lecture d'un fichier global avant

LES VARIABLES

- Faire la distinction entre
  - Les variables locales
  - Les variables globales (ou d'environnement)
- **(t)csh**
  - Modifier variable locale : `set VAR=valeur`
  - Lister variables locales : `set`
  - Modifier variable globale :

    `setenv VAR valeur`
  - Lister variables globales : `setenv`

LES VARIABLES

- **(ba)sh**
  - Variables : `VAR=valeur`
  - Rendre une variable globale : `EXPORT VAR`
  - Lister les variables locales : `env`
  - Lister les variables globales : `env`

Les variables standards

- **PATH**
  - Utilisé quand le chemin de la commande est relatif
  - Recommandé de ne pas avoir **.**
  - **(t)csh**, utiliser **rehash** après modification
- **MANPATH**
  - Chemins pour les pages de manuel

Les variables standards

- **LD_LIBRARY_PATH**
  - Chemins pour les librairies dynamiques
  - Au démarrage, `ldconfig` est lancé
    - Configuré par `/etc/ld.so.conf`
    - Maintient `/etc/ld.so.cache`

LES VARIABLES STANDARDS

LD_LIBRARY_PATH

- La commande **ldd** liste les librairies dynamiques d'un exécutable

```
mcodutti@lit1    :ldd  /bin /more
libintl  .so .1  =>  /usr /lib /libintl  .so.1
libc .so.1  =>       /usr /lib /libc  .so .1
libw .so.1  =>       /usr /lib /libw  .so .1
libdl .so .1  =>     /usr /lib /libdl  .so .1
```

Les variables standards

- **PROMPT**
  - Texte d'invite pour le shell interactif
  - Peut être modifié par la commande **prompt**
  - Tout une série de codes spéciaux
- **TERM**
- **USER**
- **HOME**
- **SHELL**
- **PWD**

Les variables standards

- HOSTNAME, HOSTTYPE
- EDITOR
- TAPE, PRINTER
- UID, GID
- MAIL
- LANGUAGE
- DISPLAY

Les alias

- Permet de définir un raccourci
- Pour le définir
  - En **(t)csh** : `alias nom valeur`
  - En **(ba)sh** : `alias nom=valeur`
- Pour voir la liste : `alias`
- Pour enlever un alias : `unalias`
- Ex : `alias ll='ls -l'`

Les quotes

- Modifient l'interprétation des arguments
- Au nombre de 4
  - ' : groupe les caractères et pas d'interprétation
  - \ : pas d'interprétation du seul caractère suivant
  - " : groupe les caractères
  - ` : commande remplacée par son output
- Ex : `alias BROL=ls ; echo "`BROL -l`"`

LES REDIRECTIONS

- Chaque fichier ouvert par un processus se voit assigner un numéro (le *descripteur de fichier*)
- Un processus commence avec 3 descripteurs :

0 standard input (par défaut, relié au clavier)

1 standard output (par défaut, relié à l'écran)

2 standard error (par défaut, relié à l'écran)

LES REDIRECTIONS

- On peut toutefois rediriger ces descripteurs
- En **(ba)sh**
  - **<file** : redirige l'input vers **file**
  - **>file** : redirige l'output vers **file**
  - **>>file** : redirige l'output vers **file** (append)
  - **2>&1** : associe le descripteur 2 au 1
  - **cmd1 |cmd2** : l'output de **cmd1** est redirigé vers l'input de **cmd2**

Les expressions régulières

- On dispose d'expressions régulières pour indiquer des fichiers
  - ***** : remplace 0 à n caractères
  - **?** : remplace 1 caractère
  - **[abc]** : remplace **a** ou **b** ou **c**

```
# ls -a
. .. .cshrc .login res1 res1b res2 res2b res3
# ls *
res1 res1b res2 res2b res3
# ls res ?b
res1b res2b
# ls res [12]
res1 res2
```

HISTORIQUE

- Le shell retient les commandes précédentes
  - **history** : liste des précédentes commandes
  - **!57** : commande numéro 57
  - **!rm** : la plus récente débutant par **rm**
  - **!!** : dernière commande
  - **^s1^ s2^** : dernière commande où **s1** est remplacé par **s2**
- On peut aussi utiliser les flèches

Ordre d'évaluation

- Algorithme d'interprétation du C-Shell :
  - 1. Traiter l'history (**!, ^**)
  - 2. Mettre la commande dans l'history
  - 3. Séparer en mots
  - 4. Gérer les alias
  - 5. Traiter les redirections et background (**<,>,|,&**)

Ordre d'évaluation

- Algorithme d'interprétation du C-Shell (suite) :
  - 6. Remplacer les variables par leur valeur (pas dans '')
  - 7. Traiter le back quoting (``)
  - 8. Expansion des noms de fichiers
  - 9. Exécuter la commande

LES PARAMÈTRES

- On peut passer des arguments à un script
- Dans celui-ci, on y fait référence par
  - **$1** , . . ., **$9** pour les arguments 1 à 9
  - **$*** pour tous les arguments
  - **$0** pour le nom du programme
- Exemple

```
# cat  brol
echo  "$0 -$1 -$*"
# brol  "a b" c d e
brol -a  b-a  b c d e
```

Expressions booléennes

- **test expr**
  - évalue une expression
  - retourne un statut adéquat (0 ou 1)
- Ex :

```
test -e t1 -a -e t2 -a t1 -nt t2
```

- Les shells ont des versions intégrées équivalentes (**[expr]**)

Expressions booléennes

- On peut
  - Tester l'existence d'un fichier
  - Son type, ses permissions, son propriétaire
  - Qu'il n'est pas vide
  - Sa date par rapport à d'autre
  - Effectuer des tests sur les chaînes
  - Ou sur des nombres
  - Combiner le tout avec des opérateurs logiques

Structures de contrôle

- On trouve tout ce qu'on peut attendre d'un langage impératif

```
# pour une alternative
if command ;
then
...
else
...
fi

# pour boucler sur une liste
for name in list ; do
...
done
```

Structures de contrôle

- (suite)

```
# pour répeter une commande
while command ; do
...
done
```

- Exemple

```
for name in *; do
    if [ -x $name ];
    then
        echo $name est executable ;
    fi
done
```

QUELQUES COMMANDES UTILES

head /tail

- Ne garder que les premières (dernières) lignes/octets d'un fichier
- Exemple : `#head -20 brol`
- Exemple : `#tail -20 brol`
- Exemple : `#tail +20 brol`

QUELQUES COMMANDES UTILES

grep

- Ne sélectionner que les lignes respectant certains motifs.
- On peut sortir les lignes candidates
- Les faire précéder d'un numéro de ligne
- Les accompagner de quelques lignes autour
- Inverser la sélection

QUELQUES COMMANDES UTILES

grep

- Travailler sur plusieurs fichiers
- Afficher le nom du fichier correspondant
- N'afficher que le nom des fichiers contenant au moins une ligne reprise
- Exemple : `#grep scsi *.h`

QUELQUES COMMANDES UTILES

sort

- On peut spécifier les colonnes à traiter
- Ainsi que le délimiteur de colonnes
- On peut trier en ordre inverse
- Indiquer si le tri est numérique ou alphabétique
- On peut juste vérifier si un fichier est trié
- Ou encore joindre plusieurs fichiers triés
- Ex : `#sort -t: -n +3 /etc/passwd`

QUELQUES COMMANDES UTILES

awk

- Commande très puissante mais un peu difficile à appréhender
- Considère le texte comme étant composé de colonnes (champs)
- On peut associer une série de commandes à différents motifs rencontrés
- Exemple :

```
#cat /etc/passwd | awk -F : '{print $7}'
#df | awk '!/Sys/ {print $5}'
```

QUELQUES COMMANDES UTILES

uniq

- Traite un fichier trié
- Ne prend qu'un exemplaire de lignes identiques
- Peut aussi ajouter à chaque ligne son nombre d'occurence
- Exemple :

```
#cat /etc/passwd | awk -F : '{print $7}'
> | sort | uniq
```

QUELQUES COMMANDES UTILES

sed

- Editeur non interactif
- Permet de programmer des modifications fines dans un texte
- Possède la même puissance que **vi**
- Modifications sur tout un texte ou juste une partie
- Exemple :

```
#cat  /etc /passwd   |  sed  s/ bash /tcsh /
#cat  /etc /passwd   |  sed  '2, $  s/bash /tcsh /'
```

QUELQUES COMMANDES UTILES

- Exercices récapitulatifs
  - Donner une liste triée de tous les shells utilisés sur la machine (du plus utilisé au moins utilisé).
  - Donner la liste des partitions qui sont remplies à plus de 90%. La liste sera donnée du plus rempli au moins rempli.

Le démarrage

DÉMARRAGE ET ARRÊT

Contenu

- Les grandes étapes
- Le moniteur
- Le noyau
- Les processus spontanés
- Les scripts de démarrage
- Arrêter/redémarrer

Les étapes du démarrage

- Le **moniteur** (en PROM) prend la main
- Etapes de **bootstrapping**
- Chargement et l'initialisation du **noyau**
- Détection/configuration des **périphériques**
- Création des **processus spontanés**
- Intervention de l'opérateur (si mode manuel)
- Exécution des **scripts de démarrage**
- Passage en **mode multi-utilisateur**

LE MONITEUR

- Petit programme en PROM dans la machine
- Sur PC, il s'agit du **BIOS**
- Identifie le périphérique de démarrage
- Initie le bootstrapping
- Possiblilité de l'interrompre
- Moniteur PC et **Linux**

LE MONITEUR SOLARIS

- La séquence de touches permettant d'y accéder est **`STOP -A`**.
- Quelques commandes
  - **`boot disk |cdrom |net`** pour booter sur le disque, le cdrom ou le réseau
  - **`probe -scsi`** liste des périphériques SCSI
  - **`boot -s`** pour booter en mode *single-user*
  - **`boot -r`** à utiliser si on a ajouté un périphérique depuis le dernier démarrage

LE NOYAU

- Programme qui va mettre en place **Unix**
- Va diriger les étapes suivantes du démarrage
- Restera en mémoire pendant le fonctionnement
- Possède un nom et un emplacement connu du loader

  /unix , /vmunix , /boot /vmlinuz
- Peut être recompilé pour être adapté aux besoins

DÉTECTION DES PÉRIPHÉRIQUES

- Noyau construit avec certains pilotes
- Au démarrage,détectent la présence effective du matériel géré
- Certains noyaux peuvent s'enrichir dynamiquement de nouveaux pilotes (modules) Cela permet :
  - de garder un noyau léger
  - de ne pas devoir recompiler à chaque ajout de pilote

Les processus spontanés

- Appelés ainsi parce que non créés par le système **fork** traditionnel
- Leurs nombres, rôles et noms varient d'un **Unix** à l'autre
- Il y a toujours **init** avec le PID 1
- C'est ce dernier qui va mettre la système en opération (démarrage des démons, des processus d'ouverture de sessions, ...)

LE MODE MANUEL

- `init` peut passer en mode manuel.
- Utile en phase de récupération/réparation (disque endommagé par exemple)
- Système minimal (au niveau des partitions montées, des processus lancés)
- Demande le plus souvent le mot de passe root
- Une fois terminé, on peut demander à continuer le démarrage

LES SCRIPTS DE DÉMARRAGE

- Ils sont lancés par **init**
  - Donner un nom à l'ordinateur
  - Définir la zone horaire
  - Vérifier les disques et les assembler
  - Nettoyer **/tmp**
  - Configurer le réseau
  - Lancer les démons
- Le mécanisme mis en oeuvre est très différent entre **BSD** et **SysV** .

LES SCRIPTS DE DÉMARRAGE BSD

- **`init`** exécute le seul script **`/etc /rc`** (shell)
- Ce script s'occupe de lancer les autres
- Il se base sur des fichiers de configuration (**`/etc /rc .conf`** , **`/etc /rc .conf .local`**)
- Et lance des scripts spécifiques (**`/etc /rc *`**)
- Le script **`/etc /rc .local`** est dédié aux modifications locales
- Il peut exister un script (genre **`/etc /rc .shutdown`**) exécuté lors d'un arrêt

LES SCRIPTS DE DÉMARRAGE SYSV

- Il existe 7 niveaux d'exécution

  0 : système arrêté

  1 (ou S) : niveau mono-utilisateur

  2 à 5 : différents niveaux utilisateurs

  6 : système redémarré

- Démarrer ou arrêter revient à changer de niveau
- A chaque niveau sont associés des scripts

LES SCRIPTS DE DÉMARRAGE SYSV

- **/etc /initttab** définit les scripts à lancer lorsqu'on entre dans un niveau
- Extrait de la version **Solaris**

```
s3:3:wait:/sbin/rc3 >/dev/console 2>&1 </dev/console
```

  - **s3** : nom arbitraire
  - **3** : concerne l'entrée dans le niveau 3
  - **rc3** : script à exécuter
  - **wait** : attendre que le script soit fini avant de passer à autre chose

Les scripts de démarrage SysV

- Chaque script **rci** a un comportement standard
- A chaque niveau correspond le dossier **/etc /rci .d**
- Contient des scripts de la forme **[K|S]nNOM**
  - **S** pour **Start** ; **K** pour **Kill**
  - le numéro permet de les ordonner
  - le nom indique son rôle (démarrage d'un démon ; aspect particulier du système, ...)

LES SCRIPTS DE DÉMARRAGE SYSV

- A l'entrée d'un niveau,
  - Exécute tous les scripts commençant par **K**
    - dans l'ordre des numéros
    - avec l'argument **stop**
  - Exécute tous les scripts commençant par **S**
    - Ordre des numéros ; Argument **start**
  - Presque toujours, il s'agit de liens symboliques vers les scripts déposés dans **/etc /init .d**

LES SCRIPTS DE DÉMARRAGE SYSV

Exemple

- On a un gestionnaire de licence à démarrer au niveau 3 et à tuer au niveau 0
- On crée le script **/etc /init .d/ licence** qui ressemble à

```
#!/ bin /sh
case "$1 " in
    'start ')
        # on lance ici le démon
    'stop ')
        # on tue ici le démon
esac
```

Les scripts de démarrage SysV

Exemple (suite)

- Il ne reste plus qu'à créer les liens symboliques qui vont définir son utilisation

```
ln -s /etc/init.d/licence /etc/rc3.d/S20licence
ln -s /etc/init.d/licence /etc/rc0.d/K20licence
```

LES SCRIPTS DE DÉMARRAGE SYSV

Particularités **Linux** . Les niveaux d'exécution 2 à 5 sont tous multi-utilisateurs et définis comme suit :

2 : sans réseau

3 : avec réseau

4 : inutilisé

5 : avec réseau et interface graphique

Arrêter, redémarrer le système

Il existe plusieurs façons d'éteindre un ordinateur.

- La commande **shutdown**
- Les commande **halt** et **reboot**
- Envoyer un signal à **init**
- Tuer **init**
- Couper l'alimentation

Décrivons cela en commençant par le plus sûr

SHUTDOWN

- Commande la plus sûre et la plus propre
- Elle permet de
  - programmer un arrêt
  - prévenir les utilisateurs (message explicatif)
  - empêcher les logins vers la fin
- Options pour indiquer le niveau
  - arrêter la machine
  - passer en mode single user
  - rebooter la machine

SHUTDOWN

- La syntaxe exacte est différente d'un système à l'autre
- Exemple : En **Linux** ,

```
shutdown   -h 23:00  'Upgrade   Systeme '
```

  - indique qu'on programme un arrêt à 23 heures
  - Les utilisateurs seront prévenus par le message indiqué

HALT ET REBOOT

- Tout aussi sûr que la commande précédente mais peut-être plus brutal
- Sur certains OS, personne n'est prévenu et l'action est immédiate
- Sur d'autres, c'est équivalent à **shutdown**

FASTHALT ET FASTBOOT

- Issu de **BSD** .
- N'indique pas que l'arrêt doit être rapide mais bien le prochain démarrage.
- Réalisé grâce au fichier vide **/fastboot**

AUTRES POSSIBILITÉS

A déconseiller fortement.

- Sur certains OS, envoyer le signal **TERM** au processus **init** permet de passer en mono-utilisateur
- Tuer le processus **init** permet de redémarrer
- Séquence de touches
- Bouton On/Off
- Sur **Linux** : **ALT -CTRL -DEL** est OK. Pq ?

Les tâches périodiques

Les tâches périodiques

Contenu

- Le démon
- Les fichiers **crontab**
- Les fichiers de permission

Les tâches périodiques

- Le démon **cron** tourne en permanence
- Exécute des commandes à intervalle régulier
- Se base sur des fichiers de configuration
- Un par utilisateur (portant son nom)
- Se trouve dans un dossier bien précis
  - **/var /spool /cron /crontabs**
  - **/var /spool /cron**
  - **/var /cron /tabs**

Les tâches périodiques

- Il a le format suivant

```
# commentaire
minute   heure   jour   mois   jour_semaine      commande
```

- ex : **0,30 8-20 * * * verif** commande **verif** lancée toutes les heures précises et les heures 30, tous les jours entre 8h et 20h30.
- Chaque commande est lancée dans un shell
- Et envoie l'output en mail à l'utilisateur
- Peut enregistrer un log de chaque exécution

Les tâches périodiques

- Les fichiers sont gérés par **crontab**

crontab -l	liste le contenu du fichier
crontab -e	éditer (en **vi**) le fichier
crontab -e user	éditer le fichier de **user** (uniquement par **root**)
crontab -r	pour supprimer le fichier

Les tâches périodiques

- Les fichiers **allow** et **deny** indiquent qui peut utiliser crontab.
- S'ils n'existent pas, alors tout le monde ou seulement **root**
- Si **allow** existe alors les utilisateurs qui s'y trouvent
- Si **deny** existe alors les utilisateurs qui ne s'y trouvent pas

Les tâches périodiques

- Ces fichiers ont des noms et des emplacements variables
- On retrouve **cron.allow** et **allow**
- On les trouvera dans **/etc**, **/var/cron**, **etc/cron.d**, ...

LA SAUVEGARDE

La sauvegarde

Contenu

- Introduction
- Les commandes
- Les supports
- Les stratégies
- Quelques conseils

INTRODUCTION

- L'information est souvent plus importante que l'ordinateur lui-même
- Il est impératif de s'assurer contre la perte d'informations due à
  - une défaillance matérielle
  - une destruction par un logiciel
  - une erreur de l'utilisateur (`rm *`)
  - un désastre (incendie, tremblement de terre, raz de marée, . . .)

INTRODUCTION

- La stratégie adoptée va dépendre de
  - la quantité, le roulement et l'importance de l'information
  - la somme que l'on est prêt à engager

LES COMMANDES

- Les commandes les plus répandues pour la sauvegarde dans le monde **Unix** sont **dump** et **restore**
- De nombreux logiciels offrent une interface agréable et des possibilités d'automatisation mais utilisent ces 2 commandes en arrière plan

LA COMMANDE DUMP

- La commande **dump**
  - on peut tout sauvegarder
  - respecte les liens
  - on sauve une partition (optimisé)
  - peut tenir sur plusieurs bandes

LA COMMANDE DUMP

- notion de **sauvegarde incrémentale**
  - se fait à un certain **niveau** (0 à 9)
  - au niveau *i* on sauve tout ce qui a été modifié depuis la dernière sauvegarde à un niveau inférieur
  - au niveau 0, on sauve tout

LA COMMANDE DUMP

- La commande ressemble à

```
dump  0uvf  /dev/nrst0  /dev/rsd0g
```

  - **0** niveau (ici full backup)
  - **u** mettre à jour **/etc/dumpdates** qui retient les dates et niveaux des backups
  - **v** verbeux
  - **f** on spécifie le nom du tape
- Sous **Solaris** , la commande est **ufsdump**

LA COMMANDE DUMP

- Il est possible d'effectuer une sauvegarde sur une autre machine

```
(r)dump  0uf  orca :/dev/nrst0  /dev/rsd0g
```

- **Attention :** Il faut les permissions réseau (**.rhosts**)
- Permet de centraliser les sauvegardes

LA COMMANDE DUMP

- ex : soit 3 machines (A,B,C) avec un tape branché sur B.
- On peut lancer sur A un script qui ressemble à ce qui suit pour sauvegarder les 3 machines.

```
# script sur A pour lancer le backup
      rdump B:tape partitions
rsh B dump tape partitions
rsh C rdump B:tape partitions
```

LA COMMANDE RESTORE

- On peut restaurer
  - tout

    restore rvf /dev /nrst0
  - une partie

    restore xvf /dev /nrst0 fichiers
  - réseau

    (r)restore xvf orca :/dev/ nrst0 fichiers
  - interactif

    restore ivf /dev /nrst0

LA COMMANDE RESTORE

- déterminer sur quelle(s) bande(s) se trouvent les fichiers à restaurer.
- restaurer dans l'ordre chronologique
- choisir le dossier où on restaure. En effet, la restauration se fait dans le dossier courant, d'où
  - dans **/tmp** puis les déplacer
  - directement à la bonne place
- Sous **Solaris** : **ufsrestore**

LA COMMANDE RESTORE

- Restauration interactive
  - **ls** idem **ls** UNIX (contenu)
  - **cd dir** idem **cd** UNIX (change directory)
  - **add file |dir** ajoute le fichier ou le dossier (récursif) dans la liste des fichiers à restaurer
  - **extract** lance la restauration.

LA COMMANDE TAR

- Collecte plusieurs fichiers en un seul
- Facilite leur manipulation
- Plus simple et plus souple d'usage que **dump**

MAIS

  - Ne propose pas de sauvegarde incrémentale
  - Pas de sauvegarde sur plusieurs volumes
  - Moins rapide

LA COMMANDE TAR

- **créer**

  `cd / ; tar cf /tmp/home.tar home`

- **contenu** : `tar tf /tmp/home.tar`

- **extraire** `bob` dans `/tmp/home/bob`

  `cd /tmp ; tar xf home.tar home/bob`

- **sur bande**

  `cd / ; tar cf /dev/nrst0 home`

- **compression**

  `cd / ; tar zcvf /tmp/home.tgz home`

LA COMMANDE MT

- Possible de mettre plusieurs enregistrements à la suite sur une bande
- Le lecteur de bandes met des repères MAIS
  - Pas de table des matières
  - Rien sur le format (tar, dump, autre, . . .)
- C'est pourquoi, il faut **tout noter soigneusement**

LA COMMANDE MT

- Exemple

```
tar cf /dev/nrst0 /home; tar cf /dev/nrst0 /usr
```

donne | /home : /usr : |

- **mt rew** pour rebobiner
- **mt fsf n** pour avancer de n enregistrements
- **mt off** pour éjecter la bande

LA COMMANDE MT

- Exemple
  - `mt rew ; mt fsf 2`
  - `tar cf /dev/nrst0 /tmp`
  - `mt rew`
  - `tar xf /dev/nrst0`
  - `mt fsf 1` (**Attention**)
  - `tar xf /dev/nrst0`

Les supports de sauvegarde

- Offre abondante. Citons

Disquettes Volume et fiabilité faibles

Superdisquettes Les lecteurs **Zip** et **Omega** sont présents mais le coût reste élevé

CD Intéressant mais capacité faible et le prix des réinscriptibles assez élevé

Bandes **QIC**, 8mm **Exabyte** ou 4mm **DAT**. Capacité allant jusqu'à 40 Gb avec un prix intéressant

Les stratégies de sauvegarde

- La stratégie de backup à adopter dépend
  - du nombre de bandes disponibles
  - de l'époque à laquelle on veut pouvoir remonter et avec quelle précision
  - de la facilité de récupération
  - du temps de backup
- Partitions fort différentes $\Rightarrow$ stratégies différentes

Les stratégies de sauvegarde

- Fichiers **personnels** : sensible.
  - Une sauvegarde **quotidienne** de niveau **9**. Une bande par jour. Eventuellement sur disque pour faciliter les opérations.
  - Une sauvegarde **hebdomadaire** de niveau **5**. Roulement avec 8 bandes.
  - Une sauvegarde **mensuelle** de niveau **0**. Une bande par mois.
  - Toutes les bandes de janvier sont gardées.

Les stratégies de sauvegarde

- Lors de la restauration, un maximum de 3 bandes seront nécessaires
- Sauvegarde quotidienne sur disque
  - Facilite la sauvegarde et la restauration
  - Rend l'information plus vulnérable
- Attention : **dump** sauvegarde une partition. Certains fichiers peuvent se trouver ailleurs
- Los d'une restauration des fichiers détruits peuvent réapparaître

Les stratégies de sauvegarde

- Autre possibilité
  - On utilise 9 bandes et le niveau de sauvegarde croit linéairement : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Avantages ? Inconvénients ?

LES STRATÉGIES DE SAUVEGARDE

- Fichiers **systèmes** : sensible mais peu de modifications.
  - La commande **tar** exécutée à intervalle régulier (hebdomadaire par exemple) peut s'avérer suffisant.
  - Avec la commande **dump** on utilisera toujours le niveau **0**

Quelques conseils précieux

- **Centraliser** et **automatiser** les sauvegardes : pour une plus grande simplicité.
- **Etiquetter** les bandes et se donner un accès rapide à la **table des matières**
- Choisir un intervalle de sauvegarde raisonnable
- Choisir les systèmes de fichier avec précaution
- Essayer de faire tenir les sauvegardes sur une seule bande

Quelques conseils précieux

- Entreposer les bandes à l'extérieur
- **Protéger** les sauvegardes : l'information sur bande est vulnérable ; il s'agit d'un trou énorme dans la sécurité des données
- Limiter autant que possible les activités pendant les sauvegardes
- **Vérifier les bandes ! Vérifier les bandes ! Vérifier les bandes ! Vérifier les bandes !**
- Préparer les scénarios catastrophes

Les fichiers journaux

LES FICHIERS JOURNAUX

- Les fichiers journaux classiques
- Le système **syslog**

LES FICHIERS JOURNAUX

- Un journal (**log** en anglais) est un fichier qui garde une trace de l'activité d'un composant
- Il sert à
  - prévenir une intrusion ou trouver le reponsable
  - détecter la source et les causes d'une panne
  - analyser les performances du système
  - calculer des statistiques sur l'utilisation d'un service

LES FICHIERS JOURNAUX

- Localisations et noms incohérents
- Exemples : **/var /adm /**, **/var /log /**, **/var /cron /**, **/usr /adm** , . . .
- Le fichier *log* effectivement utilisé par un programme peut être
  - hardcodé
  - une option de démarrage
  - déterminé par un fichier de configuration
  - défini via *syslog*

Les fichiers journaux

- Ces journaux ont tendance à croître rapidement
- Quelle stratégie adopter ?

1. On peut décider de ne rien garder
2. On peut les remettre à zéro régulièrement
3. On peut faire une rotation et garder quelques fichiers
4. On peut les archiver

LES FICHIERS JOURNAUX

Remarques

- Certains journaux sont gardés ouverts en permanence

  → dangereux de les manipuler directement
- Des scripts de domaine public existent pour la rotation des fichiers
- Souvent même, ils sont déjà intégrés au système et lancés automatiquement via **cron**

LES FICHIERS JOURNAUX

- Quelques fichiers journaux courants.

messages	Central. Messages très variés
syslog	idem
sulog	renseigne sur les **su** qui ont été effectués
sudo .log	idem pour les **sudo**
lpd -errs	système d'impression LPD
xdm -erros	erreur du système XDM

LES SYSTÈME INTÉGRÉ SYSLOG

- Mécanisme centralisé de gestion des messages de trace.
- Simplifie et standardise
  - le production des traces
  - leur gestion
- Le système est composé de trois parties
  - le démon **syslogd**
  - les appels systèmes
  - une commande interactive

LES SYSTÈME INTÉGRÉ SYSLOG

- Démon piloté par un fichier de configuration
- En fonction du message
  - sa provenance (type) : kern, mail, lpr, . . .
  - son importance (niveau) : error, warning, . . .
- un message peut être
  - jeté
  - envoyé sur les écrans
  - sauvé dans un fichier
  - passé à une autre machine (centralisation)

LES SYSTÈME INTÉGRÉ SYSLOG

- Fichier de configuration **/etc /syslog .conf**
- Exemple

```
*.emerg                          *
*.crit                           /dev /console
*.warning ; auth .info           /var /adm /messages
daemon ,auth .info               /var /adm /messages
local0 .*                        @loghost
```

- Le niveau donné est celui **à partir duquel** la règle s'applique
- Sur **Linux** syntaxe plus riche

LA COMMANDE LOGGER

- Permet d'envoyer un message au démon
- Utile :
  - pour tester le fichier de configuration
  - dans les scripts locaux
- Exemple

```
[marco@localhost marco ]$ logger -p kern.emerg
>"Formatting hard disks ..."
[marco@localhost marco ]$
Message from syslogd@localhost at Fri Mar 7
localhost marco : Formatting hard disks ...
```

- Attention : tout le monde peut utiliser ce mécanisme

Le Network File System

Le Network File System

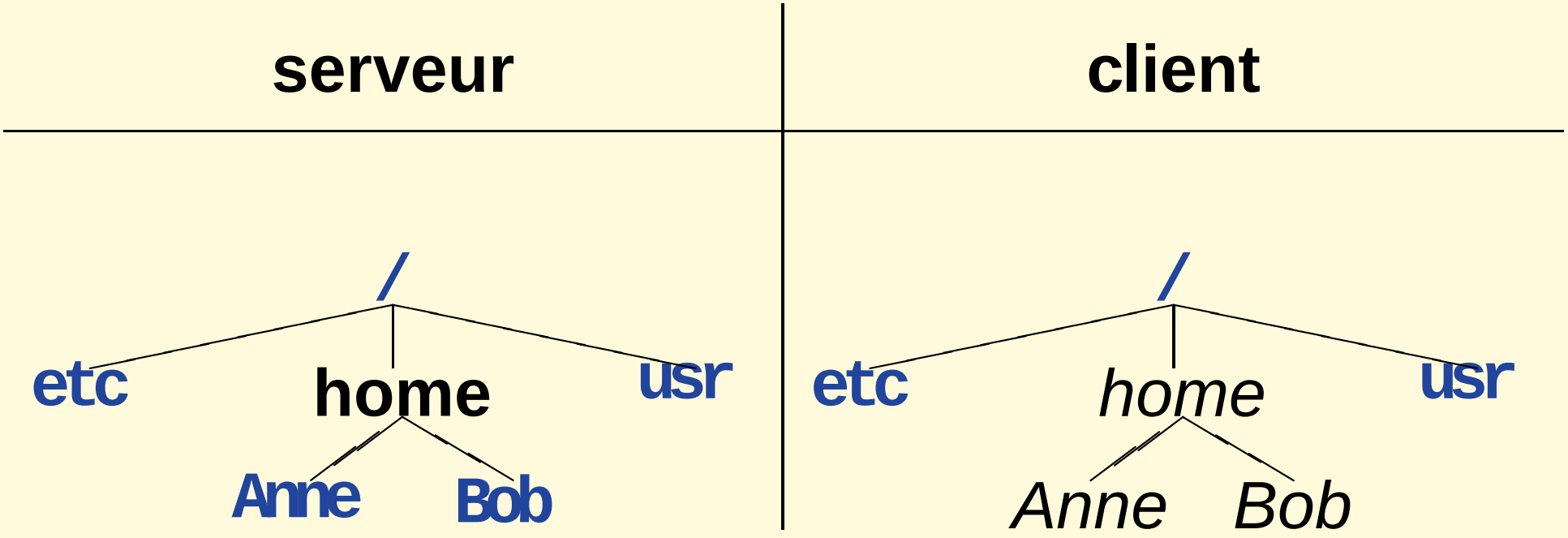
- Mécanisme
- Le serveur NFS
- Le client NFS
- Démarrage
- Statistiques
- Montage automatique

INTRODUCTION

- Permet à une machine de **partager** une partie de ses fichiers
- Accessibles à partir d'autres machines
- Introduit dans les années 80 par **SUN**
- Les avantages sont :

1. un gain de place
2. une même vision des fichiers sur toutes les machines

MÉCANISME



- Le serveur *exporte* un dossier (ici **/home**)
- Le client *monte* explicitement ce dossier
- Ensuite, tout est transparent

MÉCANISME

- Remarques :
  - **NFS** est implémenté sur le protocole **RPC** (Remote Procedure Call) qui utilisera **TCP** ou **UDP**
  - On en est à la version 3 à préférer à la version 2, plus lente
  - Il faut être attentif à la gestion des **UID** et **GID**

Du côté du serveur

- Le serveur doit **exporter** (**partager**) ses fichiers
- Deux protocoles sont alors mis en place
- Le premier pour le montage des dossiers
- Un autre pour les accès aux fichiers

Du côté du serveur

- Le démon s'occupant du partage est **mountd** (parfois **rpc .mountd**)
- Se base sur une fichier de configuration
- Le nom du fichier (ainsi que son format) peut varier d'un système à l'autre
- Une commande permet d'ajouter/supprimer dynamiquement des dossiers à exporter

Du côté du serveur

- Sur de nombreux systèmes, le fichier de configuration est **/etc /exports**
- Exemple

```
/usr /csoft          -access  =cso5 :cso6 :cso7 :cso8
/usr /share /man     -ro
```

- On indique qu'on exporte **/usr /csoft** mais uniquement pour ces 4 machines
- Par contre, **/usr /share /man** est exporté au monde entier mais en lecture seule

Du côté du serveur

- La commande associée est **exportfs**
- Permet de visualiser/ajouter/supprimer
- Après avoir modifié le fichier, il faut demander au démon de se resynchroniser (**exportfs -av**)

Du côté du serveur

- Sous **Linux** le format du fichier **/etc /exports** est différent
- On trouvera plutôt des lignes du genre

```
/usr /csoft   cso5 (rw ) cso8 (ro )
```

- Indique qu'on donne l'accès en écriture à **cso5** mais uniquement en lecture pour **cso8**

Du côté du serveur

- **Solaris** a décidé de ne pas faire comme tout le monde
- Utilise le fichier **/etc /dfs /dfstab**
- Exemple

```
share  -F  nfs  -o  rw=cso8  /usr /csoft
```

- Pour visualiser ce qui est exporté : **share**
- Pour demander au démon de relire le fichier : **shareall**

Du côté du serveur

- Options
  - lecture/écriture ou uniquement écriture (éventuellement en fonction de la machine)
  - Pas de création de SETUID bit à 1 via NFS

Du côté du serveur

- Options (suite)
  - Nécessite une cohérence des UID et GID (ou démon **ugidd** pour le mapping)
  - **root** reste **root** ou devient **nobody**
  - Imposer d'autres comptes qui deviennent **nobody**
  - Imposer le **UID** de **nobody**

Du côté du serveur

- Le démon qui répond aux requêtes d'accès aux fichiers via NFS est **nsfd** (parfois **rpc .nfsd**).
- Il est lancé avec un argument : le nombre de copies de lui-même
- Peut avoir son importance sur la qualité du service
- Trop petit → retard dans les réponses
- Trop grand → serveur saturé

Du côté du client

- Le client doit d'abord *monter* le dossier.
- La commande est **mount**

  Exemple

```
mount   cso8 :/usr /share  /man   /usr /share  /man
```

- Comme toujours le dossier **/usr /share /man** doit exister sur **cso7**
- En général un dossier vide

Du côté du client

- Pour visualiser les partitions NFS montées,

  mount ou **df**

- Ces commandes donnent des informations sur tous les filesystems montés, pas uniquement via NFS
- Pour démonter un fichier,

  umount /usr/share/man

DU CÔTÉ DU CLIENT

- La commande **mount** fonctionne au coup par coup
- Pour le démarrage, on utilise le fichier **/etc /fstab** (**/etc /vfstab** sous **Solaris**)

Exemple

```
mcodutti@mathpc    4:cat  /etc /fstab
/dev /hda2       swap        swap      defaults      1    1
/dev /hda3       /           ext2      defaults      1    2
/dev /hda1       /dos        msdos     defaults      1    3
none             /proc       proc      defaults      1    4
cso7 :/ ULB /staff  /mcodutti    /ULB /staff  /mcodutti    nfs  (...)
cso :/ var /mail                 /var /spool  /mail        nfs  (...)
```

Du côté du client

- Parmi les options, citons :
  - Montage bloquant ou non
  - Accès aux fichiers bloquant ou non
- Pour examiner le comportement du système, on utilise
  - **nfsstat** **-c** pour le client
  - **nfsstat** **-s** pour le serveur

MONTAGE AUTOMATIQUE

- Avec NFS, tout est monté au début
- Problématique si le serveur plante
- Un système de montage automatique vient se greffer sur NFS pour monter les dossiers uniquement en cas de besoin
- Automatiquement démontés après un certain temps d'inactivité
- **automount** : premier apparu, plus simple
- **amd** : plus complexe mais plus riche

MONTAGE AUTOMATIQUE

- **automount** se configure via plusieurs types de **cartes**
  - Les cartes indirectes : point de montage explicité ailleurs
  - Les cartes directes : point de montage explicite
  - La carte principale
  - Les cartes exécutables : générées par un programme

MONTAGE AUTOMATIQUE

- Les noms sont libres mais, par convention, on utilise **/etc /auto .xxx**
- Exemple

```
# cat /etc/auto.home
math  math :/home
info  info :/home

# cat /etc/auto.direct
/usr/share/man  brol :/usr/share/man

# cat /etc/auto.master
/home  /etc/auto.home
/-     /etc/auto.direct
```

MONTAGE AUTOMATIQUE

- Options et variantes :
  - on trouve la plupart des options NFS
  - on a la possibilité d'indiquer plusieurs serveurs
  - des raccourcis existent pour monter
    - tous les dossiers exportés d'une machine
    - le dossier de l'utilisateur
- Le démon s'appelle **automountd** (ou **automount**)

LE PARTAGE DE FICHIERS

Partage de fichiers

- rdist
- expect

RDIST

- Permet de copier des fichiers d'une machine centrale (serveur) vers des clients
- Le plus grand apport étant de ne copier que ce qui a été modifié depuis la dernière copie
- Par rapport à NFS :
  - Perte de place
  - Délai de mise à jour
  - Meilleure résistance à des pannes
  - Moins grande sécurité

RDIST

- Le serveur contient l'original d'un ensemble de fichiers
- Les clients recoivent d'autorité une copie
- Cette copie est effectuée à la demande du serveur

RDIST

- Fonctionnement via un fichier de configuration
- ex : pour maintenir un même dossier

  /usr /local sur les machines

  cso5 ,cso6 ,cso7 et **cso8** , on aura sur **cso8** :

```
# cat  /etc /distfile
(/usr /local ) -> (cso5 ,cso6 ,cso7 )
     install ;
     notify  mcodutti@cso   .ulb .ac .be
```

RDIST

- La copie se fait via

  rdist -f /etc /distfile .

- Généralement lancé via le **cron**
- Dipose de bon nombre d'options
  - Permettre ou pas de supprimer des fichiers
  - Quid des liens symboliques
  - Garder des traces ou pas
  - Préserver les permissions
  - Garder une sauvegarde

RDIST

- **rdist** pose des **problèmes de sécurité**
- Il utilise la commande **rsh**
- Le serveur doit avoir un accès **root** sur le client
- On doit le permettre via **/etc /hosts .equiv**
- Exemple

```
cso1 #cat  /etc /hosts  .equiv
cso2
```

- Résolu avec les nouvelles versions compatibles **ssh**

EXPECT

- Permet d'introduire de l'automatisation dans un processus interactif (en mode texte)
- Lance un processus interactif et réagit (en simulant un texte tapé) lorsqu'il reconnait le texte envoyé par le processus
- Peut être utilisé comme alternative à **rdist**
- En utilisant un serveur FTP

Exemple

```
spawn  /usr /bin /ftp  nom_serveur
while  1 {
   expect  {
      "Name *: "    {send  "nom_user  \r"}
      "Password  :"  {send  "mot_de_passe   \r"}
      "ftp > "       {break }
      "failed  "     {send_user   "Réfusé  !\r";  exit  1}
      timeout        {send_user   "Timeout  !\r";  exit  2}
   }
}
send  "lcd  qqpart  \r"
expect  "ftp > "  {send  "cd  qqpart  \r"}
expect  "ftp > "  {send  "get  qqchose  \r"}
expect  "ftp > "  {send  "quit \r"}
exit  0
```

Le Network Information Server

NIS

- Mécanisme
- Domaine NIS
- Serveur maître
- Serveurs esclaves
- Clients
- Cas des mots de passe
- Utiliser NIS
- Sécurité
- NIS+

PRÉSENTATION

- NIS = **Network Information Server**
- Base de **données centralisée** pour certains **fichiers systèmes**
- Introduit par SUN dans les années 80
- NIS s'appelait autrefois *Yellow Pages* (YP). Problème de copyright.

MÉCANISME

- Les fichiers sont maintenus sur une seule machine, le serveur
- Chaque client interroge ce fichier centralisé
- Facilite grandement le travail d'administration
- Plus lourd mais plus souple que la simple copie régulière de fichiers

MÉCANISME

- Quels fichiers partager ?
  - **/etc /passwd** et **/etc /shadow**
  - **/etc /hosts**
  - **/etc /group**
  - **/etc /mail /aliases**
  - **/etc /printcap**
- Ce mécanisme peut être utilisé pour n'importe quel fichier
- La limite sera au niveau de la consultation

MÉCANISME

- NIS est composé
  - d'un serveur maître
  - de 0 à k serveurs esclaves
  - de clients
- Les fichiers sont
  - tenus à jour sur le serveur maître
  - répercutés sur les serveurs esclaves
  - consultés par les clients

MÉCANISME

- Sur le serveur, les fichiers textes sont **compilés** en des fichiers indexés (accès rapide à l'information)
- Les clients ont accès aux enregistrements du fichier

DOMAINE

- Un serveur sert un *domaine*
- Rien à voir avec le domaine définit par DNS

```
mcodutti@lit1    :domainname
lit
# pour  modifier    le  domaine
mcodutti@lit1    :domainname    brol
```

- Sur certains OS, le fichier

  /etc /defaultdomain est lu au démarrage

  pour déterminer le domaine.

SERVEUR MAÎTRE

- On peut utiliser comme fichiers distribués
  - les fichiers standards (ex : **/etc /hosts**)
  - des copies déposées ailleurs

    (ex : **/var /yp/ nis /hosts**). Plus propre
- Généralement configuré lors de l'installation du service
- Les démons prenant en charge ce service sont

ypserv	serveur
ypxfrd	propage la base vers les esclaves

SERVEUR MAÎTRE

- Pour créer un serveur maître :

```
# cd /var/yp
# domainname brol
# ypinit -m
# ypserv
```

- **ypinit** : pose des questions à propos des esclaves, des fichiers à partager, . . .
- **ypserv** lance les serveurs
- Lorsqu'on modifie un fichier :

  make (ou **ypmake**)

SERVEUR ESCLAVE

- Optionnel
- Pour décharger le maître
- et/ou pour prendre le relais en cas de panne de celui-ci
- Pour les créer, il faut introduire la suite d'instructions

```
# cd /var/yp
# domainname brol
# ypinit -s nom_maître
# ypserv
```

CLIENT

- Il peut être configuré pour
  - demander à un serveur bien particulier
  - prendre le premier qui répond pour un domaine donné
- Le démon du client est **ypbind**
  - Cherche un serveur
  - L'interroge

FICHIER /ETC/PASSWD

- Le cas du fichier **/etc /passwd** est particulier
- Modifié aussi suite à l'intervention des utilisateurs
- La commande **passwd** n'a plus de sens
- Introduction de la commande **yppasswd**
- Sur le serveur : démon **rpc .yppasswd**

UTILISER NIS

- Ancienne méthode
  - un fichier système qui se termine par **+**

    indique qu'il faut aller consulter la base NIS si l'information n'est pas trouvée en local
  - Pas possible de bypasser les informations locales

UTILISER NIS

- Méthode actuelle
  - Fichier de configuration :

    /etc /nsswitch .conf
  - Plus vaste puisqu'il permet également de gérer **NIS +** et **DNS**
  - Exemple

```
passwd  :   files   nis
shadow  :   files   nis
hosts   :    nis  [NOTFOUND  =return  ]  files
```

UTILISER NIS

- Les fichiers locaux gardent tout leur sens
  - En cas de panne des serveurs
  - Lors du démarrage
  - Mot de passe du root doit pouvoir rester différent d'une machine à l'autre

INTERNET ET LE ROUTAGE

Le réseau Internet

- Introduction
- Adresse IP
- Sous-réseau
- Connexion au réseau
- Routage
- Tests

INTRODUCTION

- Différents types d'adresses rencontrés.
  - L'adresse **MAC** (6 octets)
  - L'adresse **IP** (4 octets)
  - Les **noms** d'hôtes
  - Les **ports** (2 octets)

ADRESS IP

- Les différentes classes de réseau

Cl.	byte 1	schéma	
A	1 à 126	N.H.H.H	Réseaux majeurs
B	128 à 191	N.N.H.H	Sites larges
C	192 à 223	N.N.N.H	réseaux de 250 machines

- Les autres sont expérimentales
- La partie réseau est allouée par des organismes
- La partie machine est de la responsabilité de l'administrateur du site

SUBNET

- On peut raffiner en créant un *subnet*
- ex : à l'ULB, l'adresse 164.15.125.1 est composée de
  - la partie réseau (164.15)
  - la partie subnet (125)
  - le numéro de la machine (1)
- Spécifié via **netmask**
- Ou via la notation CIDR (Classless Inter-Domain Routing) (ex : 164.15.125/24)

CONNEXION

- La mise en service d'une interface réseau se fait via la commande **ifconfig**
- Exemple

```
ifconfig    eth0   164.15.127.64      up
>     netmask    255.255.255.0
>     broadcast     164.15.127.255
```

- **ifconfig eth0** : renseigne sur la configuration courante
- **ifconfig** : liste les interfaces réseau connues

LE ROUTAGE

- Le **routage** consiste à déterminer le chemin à prendre par un paquet pour aller d'une machine à une autre
- Cela se fait par une suite de décisions locales
- Chaque machine dispose d'une table indiquant à qui envoyer le paquet en fonction de l'adresse IP
- La commande **netstat** **-r** affiche la table de routage

LE ROUTAGE

- Exemple : pour mon PC à l'ULB (164.15.127.?)

```
mcodutti@mathpc 4:netstat -r
Kernel routing table
Destination     Gateway         Genmask         Flags   M  Iface
localnet        *               255.255.255.0   U       0  eth0
loopback        *               255.0.0.0       U       0  lo
default         gate_127        0.0.0.0         UG      1  eth0
```

LE ROUTAGE

- Exemple : Quelle est la topologie du réseau autour de cette machine ?

  (autre **Linux** autre format de sortie)

```
# netstat   -r  -n
Dest                Mask                Gateway             Fl   If
132.236.227.0       255.255.255.0       132.236.227.93      U    eth0
default             0.0.0.0             132.236.227.1       UG   eth0
132.236.212.0       255.255.255.192     132.236.212.1       U    eth1
132.236.220.64      255.255.255.192     132.236.212.6       UG   eth1
127.0.0.1           255.255.255.255     127.0.0.1           U    lo0
```

LE ROUTAGE

- Comment la table est-elle construite ?

  (dans le cas de tables statiques)

  - Au démarrage via des fichiers de configuration
  - Au démarrage suite à des commandes (**`ifconfig`** , . . .)
  - Manuellement avec la commande **`route`**

LE ROUTAGE

- Pour ajouter un chemin (**Linux**) :

```
route   add   -net   164.15.123.0      eth0
```

- Pour définir la route par défaut (**Solaris**) :

```
route   add   net   default    164.15.125.254      1
```

- Sur certains OS (**Solaris**), le fichier **/etc /defaultrouter** est lu au démarrage pour déterminer le routeur par défaut
- Sur d'autres (**Linux**), ces infos sont reprises dans le fichier **/etc /sysconfig /network**

Tests

- Pour tester si le réseau fonctionne : **ping**
- Pour en savoir un peu plus sur le chemin suivi : **traceroute**

```
[marco@pc1   marco ] ping  lit .ulb .ac .be
PING  lit .ulb .ac .be  (164.15.123.4)     from
  80.200.121.67     : 56(84)  bytes  of  data .
64  bytes  from  lit3 .ulb .ac .be  (164.15.123.4):
  icmp_seq  =1  ttl =245  time =22.4  ms
64  bytes  from  lit3 .ulb .ac .be  (164.15.123.4):
  icmp_seq  =2  ttl =245  time =72.0  ms
64  bytes  from  lit3 .ulb .ac .be  (164.15.123.4):
  icmp_seq  =3  ttl =245  time =18.9  ms
64  bytes  from  lit3 .ulb .ac .be  (164.15.123.4):
  icmp_seq  =4  ttl =245  time =19.7  ms
```

Tests

```
[marco@pc1    marco ] /usr /sbin /traceroute    lit .ulb .ac. be
traceroute    to  lit .ulb .ac .be  (164.15.123.4),
     30  hops  max,  38  byte  packets
 1   1.121-200-80.    adsl .skynet  .be  (80.200.121.1)
     9.516   ms   9.930   ms   8.941   ms
 2   at-0-2-0-36.    adsl2  .02 bnc. skynet  .be  (194.78.255.13)
     11.013   ms   9.966   ms   8.898   ms
 3   g0-0-0.  intl1 .02ixb  .skynet  .be  (194.78.0.45)
     10.877   ms   8.916   ms   9.965   ms
 4   ge. m160. ext. science  .giga .belnet  .net  (194.53.172.65)
     11.027   ms   10.901   ms   9.989   ms
 5   oc192  .m160  .core  .science  .giga. belnet  .net  (193.191.1.1)
     9.954   ms   9.205   ms   10.038   ms
 6   *  *  *
 7   *  *  *
 8   *  *  *
 9   lit3 .ulb .ac. be  (164.15.123.4)
     14.269   ms   11.934   ms   13.940   ms
```

LE DOMAIN NAME SYSTEM

DNS

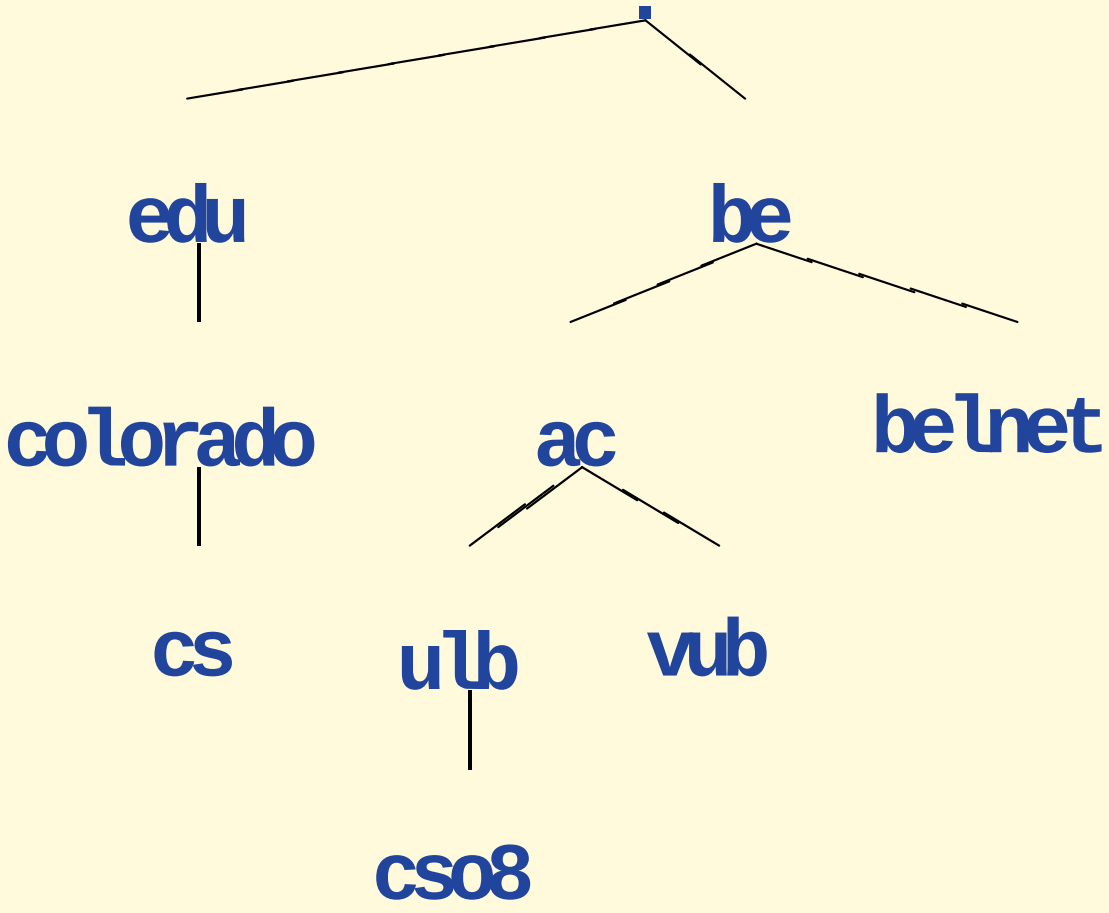
- Principes
- Client
- Ordre de résolution
- Serveur
- Tests

INTRODUCTION

- L'OS fonctionne avec des adresses IP uniquement
- Un utilisateur préfère utiliser un nom
- Mécanisme de traduction entre le nom et l'adresse IP
- Un fichier plat n'est plus gérable
  - conflit de noms
  - pertinence de l'info

PRINCIPES DE DNS

- **DNS** introduit une hiérarchisation des noms



PRINCIPES DE DNS

- **DNS** introduit une responsabilité distribuée
- A chaque niveau
  - Autorité responsable des noms
  - Des **serveurs DNS** (maîtres, esclaves) qui connaissent
    - les machines du niveau inférieur
    - le serveur global
- L'implémentation la plus utilisée est **BIND**

LE CLIENT DNS

- Tout serveur sera aussi client
- On parle aussi de **résolveur**
- Configuré via le fichier **/etc /resolv .conf**

ex : pour les machines du NO, le fichier doit ressembler à

```
mcodutti@cso8    :cat  /etc /resolv  .conf
domain                ulb .ac .be
nameserver            164.15.125.1
nameserver            164.15.59.200
```

- Le fichier **hosts** doit toujours contenir les machines essentielles

ORDRE D'UTILISATION DU DNS

- DNS et **hosts** peuvent coexister
- Dans ce cas, il s'agit de savoir dans quel ordre ils seront utilisés
- Il existe une vieille technique (**host.conf**) et une nouvelle (**nsswitch.conf**)
- Certains vieux systèmes (comme **SunOS**) imposent l'ordre

Ordre d'utilisation du DNS

- Le fichier **/etc /host .conf**

```
order   hosts , bind
```

- Le fichier **/etc /nsswitch .conf**
  - est plus général

```
hosts :         nis  dns  [NOTFOUND  =return  ] files
```

  - ou encore

```
hosts :         files  nis  dns
```

SERVEUR DNS

- Pour chaque noeud, on définit
  - Un serveur principal qui contient la base de données
    (**resu1** à l'ULB)
  - Un ou des serveurs auxilliaires qui reprennent une copie de la base
    (**plaine1** à l'ULB, **164.15.125.1**)
  - La mise à jour est régulière
  - Le démon impliqué est **named**

Serveur DNS

- Serveur maître / esclave
- Serveur cache
- Serveur autoritaire ou non
- Serveur récursif ou non

SERVEUR DNS

- Le démon se base sur un fichier de configuration : **`/etc/named.conf`**
- Possède sa syntaxe propre et reprend différentes informations comme
  - Le type de serveur
  - Les domaines (zones) gérés
  - Des options globales

SERVEUR DNS

- Egalement utilisé pour la **résolution inverse**

  IP vers nom
- Exemple :

  l'adresse **192.168.25.2**

  aura comme nom **2.25.168.192.in-addr.arpa**
- Il y aura donc 2 fichiers

  (sens direct et sens inverse)

SERVEUR DNS

Exemple : Voici le fichier d'un serveur à l'ESI

```
options   {
      directory    "/var /named ";
      };
zone   "labozen  .esi .be" in  {
      type  master  ;
      file  "labozen  .hosts ";
      };
zone   "0.16.172.   in- addr .arpa " in  {
      type  master  ;
      file  "labozen  .rev ";
      };
zone   "0.0.127.   in -addr  .arpa " in  {
      type  master  ;
      file  "local .rev ";
      };
```

SERVEUR DNS

- La base de données proprement dite pour une zone
  - Est un fichier texte
  - Facilement lisible et éditable
  - Se trouve dans un fichier spécifié dans le fichier de configuration

Serveur DNS

Exemple : Voici le fichier labozen.hosts à l'ESI

```
;;
$TTL   86400
@   IN  SOA  toutatis   .labozen   .esi .be .
           root .toutatis   .labozen   .esi .be .  (
   99022701
   28800
   7200
   604800
   86400   )
;;
   NS  toutatis   .labozen   .be.
```

SERVEUR DNS

Exemple : suite du fichier...

```
localhost       A       127.0.0.1
toutatis        A       172.16.0.1
                TXT     "Serveur    Labo    Zen "
www             CNAME   toutatis
smtp            CNAME   toutatis

kyo             A       172.16.0.2
iznogoud        A       172.16.0.3
```

SERVEUR DNS

Exemple : Voici le fichier labozen.rev à l'ESI

```
;;
$TTL  86400
@   IN  SOA  toutatis  .labozen  .esi .be .
          root .toutatis  .labozen  .esi .be . (
   99022701
   28800
   7200
   604800
   86400   )
;;
   NS  toutatis  .labozen  .be.
1  IN  PTR  toutatis
2  IN  PTR  kyo
3  IN  PTR  iznogoud
```

SERVEUR DNS

- Que faire pour des domaines qui ne sont pas des multiples d'octets ?
- ex : 128.138.243.0/26
- Qui va contrôler la zone ?
- OK si une même autorité mais pas si plusieurs autorités
- Le champ CNAME offre une astuce

SERVEUR DNS

- Au niveau du serveur

```
1  IN  CNAME   1.0-63
2  IN  CNAME   2.0-63
...
63  IN  CNAME   63.0-63
64  IN  CNAME   64.64-127
65  IN  CNAME   65.64-127
...
0-63   IN  NS  ns1 .client1  .com
0-63   IN  NS  ns2 .client1  .com
64-127   IN  NS  ns1 .client2  .com
```

Serveur DNS

- Résumons les champs les plus fréquents
  - **SOA**
  - **NS**
  - **A**
  - **CNAME**
  - **PTR**
  - **TXT**
  - **LOC**

TEST

- Pour tester le bon fonctionnement de DNS
  - **nslookup** (en abandon)
  - **dig**
  - **host**

Test

```
[marco@pc1    etc ] nslookup    -sil   lit .ulb .ac. be .
Server  :             195.238.2.21
Address  :            195.238.2.21#53

Non -authoritative      answer  :
Name :     lit .ulb .ac .be
Address  : 164.15.123.4
```

TEST

```
[marco@pc1    etc ] dig  lit .ulb.ac  .be.
(...)
;; ANSWER  SECTION  :
lit. ulb.ac.be  . 85953  IN  A 164.15.123.4

;; AUTHORITY   SECTION:
ulb. ac.be.       85421  IN  NS  resu1 .ulb.ac  .be.
ulb. ac.be.       85421  IN  NS  vnet3 .vub.ac  .be.

;; ADDITIONAL    SECTION  :
resu1 .ulb.ac  .be.  71631  IN  A  164.15.59.200
vnet3 .vub.ac  .be.  54     IN  A  134.184.15.13
(...)
;; SERVER  : 195.238.2.21#53(195.238.2.21)
(...)
```

TEST

```
[marco@pc1    etc ] dig  -x  195.238.2.21
(...)
;;  ANSWER   SECTION  :
21.2.238.195.    in- addr .arpa  .  2948   IN  PTR  dnspool1   .skynet   .be.

;;  AUTHORITY    SECTION  :
2.238.195.   in -addr  .arpa  .  71341      IN  NS  ns1 .skynet   .be .
2.238.195.   in -addr  .arpa  .  71341      IN  NS  ns2 .skynet   .be .
2.238.195.   in -addr  .arpa  .  71341      IN  NS  ns3 .skynet   .be .
2.238.195.   in -addr  .arpa  .  71341      IN  NS  ns4 .skynet   .be .

;;  ADDITIONAL    SECTION  :
ns1 .skynet   .be .              142        IN  A   195.238.3.17
ns2 .skynet   .be .              192        IN  A   195.238.3.18
ns3 .skynet   .be .              192        IN  A   195.238.3.19
ns4 .skynet   .be .              192        IN  A   195.238.3.20
(...)
;;  SERVER  :  195.238.2.21#53(195.238.2.        21)
```

Test

```
[marco@pc1 etc ] cat /etc/resolv.conf
search localdomain skynet.be
nameserver 195.238.2.21
nameserver 195.238.2.22
```

```
[marco@pc1 etc ] host lit.ulb.ac.be.
lit.ulb.ac.be has address 164.15.123.4
```

Test

```
[marco@pc1    etc ] dig  .  ns
(...)
;;  ANSWER   SECTION  :
.   329525    IN        NS        e.root-servers  .net .
.   329525    IN        NS        d.root-servers  .net .
.   329525    IN        NS        a.root-servers  .net .
.   329525    IN        NS        h.root-servers  .net .
.   329525    IN        NS        c.root-servers  .net .
.   329525    IN        NS        g.root-servers  .net .
.   329525    IN        NS        f.root-servers  .net .
.   329525    IN        NS        b.root-servers  .net .
.   329525    IN        NS        j.root-servers  .net .
.   329525    IN        NS        k.root-servers  .net .
.   329525    IN        NS        l.root-servers  .net .
.   329525    IN        NS        m.root-servers  .net .
.   329525    IN        NS        i.root-servers  .net .
```

TEST

(suite...)

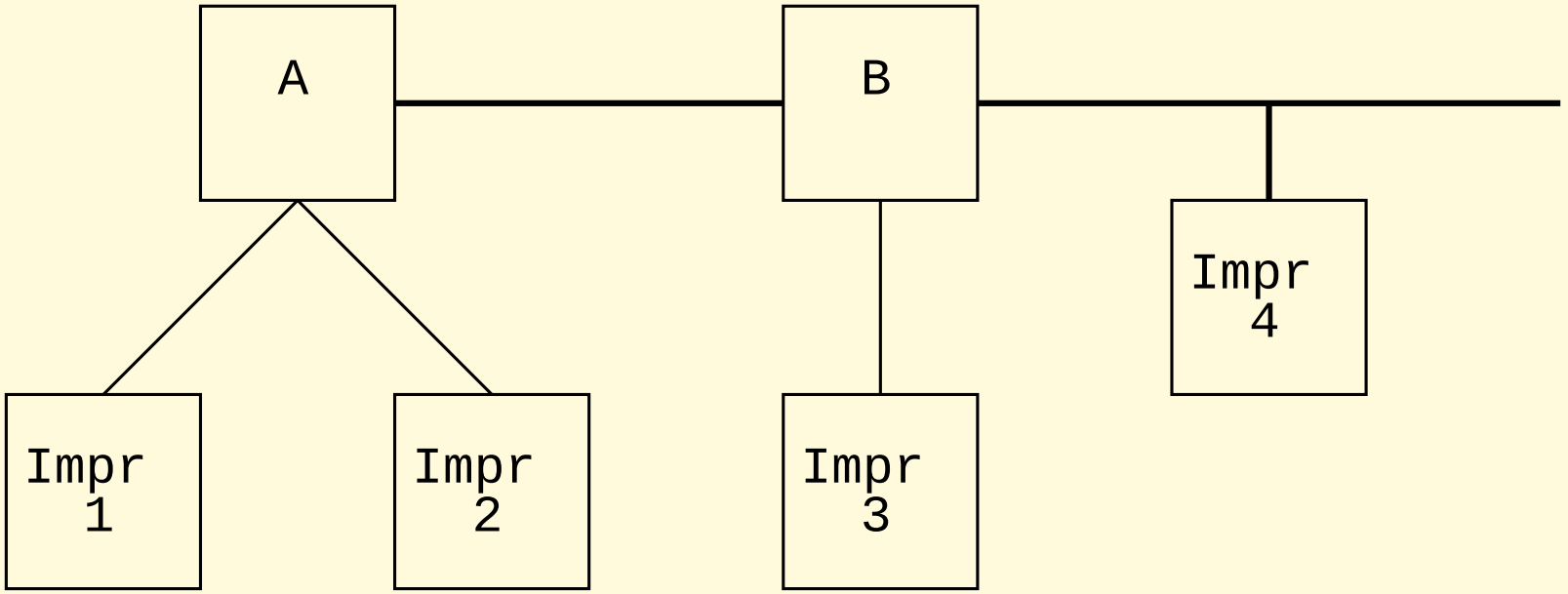
```
;; ADDITIONAL    SECTION  :
e.root -servers  .net .  415925   IN  A  192.203.230.10
d.root -servers  .net .  415925   IN  A  128.8.10.90
a.root -servers  .net .  415925   IN  A  198.41.0.4
h.root -servers  .net .  415925   IN  A  128.63.2.53
c.root -servers  .net .  415925   IN  A  192.33.4.12
g.root -servers  .net .  415925   IN  A  192.112.36.4
f.root -servers  .net .  415925   IN  A  192.5.5.241
b.root -servers  .net .  415925   IN  A  128.9.0.107
j.root -servers  .net .  415925   IN  A  192.58.128.30
k.root -servers  .net .  415925   IN  A  193.0.14.129
l.root -servers  .net .  415925   IN  A  198.32.64.12
m.root -servers  .net .  415925   IN  A  202.12.27.33
i.root -servers  .net .  415925   IN  A  192.36.148.17
```

LE SYSTÈME D'IMPRESSION

L'IMPRESSION

- Présentation
- Le système **BSD**
- Le système **SysV**
- Le système **LPRng**
- Quelques outils intéressants

PRÉSENTATION

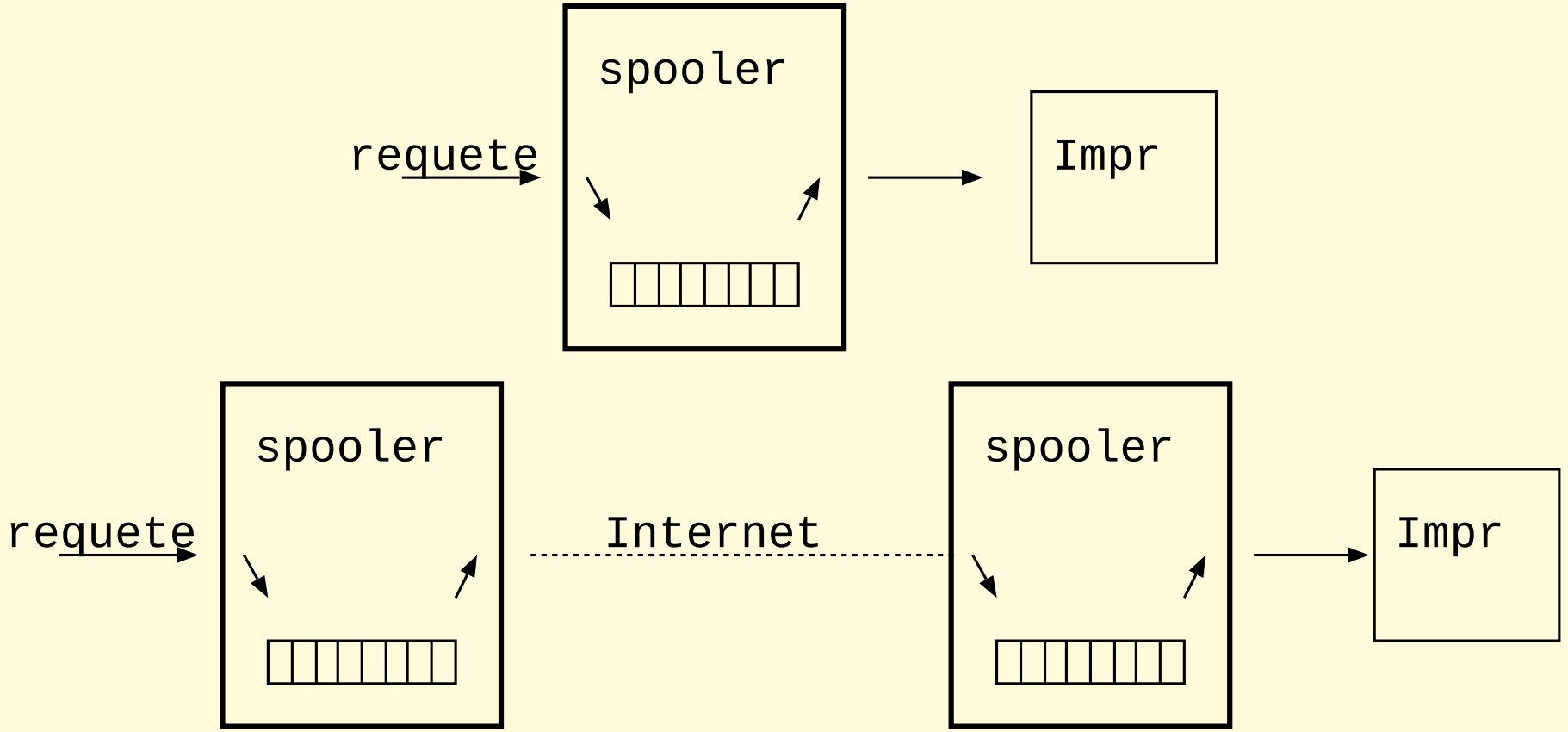


- Imprimantes facilement partageables d'une machine à l'autre
- Totalement transparent pour l'utilisateur

PRÉSENTATION

- Il existe plusieurs systèmes d'impressions
  - Celui de **BSD**
  - Celui de **AT&T** , fort différent
  - LPRng apparu il y a quelques années
  - CUPS encore plus récent et basé sur le protocole IPP

LE SPOOLING


spooler
requete
Impr
spooler
requete
Internet
spooler
Impr

LES LANGAGES

- Il existe différents langages (**PDL**, Page Description Language) dans le monde des imprimantes
- Les plus connus sont Postscript et PCL
- Souvent, une conversion du document devra être effectuée avant l'envoi à l'imprimante
- Cela se fera via l'utilisation d'un **filtre**

LE SYSTÈMME BSD

- **lpr** : envoie un fichier au spooler
- **lpd** : le démon qui gère le spoole
- Requêtes stockées dans
  /var /spool /lpd /printername /
- **/etc /printcap** : décrit les imprimantes
- **lpq** : visualiser les fichiers en attente
- **lprm** : enlever un fichier de la file
- **lpc** : administration des imprimantes (stop, restart, . . .)

LE FICHIER PRINTCAP

- **/etc /printcap** est un fichier ASCII
- Décrit les imprimantes une après l'autre
- L'imprimante peut être
  - locale (directement reliée à l'ordinateur)
  - distante (attachée physiquement à un autre ordinateur)
  - réseau (directement reliée au réseau)

LE FICHIER PRINTCAP

- Exemple : imprimante locale

```
math |lp| DEClaser   5100  at  Math  Gene  Bunker  :\
         :lp =/dev/ttyS1 :\
         :br #19200:\
         :sd =/var/spool/lp/math :\
         :af =/var/spool/lp/math/acct :\
         :lf =/var/spool/lp/math/log :\
         :sh :
```

- Exemple : imprimante à distance

```
1|cso |Epson   ex-1000   at  CSO  (N4 ):\
         :lp =:\
         :rm =cso6 :\
         :rp =cso :\
         :sd =/usr/spool/lp/cso :
```

POUR IMPRIMER

- Pour imprimer un fichier :

  lpr -Pprinter fichier

- Imprime sur l'imprimante spécifiée (dans l'ordre)
  - Par l'option **-P**
  - Par la variable d'environnement **$PRINTER**
  - L'imprimante **lp**
  - La première imprimante de **printcap**

POUR IMPRIMER

- Pour visualiser les fichiers actuellement dans la file d'attente : **lpq -Pprinter**

```
mcodutti@lit3    :lpq  -Pcso
cso  is  ready  and  printing
Rank  Owner       Job  Files  Total  Size
1st   ssaeedni    657  input  1908  bytes
```

- Pour effacer un fichier de la file : **lprm -Pprinter job**

```
mcodutti@lit3    :lprm  -Pcso  657
```

LE SPOOLING

- Le dossier désigné pour le spooling contient des fichiers de log, de statut, . . .
- Pour chaque fichier
  - Un fichier de la forme **cfA657 . . .** : infos sur le fichier (qui, quand, . . .)
  - Un fichier de la forme **dfA657 . . .** : le fichier à imprimer

GESTION DES IMPRIMANTES

- La commande est `lpc` dont la syntaxe est :

  `lpc command printer` , où `command` est

  enable/disable : ouvrir/fermer une file d'attente

  start/stop : déclencher/interrompre l'impression

  down/up : combine les 2 précédents

  clean : vider la file d'attente

  topq : placer un fichier en tête de liste

  restart : redémarrer le spooler

LES PERMISSIONS

- L'impression a distance est régulée via un fichier de permissions sur le serveur
- Son nom est **/etc /hosts .lpd**
- Liste toutes les machines qui peuvent imprimer sur ses imprimantes
- Le contrôle se fait au niveau de la machine et pas de l'utilisateur
- Le démon **lpd** consulte également le fichier **hosts .equiv**

Les filtres

- Il est parfois nécessaire de modifier le format du fichier imprimé
  - Pour commencer par une séquence d'initialisation de l'imprimante
  - Pour modifier l'ASCII en un format reconnu par l'imprimante
  - Pour imprimer plusieurs page sur une feuille
  - . . .

LES FILTRES

- Option **if** dans le fichier de configuration
- Exemple :

```
math |lp:\
        :lp =/dev/ttyS1 :\
        :br #19200:\
        :sd =/var/spool/lp/math :\
        :if =/usr/lib/lpd/filtre1  :
```

- où **filtre1** est un script recevant le fichier en entrée et fournissant la version modifiée en sortie

UTILISATION NON STANDARD

- On peut utiliser les capacités de **spooling** pour autre chose que l'impression
- Exemple : on peut l'utiliser pour gérer facilement un journal commun

```
journal  :\
          :lp =/dev/null :\
          :sd =/var/spool/lpd/journal  :\
          :if =/usr/local/lib/journal  :
```

avec le script

```
#!/bin/bash
cat  >>  /var/log/journal
```

L'IMPRESSION SOUS SYSV

- N'est guère utilisé que par **Solaris** et HP
- A la base, pas d'impression via le réseau
- Avantage sur **BSD** : les **classes**
  - Plusieurs imprimantes peuvent être inscrites dans la même classe
  - Il est possible d'imprimer sur une **classe** et l'impression se fera sur l'imprimante la moins chargée de la classe

CONFIGURATION

- Les fichiers de configuration sont manipulés via la commande **lpadmin**
- ex : Pour ajouter l'imprimante printer1 de type model1 connectée sur /dev/term/a

```
lpadmin    -pprinter1     -v/ dev /term /a  -mmodel1
```

- ex : Même situation mais la configuration est copiée de l'imprimante printer2

```
lpadmin    -pprinter1     -v/ dev /term /a  -eprinter2
```

CONFIGURATION

- ex : Pour ajouter une imprimante à une classe (qui sera créée si elle n'existe pas encore)

```
lpadmin    -pprinter1     -cclass1
```

- ex : Pour définir l'imprimante par défaut

```
lpadmin    -dprinter1
```

- + bien d'autres options

IMPRESSION

- Pour imprimer un fichier :

  lp -d printer fichier

- Si l'option **-d** n'est pas spécifiée, on regarde la variable d'environnement **$LPDEST**
- Si elle n'est pas définie, on prend l'imprimante par défaut (définie par **lpadmin -d**)

GESTION

- **lpstat** donne des infos sur le statut d'une imprimante, l'état du démon, le contenu des classes, . . .
- **cancel** permet d'enlever un fichier d'une file
- **disable** et **enable** pour suspendre/reprendre l'impression
- **lpmove** pour déplacer un travail d'une file à l'autre

Le système LPRng

- Système d'impression plus récent
- Tentant de fusionner le meilleur des deux précédents systèmes
- Introduit également une plus grande sécurité via **Kerberos** ou **PGP**
- D'un point de vue pratique, fournit une compatibilité avec les commandes **BSD** et **SysV**

Le système LPRng

- La configuration de **LPRng** se fait via différents fichiers
- Pour la configuration du système d'impression : **/etc /lpd .conf**
  - Plus de 150 options
  - On indique où sont les dossiers, les différentes options du démons, . . .

LE SYSTÈME LPRNG

- Pour la configuration des permissions :

  /etc/lpd.perms

  - Permet d'accepter (ou refuser) l'impression (ou le contrôle du spool ou l'administration) sur une imprimante à des utilisateurs bien précis sur des machines bien précises
  - Un exemple de ligne est :

```
ACCEPT SERVICE=P,R,M,Q REMOTEHOST=lit
> REMOTEUSER=mcodutti PRINTER=litpr1
```

LE SYSTÈME LPRNG

- Pour la définition des imprimantes :

  /etc /printcap

  - Compatible avec les versions **BSD**
  - Introduit quelques nouveautés (comme un filtre pour une imprimante réseau)
  - + commande **checkpc** (vérifie la validité du fichier)

Le système LPRng

- Commande identique à celle de **BSD**
- Quelques options en plus
- Citons la possibilité d'imprimer sur une imprimante réseau qui n'a pas été définie dans **printcap**

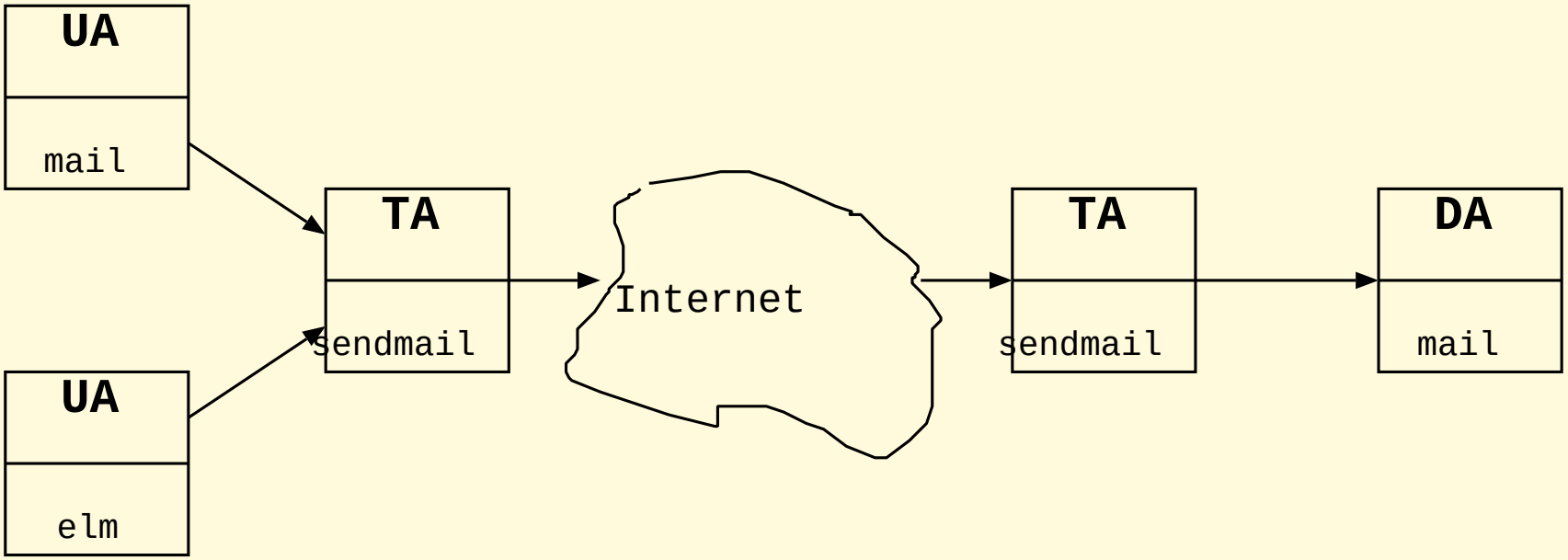
```
lpr  -P  litpr1@lit1    brol
```

QUELQUES OUTILS INTÉRESSANTS

- **Ghostscript** permet de convertir un fichier Postscript en de nombreux formats différents
- **mpage** convertit de l'ASCII en Postscript (+ mise en page)
- **enscript** : outil équivalent

Le système de Mail

Le système de Mail



- + Agent d'accès
- + Agent d'envoi

Structure d'un message

- Un message est composé de 3 parties
  - l'**enveloppe** :

    détermine la destination du message.
  - les **en-têtes** :

    ensemble de paires proriétés/valeurs
  - le **corps** même du message

Structure d'un message

- Exemple d'entête

```
Return -Path : <elevy@litpc49 .ulb .ac .be>
Delivered -To: marco@localhost .pc1. localdomain
Received : from localhost (localhost .localdomain
[127.0.0.1]) by pc1 .localdomain (Postfix )
with ESMTP id 33C83346E5
for <marco@localhost >;
Tue, 22 Apr 2003 08:01:24 -0400 (EDT )
Received : from pop .tiscali .be [62.235.14.101]
by localhost with POP3
(fetchmail -6.1.0) for marco@localhost ;
Tue, 22 Apr 2003 14:01:24 +0200 (CEST )
Received : from guppy .vub. ac.be (134.184.129.2)
by mail .tiscali .be (6.7.015)
id 3E9BA4650002B753
for Marco -Codutti@tiscali .be;
Tue, 15 Apr 2003 12:56:20 +0200
```

STRUCTURE D'UN MESSAGE

```
Received : from mach .vub. ac.be (mach .vub .ac .be
          [134.184.129.3]) by guppy .vub .ac. be
          (8.9.1 b+ Sun/3.17.1. ap (guppy )) id MAA00137 ;
          Tue, 15 Apr 2003 12:56:20 +0200 (MET DST)
          for <Marco -Codutti@tiscali .be >
Received : from litpc49 .ulb .ac .be (litpc49 .ulb .ac.be
          [164.15.123.99]) by mach. vub. ac.be
          (8.9.3/3.13.3. ap (mach )) id MAA27651 ;
          Tue, 15 Apr 2003 12:56:18 +0200 (MET DST)
          for <Marco -Codutti@tiscali .be >
Subject : cotes projets
From : Eythan Levy <elevy@litpc49 .ulb. ac.be >
To: Marco Codutti <Marco -Codutti@tiscali .be>
X-Mailer : Ximian Evolution 1.0.8-3 mdk
Date : 15 Apr 2003 12:55:00 +0200
Message -Id: <1050404100.9364.36. camel@litpc49 .ulb. ac.be >
X-Evolution -Source : mbox :/var /spool /mail /marco
```

Les alias

- Les alias peuvent être définis dans différents endroits
  - Le fichier de configuration de certains AU
  - Le fichier de configuration de l'AT
  - Le fichier de transmission d'un utilisateur (**.forward**)

LES ALIAS AU NIVEAU DE L'AT

- Fichier **/etc /aliases**

  (ou **/etc /mail /aliases** ou **/usr /lib /aliases**)

- Définit des règles d'alias pour modifier le destinataire local d'un mail

- Exemple :

```
marco :          mcodutti
techniciens   :  mcodutti@ulb    .ac .be ,
                         gpaquet@ulb    .ac .be
prof :           :include  :/ etc /aliases   .prof
suggestions   :  "/ dev /null "
info :           "| programme   "
```

LES ALIAS AU NIVEAU DE L'AT

- Fichier compilé pour accès rapide
- Utiliser **newaliases** pour recompiler

LES FORWARD

- Un utilisateur peut décider de *forwarder* ses mails
- Permet, par exemple, de tout concentrer dans une seule boîte si on en possède plusieurs
- Via le fichier `~/.forward`
- Exemple

```
cso8$  cat  ~/. forward
\mcodutti@cso    .ulb .ac .be ,
Marco -Codutti@tiscali     .be
```

- Le `\` prévient les boucles.

SENDMAIL ET DNS

- Pourquoi un courriel addressé à **mcoduttti@ulb .ac .be** arrive-t-il ?
- DNS contient également un champ MX qui permet d'indiquer une redirection pour le mail
- Pour n'importe quel niveau de hiérarchie
- Un MTA va consulter le DNS si il est configuré sur la machine

SENDMAIL ET DNS

Exemple :

```
[marco@pc1   marco ]$ dig ulb.ac.be MX
(...)
;; QUESTION   SECTION :
;ulb.ac.be.                          IN       MX

;; ANSWER   SECTION :
ulb.ac.be.  71802   IN  MX  150  mailhost   .vub.ac.be.
ulb.ac.be.  71802   IN  MX  100  mailhost   .ulb.ac.be.
(...)
```

SENDMAIL

- **sendmail** est le MTA le plus utilisé
  (on rencontre aussi **postfix** ou **qmail**)
- Implémente **(E)SMTP**
- Configuré via le fichier **sendmail .cf**
- Reprend l'emplacement d'autres fichiers
  - **/etc /aliases** : la liste des alias
  - **/etc /aliases .db** : version compilée
  - **/var /spool /mqueue /** : file d'attente des messages

SENDMAIL

- But principal : manipuler les adresses
- Ex : Imposer un domaine pour l'expéditeur

```
S1
R$+<@$*> $:$1 <@ulb .ac. be>
```

  - S1 correspond à la section : expéditeur
  - $+ correspond à 1 à plusieurs mots
  - $* correspond à 0 à plusieurs mots
  - $: stoppe l'utilisiation de la règle
  - $1 Premier match de la partie gauche
  - < et > : ajoutés par d'autres règles

SENDMAIL

- On peut parler directement à **sendmail**

```
mcodutti@cso1   :telnet   resu1   25
Trying   164.15.59.200..     .
Connected    to  resu1 .ulb .ac .be.
Escape   character    is  '^]'.
220  resu1 .ulb .ac .be  ESMTP   sendmail    8.8.8/3.17.1.    ap  (r
     ready   at  Sat ,  26  Apr   2003   16:25:33    +0200   (MEST )
ehlo  cso1
250- resu1 .ulb .ac .be  Hello   cso1 .ulb .ac .be
     [164.15.130.31]     ,  pleased    to  meet   you
250- EXPN
250- VERB
250-8 BITMIME
250- SIZE   16000000
250- ONEX
250- ETRN
```

SENDMAIL

```
250- XUSR
250  HELP
mail  from :< mcodutti@cso    .ulb .ac .be >
250  <mcodutti@cso    .ulb .ac. be >...  Sender   ok
rcpt  to :< mcodutti@cso1    .ulb .ac .be >
250  <mcodutti@cso1    .ulb .ac .be >...  Recipient    ok
data
354  Enter  mail , end  with  "."  on  a  line  by  itself
test
.
250  QAA06040   Message   accepted   for  delivery
quit
221  resu1 .ulb .ac .be  closing   connection
Connection    closed   by  foreign   host .
```

PROCMAIL

- **procmail** est un MDA (agent de livraison)
- Très utilisé dans le monde **Unix**
- Reçoit le courriel du MTA lorsqu'il est à destination
- Remplit un peu le même rôle que le fichier alias mais
  - plus souple
  - plus riche
  - plus puissant

PROCMAIL

- Basé sur un ensemble de règles
- Syntaxe très riche
- Les messages peuvent être filtrés sur
  - le contenu d'un entête ou du message
  - sa taille
- Un message peut être
  - renvoyé à une autre adresse
  - envoyé à une autre commande
  - ajouté à un fichier

PROCMAIL

- Les règles sont dans des fichiers de configuration
  - **/etc /procmailrc**
  - **~/. procmailrc**
- Exemple

```
:0
* > 100000
/dev /null

:0
*^Subject  :. *archive
archive  -‘date  +%B‘
```

PROCMAIL

- Lancé via une règle du MTA.

  Exemple : avec **sendmail**

```
S0
R$*<@some .where >$*
          $# procmail
```

- ou via le fichier **.forward**

```
"| exec /usr/bin/procmail "
```

Les listes de diffusion

- Le fichier **/etc /aliases** permet déjà de gérer une liste de diffusion
- Il lui manque beaucoup de fonctionnalités et de facilités
  - Nécessité de droits **root**
  - Tout le monde peut envoyer à la liste
  - Pas de notion de modérateur
  - . . .

LES LISTES DE DIFFUSION

- D'où systèmes spécialisés comme

  majordomo

- Exemple : Entrées du fichier **aliases**

```
test :          "|/ usr /local /majo /wrapper
   resend   -l test  test -list "
test -list :   :include  :/ usr /local /majo /lists  /test
owner -test :  mcodutti
test -request  : "|/ usr /local  /majo /wrapper
   majordomo    -l  test "
```

LES LISTES DE DIFFUSION

- Pour écrire à la liste

  mail test@host

- Pour se désabonner

  mail majordomo@host avec **unsubscribe test**

  mail test-request@host avec **unsubscribe**

Intégration Windows/Linux

Intégration Windows/Linux

- Introduction au réseau Windows
- Samba

Les bases du réseau Windows

- NetBIOS : extension de BIOS sur réseau local
- Peut s'appuyer sur NETBEUI Frame Protocol (non routable)
- Ou directement sur TCP/IP (on parle de NBT)
- NetBIOS introduit la notion de **domaine**
- Notion différente de celle de DNS
- WINS : lien entre nom NetBios et adresse IP

SAMBA

- SMB est le protocole qui permet a des machines de partager des fichiers et des imprimantes
- SAMBA : une implémentation du protocole SMB
- Basé sur 2 serveurs

smbd démon de partage (coté serveur)

nmbd gestion des noms NetBIOS

SAMBA

- Configuré via le fichier **smb.conf**
- Fichier texte formé de sections dont
  - `[global]` pour les options globales
  - `[homes]` les dossiers personnels
  - `[printers]` toutes les imprimantes d'un coup
  - une section pour chaque ressource (service)

SAMBA

- Exemple : pour la section **[global]**

```
[global ]
    workgroup    = MYGROUP
    hosts  allow  = 192.168.1.
                    192.168.2.    127.
    load  printers    = yes
    guest  account    = pcguest
    security    = user
```

SAMBA

- Exemple : pour la section **[homes]**

```
[homes ]
    comment    = Home  Directories
    browseable    = no
    writable    = yes
```

- Exemple : pour la section **[printers]**

```
[printers  ]
    comment    = All  Printers
    browseable    = no
    guest  ok  = no
    writable    = no
    printable    = yes
```

SAMBA

- Exemple : pour partager un dossier public en lecture

```
[public ]
    comment    = Public   Stuff
    path   = /home /samba
    public    = yes
    writable     = yes
    printable      = no
    write   list   = @staff
```

SAMBA

- On peut inclure un fichier. **Intérêt ?**
- On peut aussi utiliser des variables spéciales
  - %M : nom du client
  - %u : nom d'utilisateur
  - %g : groupe principal de l'utilisateur
  - %h : nom du serveur
  - ...

SAMBA

- Le serveur doit pouvoir identifier le client (et son utilisateur)
- Spécifié via l'option **security**
  - **security =share** . Pas d'autentification. Accès aux partages publiques (**guest ok = yes**)

SAMBA

- option **security** (suite)
  - **security =user** , son nom et mot de passe est reconnu par le serveur (dans **/etc /passwd**)
  - **security =server** , la vérification est déléguée à un autre serveur (peut-être Windows) (spécifié par l'option **password server = ...**)

SAMBA COTÉ CLIENT

- **Samba** offre également le côté client
  - **smbmount** pour monter un partage **SMB**

    `smbmount \\server\dir \localdir`
  - **smbclient** est un programme interactif qui fait un peu de tout
    - transfert
    - impression
    - tar
    - ...

SAMBA COTÉ CLIENT

- Pour imprimer facilement
  - Définir l'imprimante dans printcap

    avec un filtre (`if =/usr /bin /smbprint`)
  - Imprimer de façon classique

OUTIL DE CONFIGURATION

- Sur les dernières versions de **Samba**
  - Interface graphique de configuration
  - Applicaton Web sur

    http :// localhost :901

La sécurité

La sécurité

- Les règles de base
- Les problèmes potentiels
- Les outils de sécurité

La sécurité

- **Unix** a été créé par des **chercheurs** pour des chercheurs
- La sécurité n'était pas une priorité
- Elle est **binaire**
- Fonctions administratives en dehors du noyau
- Les distributions sont rarement à jour
- Les sources sont disponibles

LES RÈGLES DE BASE

- Ne pas laisser traîner de fichiers tentants (**cryptographie**)
- Boucher les trous de sécurité découverts (**patch**)
- Utiliser des programmes de test d'intégrité (**cops** , **satan** , **crack** ,. . .)
- Se documenter sur la sécurité en **Unix**
- Inspecter le système en permanence (**log** , scripts)

Les règles de base

La sécurité peut être compromise par

- Les **utilisateurs** (et administrateurs)

  Souvent le maillon faible

  Ignorance, laxisme, incompétance, . . .
- Les **logiciels** : erreurs connues ou à venir
- La mauvaise **configuration** : le mode sécurisé n'est pas toujours le mode par défaut, besoin de souplesse, . . .

LE FICHIER DES MOTS DE PASSE

- **Très sensible !**
- A vérifier régulièrement (via script dans cron)
  - Pas de compte sans mot de passe
  - Différence par rapport à la veille
  - Permissions du fichier
  - Mots de passe adéquats :

    crack , **npasswd** , . . .
  - Pas de UID dupliqué

LE FICHIER DES MOTS DE PASSE

- Pas de login de groupe
- Même pas pour les administrateurs
- Politique de renouvellement des mots de passe
- Shells valides

LE SETUID BIT

- Certaines failles découvertes
- D'autres viendront
- A éviter au maximum
- A utiliser avec précaution dans ses scripts (surtout les shells)
- On peut bloquer le setuid sur des partitions au montage
- Dresser la liste des scripts et surveiller les changements

DIVERS PROBLÈMES

- Terminaux sécurisés
- Les fichiers **hosts** **.equiv** et **.rhosts** (préférer SSH)
- Le démon **fingerd**
- Faire attention aux services réseaux : **NIS** , **NFS**
- De façon générale, ne permettre que les démons nécessaires

LES DÉMONS ET INETD

- Trop de démons à lancer au début
- **super démon inetd** : les lance à la demande
- Fichier **/etc /services** (extrait)

```
ftp                    21/ tcp
```

- Fichier **/etc /inetd .conf** (extrait)

```
ftp  stream   tcp  nowait   root
       /usr /sbin /in .ftpd  in .ftpd
```

- **xinetd** est une extension récente qui ajoute, notamment, du logging

OUTILS LIÉS À LA SÉCURITÉ

nmap : collecter plein d'infos sur le réseau

- La liste des machines actives

```
[marco@pc1    marco ]$ nmap  -sP  164.15.125.1/24
Starting   nmap  V.  3.00  (  www .insecure   .org /nmap /  )
Host  plaine1  .ulb. ac.be  (164.15.125.1)
   appears   to  be  up .
Host  cercleinfo  -pc1 .ulb .ac .be  (164.15.125.10)
   appears   to  be  up .
Host  poseidon  .ulb .ac. be  (164.15.125.24)
   appears   to  be  up .
(...)
Host  di- net.ulb  .ac .be  (164.15.125.198)
   appears   to  be  up .
Host   (164.15.125.255)     appears   to  be  up .
Nmap  run  completed   --  256  IP  addresses   (50  hosts   up )
   scanned   in  10  seconds
```

OUTILS LIÉS À LA SÉCURITÉ

- Les ports ouverts sur une machine

```
[root@pc1 marco ]# nmap -sT resu1 .ulb .ac .be
Starting nmap V. 3.00 ( www .insecure .org /nmap / )
Interesting ports on resu1 .ulb .ac. be (164.15.59.200):
Port State Service
21/tcp open ftp
23/tcp open telnet
25/tcp open smtp
80/tcp open http
161/ tcp filtered snmp
162/ tcp filtered snmptrap
443/ tcp open https
445/ tcp filtered microsoft -ds
6000/ tcp open X11
6001/ tcp open X11 :1
Nmap run completed -- 1 IP address (1 host up)
scanned in 41 seconds
```

OUTILS LIÉS À LA SÉCURITÉ

- Le système que tourne une machine

```
root@pc1 marco ]# nmap -O resu1 .ulb .ac. be
Starting nmap V. 3.00 ( www .insecure .org /nmap / )
Remote operating system guess :
  Solaris 8 early access beta through actual release
Uptime 118.175 days (since Mon Jan 6 20:51:12 2003)
```

OUTILS LIÉS À LA SÉCURITÉ

- **SAINT** est le successeur de **SATAN**
  - Comme **nmap** analyse les ports réseaux
  - Mais en plus, y cherche des failles
  - Etablit un rapport complet en HTML
- **crack** tente de percer à jour les mots de passe
  - A besoin des versions chiffrées
  - Se base sur des dictionnaires
  - Y applique des règles de modification

OUTILS LIÉS À LA SÉCURITÉ

- **tcpd** : journaux sur les connexions TCP
  - Il suffit de modifier **/etc /inetd .conf**
  - Lancer **tcpd** à la place du service
  - Avec le service en argument
  - Exemple

```
ftp  stream   tcp  nowait   root
  /usr /local /bin /tcpd   in .ftpd
```

Outils liés à la sécurité

- **COPS** : sécurité d'un système local
  - Droits d'accès et mode des périphériques
  - Fichiers essentiels de **/etc**
  - Fichiers de démarrage
  - Droits sur les dossiers personnels
  - . . .

OUTILS LIÉS À LA SÉCURITÉ

- `tripwire` : surveille les fichiers systèmes
  - Sauve l'état d'un système stable dans une BD
  - Nom, date, taille, somme de vérification, ...
  - A la demande, cherche une différence avec l'état courant du système
  - La base devrait être sur un serveur en écriture seule

OUTILS LIÉS À LA SÉCURITÉ

- **ssh** : le shell sécurisé
  - Remplace **telnet** , **rlogin** , **rcp**
  - Tout est crypté même le mot de passe
  - Un démon (**sshd**) et deux commandes (**ssh** et **scp**)
  - Configuré via /etc/ssh/ssh_config

OUTILS LIÉS À LA SÉCURITÉ

- **ssh** : 4 modes d'authentification
  - mode A : comme avant. Utilise **/etc /hosts .equiv** et **/etc /ssh /shosts .equiv**
  - mode B : clé publique/privée pour vérifier l'hôte
  - mode C : mode B + clé publique/privée pour vérifier l'utilisateur
  - mode D : mot de passe démandé (mais en crypté)

Stratégies et éthique

Stratégies et éthique

- Stratégie
- Ethique
- En guise de conclusion . . .

Stratégies

- Limiter le partage des droits super utilisateur
- Prévoir tous les cas et définir une stratégie (méthodologie, procédure, . . .) pour chaque cas
- Bien documenter chaque cas
- Former et informer ceux qui sont concernés

SÉCURITÉ

Qu’est-ce qui est important ?

- Sécurité VS Service
- Sécurité VS Simplicité
- Sécurité VS Coût

Se préparer aux catastrophes

- Préparer des plans catastrophes
  - Intrusion
  - Problème d'environnement
  - Erreur humaine
  - Erreur de matériel

Relations avec les utilisateurs

- Le super utilisateur combine souvent les pouvoirs
  - Legislatif
  - Exécutif
  - Judiciaire
- C’est mauvais, il faut essayer de déléguer
- Faire signer une charte aux utilisateurs et aux administrateurs

ETHIQUE

- Avoir une éthique forte
- Résister aux tentations
- Résister aux pressions

ETHIQUE

Etude de cas

Un chef de Département envoye par erreur un courriel à tous les employés au lieu de l'envoyer uniquement aux membres du Conseil d'Administration. Il demande à l'Administrateur Système de le supprimer dans les boîtes aux lettres des personnes. **Que faire ?**

Etude de cas

Un administrateur système qui allait se marier et n'avait pas terminé ses préparatifs a donné à son meilleur ami (quelqu'un de compétent) la clé de son bureau et le mot de passe root (identique pour toutes les machines du site). Il n'y a pas eu de problème mais cette situation a été remarquée. **Que faire ?**

La grande règle

Un administrateur système est

- au service des utilisateurs
- un citoyen responsable