Configuration et gestion d'un serveur Linux

pour contrôler un réseau scolaire constitué de PC fonctionnant sous Window$

Notes rédigées par Gérard Swinnen, animateur pédagogique au SeDESS - Liège

Avertissement

Ces notes ne prétendent pas couvrir l'intégralité de l'installation d'un parc informatique ! Elles se veulent seulement un aide-mémoire pour les professeurs qui souhaitent mettre en place dans leur école un serveur Linux destiné à contrôler un réseau de type intranet, dans lequel les machines communiquant les unes avec les autres par l'intermédiaire du protocole TCP/IP. Nous supposerons que la plupart de ces postes de travail fonctionnent sous Window$ 95/98/Me.

Il ne faut pas se cacher que la gestion d'un parc d'ordinateurs peut facilement devenir une tâche très exigeante en termes d'heures de travail. Les notes que nous proposons ici ne sont qu'un résumé des informations essentielles pour commencer. Pour maîtriser vraiment le sujet, il faudra nécessairement se donner la peine de consulter des ouvrages de référence. Il en existe heureusement beaucoup, et généralement de fort bonne qualité. Nous pouvons recommander, entre autres :

- **Le système Linux** – par M Welsh, K. Dalheimer & L. Kaufman, 3e édition traduite de l'anglais - Editions O'Reilly – ISBN 2-84177-086-9
- **Grand livre Linux** - Micro Application - ISBN 2-7429-1426-9
- **Linux : Initiation et utilisation** – par J.P. Armspach, P. Colin, F. Ostré-Waerzeggers – Dunod, Paris, 2000 – ISBN 2 10 005150 4
- Les bases de l'administration système – par Aeleen Frish Editions O'Reilly – ISBN
- **Internet et Intranet sous Linux** - par H. Holz, B. Schmitt, A. Tikart Éditions Eyrolles - ISBN 2-212-09101-X
- **Samba, l'intro.** - par G. Carter & R. Sharpe - Campus press France - ISBN 2-7440-0745-5

Vous trouverez également une documentation pléthorique dans votre distribution Linux elle-même (par ex. dans les pages *man* et *info*, les fichiers *howto* et *package*). Se plonger dans cette documentation est parfois une longue aventure, mais vous pouvez toujours être certain que le renseignement qui vous manque s'y trouve caché quelque part ! En dernier ressort, vous pouvez également utiliser l'internet (recherches par mot-clés, forums de discussion Linux, etc.).

Cahier des charges

Bien que son étude soit tout à fait passionnante, nous n'avons pas l'intention de centrer notre propos sur le système d'exploitation Linux lui-même, mais plutôt sur la mise en place d'un réseau de type intranet en milieu scolaire, dont le serveur principal sera un PC performant fonctionnant sous Linux, et pour lequel nous avons établi le cahier des charges suivant (nos choix sont très certainement discutables, mais la place et le temps nous manquent pour les justifier en détail ici) :

- Les postes de travail seront pour la plupart destinés aux élèves. Ces postes seront des machines de type PC fonctionnant sous Window$ 95, 98 ou Me, et éventuellement XP, NT ou 2000. (La connexion de postes Mac ne devrait pas poser de gros problèmes, mais nous ne l'aborderons pas ici). Si la taille de leur disque dur le permet, on y réservera un espace pour l'installation de Linux avec un certain nombre de logiciels, mais cela n'est pas un objectif prioritaire.
- Un seul protocole de communication sera mis en place, à savoir le protocole TCP/IP.

- Les postes de travail seront configurés de telle manière à présenter aux utilisateurs un espace de travail personnalisé en fonction de la tâche à accomplir ou en fonction du groupe auquel ils appartiennent (bureau, icônes, applications accessibles, etc. personnalisés). En d'autres termes, le système devra comporter une gestion des utilisateurs et des groupes, laquelle devrait rester aussi simple que possible.
- Un maximum de fonctions et de services seront centralisés sur un serveur réseau :
  - Identification des utilisateurs avec authentification par mot de passe. Attribution d'un répertoire personnel pour chacun avec un espace disque limité (système de quotas disque).
  - Installation centralisée de tous les logiciels qui le permettent, afin de faciliter la maintenance (encyclopédies, logiciels didactiques, etc.)
  - Serveur web d'intranet, pour la mise en place de pages à usage interne pour les membres de l'institution scolaire (travaux d'élèves, notes de cours, etc.)
  - Sauvegarde d'une image comprimée de la partition Window$ de chaque poste de travail, afin de permettre la restauration rapide ce chacun d'eux en cas de problème (effacement de fichiers, contamination virale, fragmentation, etc.)
  - Serveur Proxy (filtre & cache) pour contrôler l'accès à l'internet de tous les postes
  - Routage internet par le serveur lui-même, ou mieux à travers une passerelle sécurisée avec un pare-feu (firewall). Connexion au réseau via RNIS ou ADSL.

Solutions retenues

Un système répondant aux spécifications de notre cahier des charges pourrait très certainement être mis en place sous le contrôle d'un serveur fonctionnant avec le système d'exploitation Window$ (NT, 2000 ou XP). Nous lui préférerons un serveur Linux pour un certain nombre de raisons :

- Gratuité, non seulement pour le système serveur lui-même, mais surtout pour la connexion des postes de travail (Rappel : Micro$oft exige le versement de royalties pour chaque connexion à son système serveur) et les logiciels accessoires (routage, quotas disque, serveur web, proxy, etc.)
- Stabilité du système. C'est évidemment un point essentiel : si le serveur réseau présente un comportement chaotique, le travail de tous est compromis et l'ambiance devient vite détestable. Quant à l'administrateur réseau (souvent un bénévole !), il a certainement mieux à faire que de chercher sans cesse à résoudre les problèmes de pannes et plantages divers.
- Exigence moindre en termes de ressources matérielles. Linux est beaucoup moins "gourmand" que les produits Micro$oft quant aux spécifications requises de la machine serveur. Il ne faudra cependant pas en conclure que l'on peut se contenter de n'importe quel PC pour jouer ce rôle. Si l'on souhaite mettre en place un système à la fois performant et facile à gérer, on sera amené à installer de nombreux services sur cette machine, et les données vont également exiger beaucoup d'espace. Nous conseillerons donc que la machine serveur soit une machine bien équipée, surtout en termes de taille mémoire et de capacité disque. Les disques durs devront dans toute la mesure du possible être des modèles SCSI.
- Plus grande facilité d'administration. Le système de droits d'accès aux fichiers Linux est nettement plus simple que celui de Window$. (Certains lui reprochent même d'être trop simple !). Configurer les différents services (ou "*daemons"*) est également une tâche plus simple grâce à l'usage systématique de fichiers de configuration auto-documentés. Les outils de gestion sont extraordinairement puissants (ils ont fait leurs preuves sous Unix depuis fort longtemps) et on peut facilement automatiser toutes sortes de tâches à l'aide de scripts (De nombreux langages sont disponibles, y compris de haut niveau).

Distribution choisie

Linux est un système totalement gratuit, mais à condition de l'assembler soi-même à partir d'une multitude de sources dispersées sur l'internet, ce qui représente une tâche colossale. Partant de ce constat, de petites sociétés commerciales se sont créées qui proposent des ***distributions*** de Linux, c.à.d. des ensembles cohérents de paquetages déjà compilés, testés, et généralement accompagnés de bons manuels expliquant en détail l'installation du système (et parfois aussi l'initiation).

Nous avons arrêté notre choix sur la distribution allemande ***SuSE***, qui est très complète (7 CD + 1 DVD dans la dernière version), s'installe facilement, et est accompagnée de manuels en français totalisant + de 1000 pages de documentation technique. Libre à vous de tester d'autres distributions, telles ***Mandrake***, par exemple (distribution d'origine française), ou encore ***Red Hat*** ou ***Debian***.

Installation

Le processus d'installation est très bien documenté dans le manuel qui accompagne votre distribution. Les deux pages qui suivent vous indiquent une procédure résumée, valable essentiellement pour les machines du centre de formation.

Guide d'installation rapide (Distribution SuSE Linux 7.2 - boot depuis le CD 1)

Démarrage → Attendre → Choix de la langue → Choix du type de clavier : français au départ (le clavier belge sera proposé + loin) – Zone horaire : Europe/Belgique

➔ Nouvelle installation de Linux

➔ Choix du disque dur : Choisir **Partitionnement manuel – pour experts**

➔ Affichage des partitions → effacer toutes les partitions existantes à partir de /dev/hda2 (y compris la partition "extended") **!!! ATTENTION !!!**
!!! ne pas effacer la partition /dev/hda1 qui contient Window$!!!

Créer ensuite :

- une partition étendue /dev/hda2. Conserver les choix par défaut pour la taille (cette partition occupera donc toute la place restante sur le disque dur, soit env. 12.4 Gb)

Toutes les partitions suivantes seront des partitions logiques (dans la partition étendue ci-dessus) :

- une partition de 256 Mb environ (utiliser l'option +256M pour trouver le cylindre final), à formater avec le type SWAP – pas de point de montage.
- une partition de 15 Mb environ, à formater avec le type ext2 – point de montage : /boot
- une partition de 500 Mb environ, format ext2, point de montage : /home
- une partition de 800 Mb environ, format ext2, point de montage : /var
- une partition de 2 Gb environ, format ext2, point de montage : /opt
- une partition de 2.6 Gb environ, format ext2, point de montage : /
- une partition avec toute la place restante, format ext2, point de montage : /reserve

➔ Noter la désignation physique de la partition racine (ce devrait être /dev/hda10)

➔ **Sélection de logiciels** : Choisir **Système minimal (X11) sans KDE** , cliquer sur "Sélection détaillée", puis sur "Sélection de paquetages déterminés" → Ajouter les paquetages suivants :

- Applications/Editors: joe
- Applications/File : mc
- Applications/Graphics : sketch
- Applications/Internet : mozilla
- Applications/Productivity : so-fr
- Applications/Publishing : texinfo
- Development/languages/Python : python
- Development/Libraries/Python : python-imaging
- Networking/daemons : apache, samba, squid,squidguard
- Networking/utilities : yp-tools
- System environment/Base : quota
- System environment/Daemons : mod-python, nfs-utils
- System environment/Kernel : ipchains

- System environment/Yast : yast2-config-cups, yast2-config-sound
- Utilities/Printing : aps (→ gs_X11 et cups)
- X11/Applications/Graphics : gimp, acroread
- X11/Applications/Internet : netscape
- X11/Kde/Base : kdeadmin, kdebase, kdegraphics, ksupp, kdenetwork, kdeutils, kdemultimedia
- X11/Kde/i18n : kde-i18n, kde-i18n-fr

➔ Suivant → **Configurer l'amorçage du système** : Cliquer sur "Autre configuration de Lilo"
→ Choisir : Installation de Lilo sur C:('MBR' du disque dur)

➔ Choisir un utilisateur type (login utilisateur en minuscules, sans accents ni cédille). Mot de passe : abcde

➔ Choisir un mot de passe pour "root" : abcde (on en choisira un meilleur + tard)

➔ Confirmer l'installation

➔ L'amorçage de votre système est en cours : → OK.

➔ Insérer les autres CD à la demande

➔ Configuration du moniteur : Type quelconque, mais : Fréq. horiz. = 30-86 kHz ; Fréq. vert. = 50-100 Hz

➔ Environnement graphique 1024x768, couleurs 16 bits → Tester → Centrer l'image, la dilater un peu latéralement (→ 70 kHz, 87 Hz) → Cliquer sur "Sauvegarder la configuration".

➔ Voulez-vous voir le protocole d'installation : Non

➔ Congratulations ! etc. : Cliquer sur "Configuration périphériques", bouton "Réseau"
→ Choisir "Assignation statique d'adresses **!! Attention !!**
Pour ce qui suit, demander les informations caractéristiques de votre machine au formateur :
Adresse IP : **192.168.0.xxx** , masque : **255.255.255.0**
Nom d'hôte : "**CFxx**" , nom de domaine "**CFORM**" (en majuscules)
→ Terminer → Enregistrer la configuration réseau

➔ Terminer l'installation

➔ Premier démarrage de Linux, directement en mode graphique (Login graphique avec KDM)

➔ Votre première fenêtre KDE apparaît.

"Have a lot of fun !"

Quelques informations en vrac concernant l'installation :

Partitions : Si l'espace disque est limité (installation sur un poste de travail), vous pouvez vous contenter d'une partition de swap (64 à 128 Mb) et d'une partition racine (/). Par contre, si vous installez un serveur, vous devez forcément disposer de plus d'espace. Prévoyez notamment au moins une partition séparée pour /home (c'est cette partition qui contiendra toutes les données des utilisateurs). Il est bon de prévoir aussi une partition indépendante pour /var (où s'accumuleront peu à peu les fichiers de *log*). A titre informatif, essayez de prévoir 1 à 2 Gb pour / , 1 à 1,5 Gb pour /usr, 1 Gb pour /var (davantage si vous installez un serveur de bases de données tel que MySQL), quelques Gb pour /home (au moins 10Mb par utilisateur). Si vous disposez de beaucoup de place, n'hésitez pas à séparer encore d'autres partitions (/opt, par exemple, pour les grosses applications annexes, les encyclopédies, les applications destinées à Window$, etc.)

Paquetages à installer : Sur un gros système serveur (avec de la place en suffisance), le plus simple est peut-être d'installer l'ensemble de paquetages proposé comme "standard" par SuSE. Nous préférons pour notre part choisir une installation minimale (avec système graphique X11), à laquelle nous ajoutons un certain nombre de paquetages (plus tous ceux dont ils dépendent) :

- Interface KDE2 (y compris programmes graphiques et utilitaires)
- samba (série n) : c'est le logiciel qui permettra l'interfaçage avec le monde Window$
- squid (série n) : c'est le serveur proxy pour contrôler l'accès internet.
- squidguard (série n) : c'est le complément de squid pour censurer les sites "scabreux"
- apache (série n) : ce sera votre serveur Web d'intranet.
- ipchains : (série sec) : nécessaire si vous voulez utiliser la technique d'IP-masquerade.
- quota (série ap) : permet de gérer les quotas d'espace disque (indispensable dans une école !!!)
- ypclient (sur les clients), ypserv (sur le serveur) : paquetages indispensables si votre réseau comporte des postes de travail fonctionnant sous Linux – inutiles si tous vos postes de travail fonctionnent sous Window$.
- Netscape (série xap) : faut-il le présenter ?
- StarOffice 5.2 (série pay) : Voir plus loin pour l'installation réseau.
- MySQL (gestionnaire de base de données), Python (si vous souhaitez découvrir ce merveilleux langage de programmation) Forte for Java (si vous voulez programmer en java), gcc, gpp et Kdevelop (si vous voulez programmer en C++), etc.

Carte réseau : Linux peut fonctionner avec la plupart des cartes réseau disponibles.

Le logiciel d'installation arrive généralement à les reconnaître et les configurer automatiquement. Sinon, votre principal problème sera d'identifier correctement ce dont vous disposez (les cartes bon marché sont souvent vendues dans le commerce sous des appellations "exotiques"). Les "clones" les plus fréquemment rencontrés, notamment sous la marque DLink, sont VIA Rhine et DEC "Tulip"

Si vous avez une machine très récente, il se peut que le pilote correspondant à votre carte réseau n'ait pas encore été incorporé dans votre distribution de Linux. Consulter votre Gourou local, ou le site Internet officiel de votre distribution, ou encore les HOWTOs et les forums de discussion.

Nom de la machine - Domaine - Adresse IP : Nous vous conseillons vivement de choisir une convention pour la casse des caractères, et de vous y tenir par la suite. Nous-mêmes utilisons exclusivement des majuscules (8 caractères max. - pas d'espaces !) pour le nom de domaine et les noms de machines.

La plage d'adresses IP choisie doit être soit 192.168.0.1 => 192.168.0.253 , soit 192.168.1.1 => 192.168.1.253 (Masque = 255.255.255.0 - réseau de classe C).

Brève incursion dans le monde de Linux - Quelques commandes importantes

L'une des premières opérations à effectuer, lorsqu'on commence à utiliser un système Linux, est de définir nouvel utilisateur. Vous pouvez effectuer cette opération à partir des utilitaires **yast** ou **yast2** (utilitaires spécifiques de la distribution *SuSE*), ou bien lancer la commande **useradd**.

Lors de l'installation de Linux, il vous a été demandé de choisir un mot de passe pour l'administrateur du système. Cet utilisateur particulier, toujours désigné par le nom de **root**, possède absolument tous les droits sur le système. Vous pourriez en principe toujours utiliser votre système Linux en vous y connectant en tant que "root", mais cela est déconseillé. Lorsque vous êtes connecté ainsi, vous pouvez tout faire, y compris beaucoup de gaffes. Dans la mesure du possible, il ne faut se connecter en tant qu'administrateur que pour effectuer de véritables tâches d'administration. Le reste du temps, il vaut mieux se connecter en utilisateur plus ordinaire (c.à.d. avec des droits limités), même si on est le propriétaire de la machine. De cette manière, on se protège soi-même automatiquement contre un grand nombre de fausses manoeuvres. Il est bien évident aussi que dans le cas du serveur réseau que nous voulons mettre en place, il nous faudra créer des comptes utilisateur - avec des droits assez limités - pour les élèves qui s'y connecteront.

Créez donc un nouvel utilisateur, et reconnectez-vous en utilisant son nom et son mot de passe.

Veuillez vous référer aux exercices effectués en cours de formation pour tout ce qui concerne la découverte de l'interface graphique *KDE* et de l'explorateur *Konqueror*.

Il est important de bien comprendre que cette interface graphique n'est qu'une "garniture" facultative, très agréable à utiliser pour accomplir certaines tâches d'une manière similaire à ce que l'on est habitué à trouver dans les mondes *M@c* ou *Window$*, mais qui n'est absolument pas indispensable au fonctionnement de Linux (Vous devrez d'ailleurs vous faire à l'idée que l'on n'est vraiment efficace sous Linux que lorsque l'on maîtrise la ligne de commande "en mode texte").

Même lorsque vous travaillez avec une interface graphique telle que *KDE* ou *WindowMaker*, vous pouvez à tout moment "ouvrir une fenêtre de terminal" pour introduire des commandes texte.

Modification et copie de fichiers texte

Lorsque vous aurez un peu exploré l'interface graphique *KDE* et lancé l'un ou l'autre petit programme, vous allez réaliser vos premiers exercices de gestion d'un système Linux. Pour commencer, ouvrez l'explorateur *Konqueror* et habituez-vous à "voyager" dans l'arborescence des fichiers et répertoires. (Notez au passage qu'il n'est fait mention nulle part des disques A:, C:, D:, etc. comme sous DOS/Window$: veuillez s.v.p. vous référer aux explications données en cours de formation concernant le "montage" des ressources sous Linux).

Activez l'option "Affichage des détails" de l'explorateur, afin de visualiser en permanence tous les attributs de chaque fichier. Si vous cliquez sur un fichier contenant un simple texte, en principe il sera reconnu comme tel par *konqueror* et un éditeur de texte s'ouvrira automatiquement pour vous en montrer le contenu.

Essayez par exemple de repérer et de visualiser le contenu du fichier /etc/motd (ce petit fichier contient le message de bienvenue du système). Êtes-vous bien connecté en utilisateur ordinaire (comme indiqué au paragraphe précédent) ? Si oui, effectuez une modification quelconque dans ce fichier, puis tâchez de sauvegarder le fichier modifié. Vous n'y parviendrez pas, et ainsi vous ferez connaissance pour la première fois avec le système de gestion des droits d'accès de Linux : jusqu'à présent, vous avez en effet l'autorisation de lire ce fichier, mais non d'y changer quoi que ce soit.

Tâchez maintenant de réaliser une copie de ce fichier. Pour ce faire, vous pouvez par exemple ouvrir une seconde fenêtre dans *Konqueror* et effectuer un glisser-lâcher de l'une à l'autre. En procédant ainsi, la copie que vous effectuez n'a plus les mêmes attributs que l'original. Par défaut, une copie est un nouveau fichier dont la propriété revient à celui qui le crée, c.à.d. l'utilisateur qui effectue la copie. Ouvrez et éditez la copie : cette fois vous pouvez l'enregistrer sans problème.

Gestion des droits d'accès aux fichiers sous Linux

Comme tout bon système d'exploitation multi-utilisateurs, Linux comporte des dispositifs efficaces pour le contrôle de l'accès aux fichiers. Si vous avez l'habitude du système de fichiers FAT16/FAT32 de DOS/Windows, cela constituera un changement important pour vous.

Lorsque l'on consulte un répertoire Linux en demandant l'affichage de tous les détails, on constate que chaque fichier est accompagné d'un ensemble d'attributs :

- l'identifiant du propriétaire du fichier (en général, l'utilisateur qui a créé ce fichier)
- l'identifiant d'un groupe d'utilisateurs qui peut posséder des droits sur ce fichier (groupe propriétaire)
- 3 groupes de 3 bits définissant les droits d'accès de chacun :

Droits du propriétaire	*Droits du groupe prop.*	*Droits du reste du monde*
r w x	r w x	r w x

S'il sont activés, le bit **r** indique un droit de lecture, le bit **w** un droit d'écriture et le bit **x** un droit d'exécution.

Par exemple, si un fichier est accompagné de la séquence **rwx r-x ---** , cela signifie que son propriétaire a tous les droits, que son groupe propriétaire peut lire et exécuter le fichier (en supposant qu'il soit exécutable), et que le reste du monde n'y a pas accès du tout (aucun droit).

En réalité, ces attributs sont de véritables bits, évidemment regroupés dans des octets par la machine. On peut donc représenter l'ensemble des droits associés à un fichier par un groupe de 3 octets : un octet pour les droits du propriétaire, un octet pour les droits du groupe, un octet pour les droits du reste du monde. Il est commode de représenter ces octets sous forme décimale, mais il faut pour cela faire un petit rappel d'arithmétique concernant la conversion binaire => décimal :

Binaire *r w x*	*Décimal*	*Droits correspondants*
0 0 0	0	aucun
0 0 1	1	exécution
0 1 0	2	écriture
0 1 1	3	écriture & exécution
1 0 0	4	lecture
1 0 1	5	lecture & exécution
1 1 0	6	lecture & écriture
1 1 1	7	lecture, écriture, exécution

Avec un peu d'habitude, on peut ainsi arriver à décrire l'ensemble des droits associés à un fichier d'une manière extrêmement concise. Par exemple :
- 777 signifie que tout le monde a tous les droits (lecture, écriture, exécution) sur ce fichier
- 750 signifie que le propriétaire a tous les droits, le groupe peut lire et exécuter, les autres n'ont aucun droit
- 644 indique un fichier que tous peuvent lire, mais où le propriétaire seul est autorisé à écrire. Etc.

La commandes en ligne **chmod** permet de modifier les droits d'un fichier - ou mieux encore d'un ensemble de fichiers - d'une manière très efficace. Exemple :

`chmod 750 *` => tous les fichiers du répertoire courant reçoivent les attributs : **`rwx r-x ---`**
`chmod 754 ar*` => tous les fichiers commençant par ar reçoivent les attributs : **`rwx r-x r--`**

La commande **chown** permet de changer le propriétaire. Exemple :

`chown louis *` => **louis** devient le propriétaire de tous les fichiers du répertoire courant.

D'une manière analogue, la commande **chgrp** permet de changer le groupe. Exemple :

chgrp profs *.doc => tous les fichiers dont le nom se termine par **.doc** sont associés au groupe **profs**.

Copie de fichiers sans modification de leurs attributs

En tant qu'administrateur, il vous arrivera fréquemment d'avoir à réaliser des copies de fichiers à l'identique, c.à.d. De telle façon que les copies conservent exactement les mêmes attributs que les originaux (pour des *backups*, par exemple).

Utilisez alors la commande en ligne **cp** , avec l'option **-au** :

```
cp -au <source> <destination>
```

Montage de répertoires

Sous Linux, tous les répertoires sont rassemblés dans une arborescence unique (y compris les répertoires de disques différents, y compris les répertoires de systèmes d'exploitation différents (Window$ par exemple), y compris les répertoires situés sur des machines distantes).

Il ne nous est pas possible de détailler ce mécanisme dans le contexte de ces notes résumées. Veuillez vous référer aux exercices effectués durant la formation et/ou consulter la littérature spécialisée.

A titre d'exercice, il vous sera notamment demandé d'effectuer un montage distant par NFS, c.à.d. l'incorporation à votre propre arborescence de répertoires, d'un répertoire de fichiers situé sur une autre machine, via le réseau. (NFS = Network File System)

Fichiers de configuration

Les principaux fichiers de configuration de Linux se trouvent traditionnellement dans **/etc**
C'est dans ce répertoire /etc que vous devrez trouver, entre autres :

rc.config : fichier utilisé par les logiciels **yast** et **suseconfig** de la distribution SuSE pour centraliser l'administration du système. Veuillez consulter la documentation SuSE pour l'utilisation de ce fichier.

XF86Config : contient la configuration du serveur X, c.à.d. le système vidéo graphique (y compris la configuration de la souris et celle du clavier pour le mode graphique)

passwd : contient la liste des utilisateurs du système et leurs paramètres caractéristiques

shadow : contient les cryptages des mots de passe utilisateurs

group : contient la liste des groupes d'utilisateurs (classes, par ex.)

gshadow : contient les cryptages des mots de passe éventuels des groupes

exports : contient la liste des répertoires partagés sous **NFS** (voir plus loin)

smb.conf : fichier de configuration de **Samba** (interface Linux/Window$: voir plus loin)

fstab : contient la liste des partitions et les modes de montage correspondants. Ce fichier ne doit être modifié qu'en bonne connaissance de cause (voir exemple ci-dessopus)

hosts : Système de résolution de noms minimal. Il peut être utile d'y indiquer les noms et adresses IP de tous les postes connectés au réseau, surtout s'il y a des postes clients fonctionnant sous Linux et si l'on n'utilise aucun autre système de résolution de noms (DNS). Les clients Window$ utiliseront le serveur Wins inclus dans Samba.

squid.conf : fichier de configuration du serveur proxy **Squid2** (voir plus loin)

Fichier /etc/fstab

Ce fichier est indispensable au démarrage correct de Linux. Ne le modifiez donc qu'en connaissance de cause ! Il contient la description des partitions qui doivent être "montées" dans l'arborescence de fichiers au démarrage de la machine. Ces partitions peuvent être des partitions locales ou même distantes (accessibles sur d'autres machines via le réseau).

Exemple :

```
/dev/hda1            /dos           vfat      umask=0000            0  0

/dev/hdb2            swap           swap      defaults              0  0
/dev/hdb3            /              ext2      defaults              1  1
/dev/hdb4            /home          ext2      defaults,usrquota     1  2

/dev/hdd             /cdrom         iso9660   ro,noauto,user,exec   0  0
/dev/hdc4            /zip           auto      noauto,user,exec      0  0
/dev/fd0             /floppy        auto      noauto,user           0  0
Boss:/opt/soffice    /mnt/soffice   nfs       defaults              0  0

none                 /proc          proc      defaults              0  0
none                 /dev/pts       devpts    defaults              0  0
```

Remarques :

- La première partition correspond au premier disque dur et est réservée à MSDOS(Window$) sur cette machine. L'indication umask =0000 signifie que l'accès à ces fichiers depuis Linux aura lieu avec des droits correspondant à l'inverse binaire de ce masque, soit 7777 (= tous les droits).
- Un système de quotas disque est instauré pour la partition /dev/hdb4 (partition /home). Ces quotas seront définis pour chaque utilisateur (voir + loin).
- dev/hdc4 correspond à un lecteur Zip interne. Ce lecteur acceptera des disquettes formatées DOS ou Linux sans rechigner, grâce à l'indication auto utilisée en lieu et place du système de fichiers. Le lecteur de disquettes (dev/fd0 = /floppy) fonctionnera d'une manière analogue.
- La partition Boss:/opt/soffice est montée à partir du réseau, par NFS. Il s'agit en fait d'une partition située sur un serveur distant (et qui contient la suite bureautique Star office).

Fichier /etc/hosts

Dans un réseau fonctionnant sous le protocole TCP/IP, il faut qu'un système soit mis en place pour la résolution des noms des machines locales en adresses IP (pour la résolution des noms des machines distantes, accessibles via l'internet, il faudra faire appel à un serveur DNS, mais dans ce cas on utilisera généralement l'un des serveurs proposés par le fournisseur d'accès). Pour un petit réseau scolaire composé de postes Window$, on fera appel au service Wins inclus dans le logiciel Samba (voir + loin). Pour les machines fonctionnant sous Linux (dont le serveur lui-même), le plus simple est de doter chacune d'elles d'un fichier /etc/hosts contenant la liste des noms de toutes les machines locales en regard de leur adresse IP (voir exemple ci-dessous). L'inconvénient de cette méthode ultra-simple est qu'il faut mettre à jour le fichier /etc/hosts de chaque machine chaque fois que l'on modifie la structure du réseau (ajout de machines, changement de leurs noms, etc.). En contexte scolaire, nous pouvons supposer que cela n'arrivera pas trop souvent.

Exemple de fichier /etc/hosts :

```
127.0.0.1       localhost
192.168.0.100   BOSS.CFORM      BOSS
192.168.0.102   PC2.CFORM       PC2
192.168.0.103   PC3.CFORM       PC3
192.168.0.104   PC4.CFORM       PC4
```

Fichier /etc/exports

S'il existe plusieurs machines Linux sur le réseau, il faut mettre en place quelques services pour que ces machines puissent mieux communiquer. Il faut par exemple lancer sur chaque machine un serveur NFS (Network File System) pour que ces machines puissent s'échanger des fichiers. Il faut également préciser sur chaque machine les répertoires que l'on accepte de rendre accessibles aux autres, et à quelles conditions (notion équivalente aux "partages" sous Window$). Si vous avez installé Linux suivant nos indications, le serveur NFS est déjà en place. On peut le faire activer automatiquement au démarrage de la machine à l'aide d'une simple option dans /etc/rc.config

Les "partages" eux-mêmes sont définis dans le fichier /etc/exports, dont les lignes doivent être du type ci-dessous (nom du répertoire à partager + nom des machines autorisées + droits concédés) :

```
/dos2/graphics/bmp        PC4(rw)  PC5(ro)  PC6(rw)
/home/public              *.CFORM(rw)
```

Le nom de la ou des machines est indispensable (sinon transferts très lents, messages d'erreur...) Après chaque modification dans /etc/exports, relancer le serveur NFS par la commande :

```
rcnfsserver restart
```

ou bien :

```
/etc/init.d/nfs restart
```

Attention : pour que tout ceci fonctionne, il faut absolument que la résolution des noms soit assurée (voir paragraphe précédent). Si ce n'est pas le cas, vous pouvez aussi remplacer les noms de machines par leurs adresses IP.

Documentation interne (pages "man" et "info")

Comme les systèmes Unix dont il dérive, Linux dispose d'un système d'aide en ligne toujours accessible. Il suffit d'entrer la commande **man** (pour "manuel") suivie du nom du dispositif ou de la commande pour laquelle on désire obtenir des explications. Exemple : `man exports`

Ce système de documentation hérité d'Unix est maintenant complété d'un autre système plus performant qui intègre des liens hypertexte. S'il est installé sur votre machine, vous pouvez appeler ce système à l'aide de la commande **info**. Essayez par exemple : **info exports**

Configuration d'un serveur Samba

Samba est un extraordinaire logiciel permettant de contrôler un parc d'ordinateurs reliés les uns aux autres par l'intermédiaire du protocole SMB de Micro$oft, c.à.d. des machines fonctionnant sous Window$ et qui "croient" qu'elles sont contrôlées par un serveur Window$ NT. Samba effectue donc une émulation de NT, totalement transparente pour les machines clientes. Celles-ci "voient" le réseau comme un domaine Window$ NT, avec cependant quelques limitations sans importance pour nous (et dont nous ne parlerons pas ici), et quelques avantages très appréciables (stabilité, vitesse accrue, insensibilité aux virus, gratuité ...)
Si vous avez suivi nos indications, Samba est déjà installé sur notre serveur. Il reste cependant à le configurer. Si vous avez "tâté" un peu de Window$ NT auparavant, vous découvrirez à cette occasion la grande différence "philosophique" qui séparant la gestion d'un système de type UNIX d'un système de type Window$. Ici, point de fenêtres avec des cases à cocher, ni d'onglets ni de "boutons radio". Tout se trouve dans un unique fichier texte.
Au début, vous aurez peut-être l'impression de "régresser" vers un monde informatique antédiluvien. A l'usage journalier, par contre, vous bénirez les auteurs de ce système de n'avoir pas succombé aux charmes des clickodromes et autres interfaces graphiques de style M@c ou Window$, parce qu'à l'usage, la ligne de commande et le fichier de configuration en mode texte se révéleront bien plus souples et bien plus faciles à contrôler.

Mais trève de discours. Ce que vous devez examiner et adapter à votre situation particulière se trouve dans le fichier /etc/smb.conf . Nous fournissons en annexe un fichier /etc/smb.conf typique, avec des commentaires intégrés traduits en français. Il ne nous est pas possible de vous proposer davantage dans le cadre de ces notes résumées. Pour aller plus loin, il vous faudra consulter l'excellent ouvrage sur Samba cité dans la bibliographie.

Configuration de Samba en tant que "client" sur un poste de travail Linux

(pour réaliser un interfaçage simple avec les postes Window$ du réseau.)

Dans cet exercice, on configure un poste de travail Linux de telle manière qu'il puisse être considéré par les postes de travail Window$ du même réseau comme si c'était l'un des leurs. Dans cette configuration, la machine Linux joue seulement le rôle d'un serveur de fichiers dans une structure de type "poste à poste". Pour obtenir ce résultat, il suffit de définir un certain nombre de paramètres dans le fichier de configuration /etc/smb.conf, puis de (re)démarrer Samba. Une version "dépouillée" du fichier /etc/smb.conf a été préparée à votre intention et elle est accessible sur le serveur réseau (voir ci-dessous). Entre autres choses, ce fichier contient la liste des répertoires que l'on souhaite partager au départ de la machine Linux.

- Se connecter en tant que "root". Créer deux répertoires : /home/public et /home/graphics
- Placer un ou plusieurs fichiers quelconques dans ces répertoires :

```
cd /home
mkdir public
chmod 777 public                    (→"full access" pour tout le monde)
mkdir graphics
chmod 755 graphics                  (→accès en lecture pour tout le monde)
ls -l                               (→contrôle de ce que l'on vient de faire)
cp /dos/logo.sys  /home/graphics    (copie d'une image dans /home/graphics)
cd /home/graphics
ls -l                               (→contrôle)
```

- Installer un fichier de configuration simplifié pour Samba.
  Vous pouvez trouver ce fichier dans les annexes accompagnant ces notes (fichier smbclient.conf). Les participant à la formation organisée à la Maison diocésaine de Liège peuvent aussi récupérer ce fichier par NFS sur le serveur réseau :

```
mount -t nfs 192.168.0.100:/home/archives  /mnt
cd /mnt
ls -l                                (pour voir les fichiers disponibles)
cp smbclient.conf  /etc/smb.conf  (on copie le fichier dans le répertoire /etc
cd /                                  en veillant à le renommer smb.conf)
umount /mnt
```

- Analyser le fichier /etc/smb.conf ainsi installé. Pour ce faire, vous pouvez par exemple entrer la commande : **less /etc/smb.conf** (visualisation directe en mode texte).
  Note : Si vous avez activé l'interface graphique, vous pouvez passer du mode "texte" au mode graphique à volonté avec les combinaisons de touches :
  CTRL-ALT-F7 pour repasser à la console graphique
  CTRL-ALT-F2 ou CTRL-ALT-F3 ou CTRL-ALT-F4, etc. pour activer une console texte, et passer ainsi de l'une à l'autre à volonté.

Dans /etc/smb.conf, on notera surtout l'utilisation de l'option :
security = share, ainsi que les options empêchant Samba de prendre le contrôle du domaine.

- **Démarrer le serveur Samba :**
  /etc/init.d/smb start # (ou bien : /etc/init.d/smb restart)
  (Si vous voulez que Samba soit lancé automatiquement au démarrage de l'ordinateur, lancez YAST → Administration du système → Modifier le fichier de configuration → START_SMB = Yes)
- Depuis une machine Window$ quelconque reliée au réseau, ouvrir "voisinage réseau" : la machine Linux est accessible, avec notamment ses deux partages "Public" et "Graphics".

Vous devriez pouvoir accéder à l'image stockée dans "Graphics", et copier des fichiers quelconques dans "Public". Effectuez maintenant diverses modifications des droits d'accès à ces répertoires, pour voir comment les choses se présentent pour la machine Window$ qui cherche à utiliser ces partages.

Vous pouvez également expérimenter diverses modifications dans /etc/smb.conf au sujet de ces partages. Essayez par exemple les options browsable = no et writable = no .

Depuis une machine Window$, essayer d'accéder au lecteur CD de la machine Linux. N'oubliez pas que cette ressource "amovible" doit d'abord être "montée".

Rappel : commande à utiliser pour effectuer le montage d'un CD situé physiquement en /dev/hdd :

```
mount -t  iso9660  /dev/hdd   /cdrom
```

Utilisation de Samba comme contrôleur principal de domaine

Dans les pages qui suivent, nous allons maintenant nous attacher à décrire l'installation d'un véritable serveur principal, qui prenne en charge l'authentification des utilisateurs qui se connectent au réseau et leur attribue un environnement de travail personnalisé.

Configuration des machines clientes (Win95 ou Win98)

Pour qu'il puisse s'intégrer à un réseau, un ordinateur doit être pourvu des dispositifs suivants :

- une carte réseau
- un protocole de communication
- un logiciel client

Tous ces dispositifs doivent être "installés". Sous Window$ 95/98, on utilise pour cela le panneau de configuration => icône Réseau. On obtient une fenêtre avec 3 onglets :

Onglet "Configuration" :

La liste affichée devrait contenir les 4 éléments suivants (sinon il faut les installer) :

- un client pour les réseaux Micro$oft
- un pilote spécifique pour la carte réseau effectivement présente dans la machine
- le protocole TCP/IP activé pour cette carte. Dans "propriétés" : définir l'adresse IP, le masque de sous-réseau, et activer la résolution WINS (fournir l'adresse IP du serveur).
- l'activation du partage des fichiers et imprimantes pour les réseaux Micro$oft.

La rubrique "Ouverture de session réseau principale" devrait contenir : "Client pour les réseaux Micro$oft"

Onglet "Identification" :

Veiller surtout à ce que le nom du domaine soit correct.

Onglet "Contrôle d'accès" :

Vous devez impérativement opter pour le **contrôle d'accès au niveau ressource**. En effet : dans l'état actuel des choses, il n'est pas encore possible de gérer la fonctionnalité "Contrôle d'accès au niveau utilisateur" des domaines Window$ NT via Samba sous Linux.
A notre humble avis, cette fonctionnalité complexe ne présente d'ailleurs aucun intérêt dans le contexte d'un réseau scolaire.

Autres actions à effectuer sur les postes clients Window$:

Installer **Poledit** (éditeur de stratégie système) sur toutes les machines. Pour l'installer, il faut ouvrir le panneau de configuration => Ajout/Suppression de programmes => Installation de Windows => Disquette fournie => *parcourir le CD de Win95* => \admin\apptools\poledit (ou bien \tools\Reskit\Netadmin\Poledit dans le cas de Win98) -> installer.

Lancer Poledit => ouvrir la base de registres =>

Utilisateur local : *laisser toutes les cases à cocher en blanc*
Ordinateur local :

- Réseau :
  - Contrôle d'accès => ***décocher*** *la case* "contrôle d'accès au niveau utilisateur"
    (*pour que le contrôle d'accès se fasse au niveau **ressource***.)

  Pour les options suivantes, cochez les cases indiquées ci-après :
  - Ouverture de session =>
    - * Bannière d'ouverture de session => *insérer un texte à votre convenance*

        - * Nécessite une validation par réseau pour l'accès Windows
    - Mettre à jour =>
        - * Mise à jour distante => *choisir le mode mise à jour manuel : il faut alors préciser l'emplacement du fichier config.pol (ex : \\Boss\Netlogon\Config.pol).*
    - Mots de passe =>
        - * Masquer les mots de passe de partage par des astérisques
        - * Désactiver la mise en antémémoire des mots de passe
    - Client Microsoft pour réseaux Windows =>
        - * Ouverture de session sur Windows NT => *indiquer le nom du domaine*
            - * Désactiver la mise en antémémoire du mot de passe de domaine
        - * Nom du domaine => *indiquer le nom du domaine*
    - Accès réseau à distance =>
        - * Désactiver la réception d'appels
- Système :
    - * Activer les profils d'utilisateurs

Ne pas oublier qu'il faudra rebooter (il s'agit de Window$!)

Toujours à l'aide de Poledit, créer un fichier Config.pol (Ce fichier devra ensuite être installé sur le serveur, dans /home/netlogon. Voir plus loin).

Ordinateur par défaut : *laisser toutes les cases à cocher en gris.*
Utilisateur par défaut :
- Panneau de configuration : *cocher toutes les cases (= restrictions)*
- Bureau :
    - * Papier peint => *\\Boss\homes\background* ("Boss" = nom du serveur)
- Réseau : *cocher toutes les cases (= restrictions)*
- Environnement :
    - Dossiers personnalisés :
        - * Dossier Programmes personnalisé => *\\Boss\homes\apps*
        - * Icônes personnalisées du bureau => *\\Boss\homes\bureau*
        - * Masquer les sous-dossiers du menu Démarrer
        - * Menu Démarrage personnalisé => *\\Boss\homes\smenu*
        - * Dossier Démarrage personnalisé => *\\Boss\homes\start*
    - Restrictions :
        - * Supprimer la commande 'Exécuter'
        - * Supprimer les dossiers de 'Paramètres' dans le menu Démarrer
        - * Supprimer la barre des tâches de 'Paramètres' dans le menu Démarrer
        - * Supprimer la commande 'Rechercher'
        - * Ne pas enregistrer les paramètres à la sortie
- Système :
    - Restrictions => * Désactiver les outils d'édition de la base de registres

Si l'on décoche la case "Dossier Démarrage personnalisé", l'utilisateur aura accès aux raccourcis présents dans *C:\windows\menu démarrer* (généralement explorateur & prompt MSDOS).

Autres utilisateurs :

En effectuant le travail ci-dessus, vous établissez une stratégie système qui sera la même pour tous les élèves (tout utilisateur pour lequel il n'existe pas une stratégie système spécifique reçoit automatiquement celle de "l'utilisateur par défaut"). Vous pourrez bien évidemment définir aussi quelques autres stratégies systèmes personnalisées au nom de divers utilisateurs particuliers (professeurs, par exemple). Veillez au moins à définir une stratégie système pour l'utilisateur **root** dans laquelle vous n'activerez aucune restriction (toutes les cases à cocher seront laissées en blanc).

Configuration de Samba comme "Patron"du domaine

Dans cet exercice, on configure une machine tournant sous Linux de telle sorte qu'elle puisse assumer les tâches d'un véritable serveur : contrôle de l'accès aux machines Window$ par authentification des utilisateurs (gestion de leurs mots de passe), gestion de l'environnement de travail octroyé aux groupes d'utilisateurs (bureau, icônes, applications, répertoires personnels), etc.

Attention : pour éviter les conflits avec les autres machines du réseau, attendez le feu vert de votre instructeur avant de redémarrer Samba lorsque vous aurez tout configuré.

Commençons donc par arrêter Samba au cas où il serait déjà actif :

```
/etc/init.d/smb  stop
```

Un fichier de configuration type (smbserver.conf) a été préparé à votre intention. Vous le trouverez en annexe de ces notes, ou sur le serveur réseau si vous êtes participant d'une formation. Dans ce denier cas, effectuez un montage NFS pour le rapatrier :

```
mount  -t  nfs  192.168.0.100:/home/archives  /mnt
cd  /mnt
ls  -l                                        (→ liste des fichiers disponibles)
cp  smbserver.conf  /etc/smb.conf             (→ copier et renommer le fichier)
cd  /
umount  /mnt
```

Mise en place des comptes utilisateurs

Analysez ce fichier */etc/smb.conf* . Il est cette fois assez similaire à la version commentée qui est reproduite à la fin des présentes notes (les commentaires en moins). Notez l'utilisation de l'option security = user, ainsi que les paramètres définissant Samba comme "grand patron" du domaine.

Il vous faut à présent accomplir un certain nombre de tâches sur votre nouveau serveur. Comme déjà indiqué par ailleurs, une bonne partie de ces tâches pourront être automatisées par l'emploi d'un script qui vous sera fourni plus loin (sans cela, l'installation de comptes pour tous les élèves d'une école deviendrait un vrai cauchemar !). Dans l'exercice présent, nous allons cependant effectuer pas à pas la mise en place complète d'un compte utilisateur, "à la main", de manière à nous familiariser avec tous les mécanismes sur lesquels nous serons peut-être amenés à effectuer des corrections (en tant qu'administrateur compétent).

Nous allons donc mettre en place tout ce qu'il faut pour qu'un utilisateur nommé "jules", faisant partie d'un groupe d'élèves nommé "CLASSE6" ait son compte personnel sur le serveur.

A) Création du groupe-classe et de son "chef"

Vous pourriez utiliser Yast : → Administration système → Gestion des groupes → Gestion des utilisateurs, mais pour mieux comprendre le mécanisme, effectuez plutôt les opérations suivantes :

```
groupadd  CLASSE6                (création d'un groupe CLASSE6)
cd  /home
mkdir  CLASSE6                   (création d'un répertoire de même nom)
```

Il sera bon de disposer plus tard d'un utilisateur possédant des droits d'accès étendus sur les répertoires contenant les paramètres d'environnement du groupe (icônes du bureau, programmes du menu démarrer, etc.). Par convention, et pour nous faciliter la vie plus tard, le nom de ce "chef" sera tout simplement identique au nom du groupe lui-même, mais nous l'encoderons en caractères minuscules pour le distinguer du groupe lui-même. Cet utilisateur fait bien évidemment partie du groupe, et son répertoire personnel est celui que nous venons de créer. Ajoutons cet utilisateur :

```
useradd  -g  CLASSE6  -d  /home/CLASSE6  classe6
chown  classe6  CLASSE6      (le propriétaire du répertoire CLASSE6 est classe6)
chgrp  CLASSE6  CLASSE6      (le groupe associé à ce répertoire est CLASSE6)
chmod  755  CLASSE6          (définition des droits d'accès au répertoire)
ls  -l                       (contrôle de ce qui a été fait)
```

Ainsi nous avons créé un groupe CLASSE6, un utilisateur classe6, et un répertoire personnel pour cet utilisateur : /home/CLASSE6. Dans ce répertoire de groupe (la classe, en l'occurrence), nous créerons ensuite des répertoires personnels pour chaque élève. Avant cela, nous allons définir maintenant quatre sous-répertoires pour la personnalisation du bureau Window$, les applications autorisées, le menu démarrer, les icônes etc. qui seront attribués à ce groupe :

```
cd  CLASSE6
mkdir  apps
mkdir  bureau
mkdir  smenu
mkdir  start
chown  classe6  *       (l'utilisateur classe6 est propriétaire de ces rép.)
chgrp  CLASSE6  *       (le groupe associé est CLASSE6)
chmod  755  *
```

L'utilisateur classe6, propriétaire du répertoire /home/CLASSE6 sera seul autorisé à écrire (et à effacer) dans ces répertoires. Les autres utilisateurs (les élèves) pourront accéder à ces répertoires, mais sans pouvoir ni y écrire ni y effacer quoi que ce soit.

B) Création d'un compte utilisateur élève

Avant d'aller plus loin, nous allons encore définir un autre groupe, de professeurs cette fois, lesquels disposeront de droits d'accès sur les répertoires personnels des élèves :

```
groupadd  profs                          (création d'un groupe profs)
```

Pour désigner les membres de ce groupe, nous éditerons + loin le fichier /etc/group.

Créons à présent notre premier utilisateur élève, en l'occurrence un certain "jules" :

```
mkdir  jules
useradd  -g  CLASSE6  -d  /home/CLASSE6/jules  jules
```

Explication : cet élève fait partie du groupe CLASSE6, et son répertoire personnel (privé) est un sous-répertoire du répertoire /home/CLASSE6. Continuons :

```
chown  jules  jules
chgrp  profs  jules
chmod  2770  jules
```

Explications : Après création du répertoire, on définit son propriétaire : **jules**, puis on l'associe au groupe **profs**, de manière à ce que les membres de ce groupe (des professeurs désignés) puissent aussi accéder à ce répertoire. Les permissions sont fixées à 770 (*full access* pour le propriétaire **jules** et pour le groupe **profs**), et le **2** ajouté devant ces attributs positionne le bit GID pour ce répertoire, ce qui signifie que tous les fichiers qui seront créés ensuite dans ce répertoire seront automatiquement associés au même groupe que le répertoire lui-même (en l'occurrence, il s'agit ici du groupe "profs" -> les professeurs auront droit de regard sur ces fichiers).

Comme nous l'avons déclaré plus haut, nous souhaitons que tous les élèves d'un même groupe-classe partagent le même bureau (fond d'écran et icônes), le même menu *Démarrer*, etc. Pour obtenir ce résultat, nous devons faire en sorte que le répertoire personnel de chaque élève paraisse contenir les mêmes sous-répertoires **apps, bureau, smenu** et **start** que nous avons déjà mis en place pour le groupe. Il existe pour ce faire une technique très efficace, qui s'apparente un peu à celle des raccourcis utilisés sous Window$: Nous allons créer dans le répertoire de l'élève quatre ***liens symboliques*** pointant vers les répertoires qui nous intéressent dans le répertoire du groupe :

```
cd  jules
ln  -sn  /home/CLASSE6/apps  apps
ln  -sn  /home/CLASSE6/bureau  bureau
ln  -sn  /home/CLASSE6/smenu  smenu
ln  -sn  /home/CLASSE6/start  start
```

En procédant ainsi, tout se passe ensuite comme si le répertoire personnel de l'élève contenait ces 4 sous-répertoires. Si l'on procède de la même façon pour tous les élèves de la classe, le contenu de

ces répertoires est commun (et situé en réalité dans le répertoire du groupe).

Note : il faudra informer les élèves pour qu'ils n'effacent pas par inadvertance ces liens importants (Comme ces petits fichiers sont situés dans leur répertoire personnel, ils ont en effet le droit de les effacer. Vous pourrez recréer facilement ces liens, voire forcer leur régénération automatique lorsque l'élève se connecte : voir à ce sujet le script backprint.py décrit plus loin)

C) Définition des mots de passe "Linux" & "Window$"

A ce stade de notre travail, il reste à définir les mots de passe d'accès, pour l'utilisateur **jules** bien entendu, mais aussi pour l'utilisateur **classe6** (le "chef" de groupe).

Commençons d'abord par définir les mots de passe valables dans l'univers Linux :

```
passwd  jules          → pour cet exercice, choisissez de préférence "abcde"
passwd  classe6        → idem
```

Il faudra définir séparément les mots de passe requis par l'univers Window$. Cette distinction est nécessaire, parce que l'encryptage des mots de passe est différent dans les deux systèmes Window$ et Linux. On doit donc installer une base de données de mots de passe pour Samba. Il faut d'abord créer la table **smbpasswd** (une seule fois), puis y encoder les mots de passe proprement dits.

Tous les mots de passe doivent être encodés séparément pour les deux systèmes.[1]

Procédure : On commence par créer un sous-répertoire bien protégé dans **/etc** :

```
cd  /etc
mkdir  private
chmod  700  private          → répertoire accessible à root seulement
```

On crée ensuite la base de données proprement dite (fichier **smbpasswd**), en y incorporant des informations extraites de **/etc/passwd** pour les utilisateurs déjà connus du système.

```
cat  /etc/passwd  |  mksmbpasswd.sh  >  /etc/private/smbpasswd
```

(Cette ligne de commande est un bel exemple d'instruction utilisant le mécanisme de redirection des entrées/sorties. Veuillez à ce sujet vous référer aux explications fournies pendant la formation).

Note : Si la commande ci-dessus ne s'exécute pas, c'est que le script **mksmbpasswd.sh** n'est pas actif sur votre machine. Ce script devrait au moins se trouver dans /usr/lib/samba/scripts. Il faut le copier dans un répertoire toujours accessible, tel que /usr/local/bin, et **le rendre exécutable**.

A la suite de cette commande, un fichier **smbpasswd** est créé dans **/etc/private**. Pour des raisons de sécurité évidentes, l'accès à ce fichier doit être strictement réservé à l'administrateur :

```
cd  private
chmod  600  smbpasswd
```

Nous pouvons à présent encoder les mots de passe "Window$"

```
smbpasswd jules        → etc.
smbpasswd classe6      → etc.
smbpasswd              → etc. (pour définir aussi celui de 'root')
```

1 Ils peuvent d'ailleurs être différents, mais cela présente peu d'intérêt dans le contexte scolaire.
(Le script que nous décrivons plus loin créera automatiquement tous ces mots de passe)

D) Création du répertoire "netlogon" pour installer les "logon scripts" et le fichier config.pol

Lors de la connexion d'un poste de travail à un domaine Window$ NT , la machine qui se connecte recherche divers paramètres de démarrage dans un répertoire particulier, situé sur le serveur avec par convention le nom de partage "netlogon". Nous placerons ce répertoire dans /home, et nous lui donnerons également le nom "netlogon" pour nous faciliter la vie. Créons ce répertoire :

```
cd  /home
mkdir  netlogon
chmod  755  netlogon
```

Dans ce répertoire, nous installerons le fichier config.pol (il y a moyen d'en prévoir plusieurs pour des groupes de machines différentes) ainsi que les logon scripts (Tous ces fichiers doivent être préparés sur une machine Window$). Dans cet exercice, vous pouvez profiter de fichiers exemples mis à votre disposition sur le serveur du centre de formation :

```
mount  -t  nfs  192.168.0.100:/home/archives  /mnt
cd  /mnt
cp  config.pol  /home/netlogon
cp  forma.bat  /home/netlogon/CLASSE6.bat          (on renomme le fichier)
```

(Vous verrez plus loin comment créer ces fichiers au départ d'un poste de travail Window$)

E) Installation d'une image de fond pour le bureau

En raison des choix que nous avons effectués lors de la préparation des postes de travail Window$, l'image de fond (obligatoirement un fichier de type bitmap) doit s'appeler *background* et être placée dans le répertoire personnel de l'utilisateur (/home/classe6 /jules, par exemple). Si nous souhaitons que tous les élèves d'une même classe partagent la même image de fond, nous pouvons installer cette image dans le répertoire du groupe (/home/classe6, par exemple), puis créer un lien symbolique pointant vers cette image dans le répertoire personnel de l'élève.

Si le montage NFS utilisé au paragraphe précédent est toujours actif, on peut donc faire :

```
cp  neanderthal.bmp  /home/CLASSE6/background
cd  /home/CLASSE6/jules
ln  -s  /home/CLASSE6/background  background
umount  /mnt
```

F) Démarrage du serveur Samba

Attention !!! Pour la suite de l'exercice, il faudra demander à votre instructeur de déconnecter le serveur réseau habituel (il suffit de débrancher le câble correspondant sur le Hub), parce que c'est désormais votre serveur qui va devenir le "patron". L'idéal sera de relier à votre machine un autre PC fonctionnant sous Window$ pour pouvoir y lancer des essais de connexion.

Procédure :

A l'aide de Yast, redéfinissez l'adresse IP et le nom de votre machine. En effet : vous devez vous rappeler que nous avons configuré nos clients Window$ de telle manière qu'ils recherchent leur contrôleur serveur sous le nom SERVER01, à l'adresse IP 192.168.0.100. Or c'est votre machine qui doit maintenant se présenter ainsi. Pour rappel :

Yast → Administration système → Configurer le réseau → Configuration de base du réseau → Changer l'adresse IP → 192.168.0.100 → ESC → Changer les noms des machines → SERVER01.

Passez en mode console (si ce n'est pas déjà le cas).
Arrêtez les services réseau en entrant la commande :

```
init  2
```

Puis redémarrez ces services à l'aide de la commande :

`init  3` (si vous ne souhaitez pas disposer de l'interface graphique), ou bien :

`init  5` (si vous voulez redémarrer aussi le serveur X11)

Démarrez maintenant Samba par :

`/etc/init.d/smb  start` (En cas d'erreur, il faudra peut-être rebooter).

Note : sur votre serveur effectif (dans votre école), vous souhaiterez certainement que Samba soit automatiquement activé lors du démarrage du serveur lui-même. Pour obtenir ce résultat, il vous suffit encore une fois d'aller faire un petit tour dans Yast :

Yast → Administration système → Modifier le fichier de configuration → START_SMB = yes

G) Installation d'icônes pour le lancement d'applications à partir du bureau

Le principe est simple. Il suffit de placer les raccourcis correspondants dans le sous-répertoire "bureau" du répertoire attribué au groupe (en l'occurrence : /home/CLASSE6/bureau). Il existe plusieurs possibilités pour établir ces raccourcis : on peut les copier depuis un autre compte (celui de root, par exemple), puis en vérifier les attributs (755 de préférence).

On peut aussi les créer en se connectant en tant que "chef" du groupe sur une machine Window$ (dans notre exemple, il faudrait donc se connecter en tant qu'utilisateur **classe6**), puis créer les raccourcis de la manière habituelle sous Window$. Ces raccourcis seront automatiquement mémorisés au bon endroit sur le serveur. On travaillera d'une manière analogue pour placer des raccourcis dans les sous-répertoires **apps, smenu** et **start**.

H) Accès aux répertoires personnels des élèves

Pour que les professeurs aient accès aux répertoires personnels des élèves, il suffit qu'ils soient incorporés au groupe **profs**. Supposons par exemple que l'on veuille octroyer cet accès aux professeurs dont les noms d'utilisateur sont dupont et durand. Il suffit d'éditer le fichier **/etc/group** et d'y rechercher la ligne qui commence par profs. On ajoute les noms dupont et durand à la fin de cette ligne. Exemple :

```
profs:x:107:swinnen,hanon,klich,gillmann,dupont,durand
```

Le script décrit plus loin vous permettra de créer automatiquement des groupes de professeurs titulaires différents pour chaque classe si nécessaire.

Révision d'ensemble
Création de comptes et de répertoires pour tous les élèves de l'école

Nous supposons ici que vous souhaitez centraliser sur votre serveur Linux le système de contrôle d'accès de tous vos élèves, ainsi que leurs données personnelles (travaux en cours, etc.).

Cette façon de procéder comporte en effet un grand nombre d'avantages :

- les données peuvent être protégées dans des répertoires privés accessibles à leur seul propriétaire
- leur regroupement sur une seule machine facilite les sauvegardes périodiques
- on peut facilement mettre en place un système de quotas disque pour éviter le remplissage anarchique des disques durs. Chaque élève se voit attribuer un certain espace disque qu'il ne peut pas dépasser et qu'il doit donc apprendre à gérer au mieux.

Il est possible de permettre à chaque élève de personnaliser son espace de travail, de mémoriser ses favoris personnels ainsi qu'une personnalisation de chacune des applications utilisées. Ces possibilités présentes dans Window$ sont intéressantes, mais nous estimons qu'il faut en user avec modération afin de limiter le trafic réseau au moment des connexions et afin d'éviter toute une série de problèmes annexes. En particulier, nous estimons inutile de de mémoriser les "favoris" Internet et la configuration de chaque application pour chaque élève. Il nous semble également préférable de forcer la configuration du bureau pour qu'elle reste la même au moins pour les élèves d'une même classe, en généralisant l'usage des liens symboliques comme nous l'avons expliqué plus haut.

Nous utiliserons donc la stratégie suivante (très largement automatisée par l'utilisation d'un petit programme que nous décrirons par après) :

A) Création d'un compte pour chaque groupe-classe et pour chaque élève

On utilise pour cela les commandes système : **groupadd** et **useradd.** Veuillez consulter la documentation de Linux (en faisant appel par exemple aux pages de "man") pour les détails d'utilisation de ces commandes. Le programme que nous décrivons plus loin effectue ces opérations automatiquement.

B) Création de répertoires pour chaque groupe-classe et pour chaque élève

Sur notre serveur, nous allons créer dans le répertoire /home des sous-répertoires pour chacun des groupes-classe (ex : classe5, classe6, ...), avec des droits d'accès 755 (*full access* pour le "propriétaire" du répertoire, accès en lecture seulement pour le reste du monde). Le "propriétaire" de ces répertoires sera défini automatiquement comme un utilisateur dont le nom est identique à celui du groupe-classe (mais converti en minuscules), avec par défaut le mot de passe d'accès "abcde" (Voir plus loin l'utilité de ces choix). Ces répertoires étant créés sous 'root', on utilise les commandes

```
chown  nom-du-propriétaire  nom-du-répertoire
chgrp  nom-du-groupe  nom-du-répertoire
```

pour s'assurer que leurs propriétaires et groupes sont bien ce qu'il faut.

Dans chacun de ces répertoires de classe, on crée les sous-répertoires suivants :

/bureau /apps /start /smenu

Ces répertoires contiendront tout ce qui est nécessaire pour personnaliser l'environnement de travail associé à ce groupe-classe sous Window$ (voir plus loin).
Si l'on réalise tout ceci "à la main", il faut lancer les commandes :

```
chgrp nom-du-propriétaire  *
chmod 755  *
```

... pour s'assurer que les propriétaires et les droits d'accès soient corrects.

A ces quatre sous-répertoires communs, il faut alors ajouter un sous-répertoire personnel pour chaque élève du groupe, chacun étant instauré propriétaire de son sous-répertoire personnel, avec des droits d'accès "2770" (*full acces* pour lui, *full access* pour un groupe de professeurs à préciser, et attribution automatique à ce même groupe de professeurs de tous les fichiers créés ultérieurement dans ce répertoire).

La structure du répertoire /home devrait dès lors ressembler à ceci :

répertoires	attributs			remarques
/home	755	root	root	
/classe4	755	chef	classe4	
/apps	755	chef	classe4	ces quatre sous-répertoires contiennent l'environnement de travail attribué au groupe d'élèves (paramétrage du bureau Windows, icônes, applications accessibles, etc.)
/bureau	755	chef	classe4	
/start	755	chef	classe4	
/smenu	755	chef	classe4	
/jules	770	jules	profs3	ce sont les sous-répertoires personnels de chaque élève. Ils sont accessibles aussi aux membres d'un groupe de professeurs
/henri	770	henri	profs3	
/olga	770	olga	profs3	
...				
/classe4b				
...				

Il va de soi que la création "manuelle" de tous ces répertoires serait longue et fastidieuse, d'autant plus qu'il faudra ajouter à tout ceci la définition des quotas d'espace disque réservés à chacun. C'est la raison pour laquelle nous vous proposons deux petits logiciels destinés à automatiser les opérations. Le premier n'est pas indispensable, mais tout de même précieux : il extrait les données concernant chaque élève de la base de données PROSEC de votre école, crée automatiquement pour chacun d'eux un identifiant unique et un mot de passe, et place le tout dans un fichier texte qui sera lui-même utilisé par le second logiciel. Vous pourriez bien évidemment générer ce fichier texte par d'autres moyens (à l'aide d'un logiciel tableur, par exemple). Ce premier logiciel fonctionne sous Window$ (Utilisez-le sur une machine quelconque). Le second logiciel crée véritablement les comptes utilisateurs décrits précédemment. Il sera lancé sur le serveur Linux.

C) Logiciels pour la création automatique des comptes utilisateurs :

NetUsers est un programme fonctionnant sous Window$. Ce logiciel utilise la base de données ***Prosec*** (disponible en principe au secrétariat de l'école). Il en extrait les informations nécessaires afin que vous puissiez aisément définir des groupes d'élèves en filtrant la base de données à l'aide de critères quelconques. Des noms d'utilisateur et des mots de passe faciles à mémoriser sont alors générés automatiquement pour chaque élève. Le tout est transcrit dans un simple fichier texte, lequel devra être transféré ensuite sur le serveur Linux. Le logiciel s'assure que les noms d'utilisateur soient uniques. Il vérifie également que les noms d'utilisateur et de groupe choisis par vous soient conformes aux conventions Linux décrites ci-après.

Vous pouvez vous passer de ce logiciel, et effectuer le même travail par exemple à l'aide d'un tableur classique. L'essentiel est d'arriver à produire un fichier texte contenant les informations nécessaires pour chaque élève. Sa structure doit être la suivante :

Une ligne pour chaque élève. Chaque ligne contient dans l'ordre, séparés par des espaces :

- le nom d'utilisateur (l'identifiant) choisi pour cet élève. Ce nom doit commencer par une lettre, qui sera suivie d'autres lettres ou de chiffres (pas d'accents !). Il ne peut comporter que 8 caractères au maximum.
- le nom du groupe-classe (mêmes consignes que pour le nom d'utilisateur).
- le mot de passe attribué à l'élève (caractères quelconques : la casse sera significative)
- un commentaire quelconque (40 car. max.). On y placera au moins le nom complet de l'élève.

Exemple :

```
dupoju  Classe4 a6tq95z Jules Dupont – classe de 4e
bertmo  cl5TQ   jb54r2  Monique Bertrand – classe de 5e technique Q
morgsyl cl5TQ   lk34hj2 Sylvie Morgan – 5e TQ
...
```

Le logiciel à utiliser sous Linux pour la création des comptes s'appelle **accounts.py**
Il s'agit d'un script (un programme) écrit en langage***Python***[2]. Il faudra donc qu'un interpréteur ***Python*** soit installé sur votre serveur (cela devrait être le cas si vous avez installé votre serveur en suivant nos directives).
Il faudra également s'assurer que le *démon* **Samba** soit actif, sinon les mots de passe risquent de ne pas être transcrits dans la base de données de Samba (fichier /etc/private/smbpasswd).

*Pour les curieux, voici en résumé le travail effectué par a**ccounts.py** :*

Le programme vérifie si le script mksmbpasswd.sh est disponible, et finit de l'installer si Yast ne l'a pas déjà fait (cfr note de la page 17).

Si la base de données des mots de passe **Samba** n'existe pas encore, elle est créée automatiquement dans /etc/private/smbpasswd. (Le programme crée d'abord un répertoire bien protégé : /etc/private, avec des droits "500", puis y installe le **fichier** smbpasswd à l'aide de la commande :
`cat /etc/passwd | mksmbpasswd.sh > /etc/private/smbpasswd` , laquelle crée le fichier initial et y inscrit des entrées pour chacun des utilisateurs éventuellement déjà présents dans /etc/passwd. Les droits d'accès au fichier /etc/private/smbpasswd sont ensuite établis à "600". Les mots de passe eux-mêmes seront insérés plus tard, à l'aide de la **commande** smbpasswd).

Les groupes-classes, les répertoires pour ces groupes-classes, les comptes utilisateurs sont créés ensuite. Des quotas d'espace disque seront définis pour chacun (par copie d'un prototype). Pour cette question des quotas, veuillez consulter les explications de la page 30.

Pour chaque classe, un groupe et un utilisateur "chef" sont créés automatiquement. Le groupe sera celui auquel les élèves sont incorporés. L'utilisateur chef sera le propriétaire des répertoires créés pour la classe (Ceci permettra de gérer aisément l'environnement de travail, sans que les élèves n'y aient accès). Cet utilisateur reçoit le même nom que celui du groupe.

Pour chaque classe, on crée deux sous-répertoires : l'un dans */home* et un autre dans */home/siteweb :*

- Le premier contiendra les 4 sous-répertoires *bureau, apps, start, smenu* nécessaires pour gérer l'environnement de travail sous Windows, ainsi que des sous-répertoires personnels indépendants pour chacun des élèves de la classe (Un sous-répertoire personnel est accessible seulement à l'élève lui-même et à un groupe de professeurs désigné spécifiquement pour cela).
- Le second contiendra également des sous-répertoires personnels pour chaque élève, mais ceux-ci sont "publics" (c.à.d. accessibles à tous, mais en lecture seulement). Les élèves pourront donc "publier" dans ces répertoires des documents consultables par tous (pages Web, par exemple) qui restent cependant leur propriété protégée.

Les élèves n'ont le droit d'écriture (et donc aussi d'effacement) que dans leurs répertoires personnels. On évite ainsi qu'un élève ne puisse perturber ou pirater le travail d'un autre.

Le groupe associé à chaque répertoire personnel d'élève n'est pas le groupe dont il fait lui-même partie, mais bien un groupe particulier dans lequel on aura inscrit un certain nombre de professeurs désignés spécifiquement pour gérer cette classe (rappelons ici qu'un même utilisateur peut faire partie de plusieurs groupes différents). Sans être administrateurs système, ces professeurs pourront donc accéder aux répertoires de leurs élèves pour y effectuer différents travaux ou contrôles.

2 Python est un extraordinaire langage de programmation de haut niveau, facile à apprendre, utilisable aussi bien sous Window$ ou M@c que sous Linux, et absolument gratuit !
Voir à ce sujet http://www.ulg.ac.be/cifen/inforef/swi/python.htm

Le script a*ccounts.py* vous proposera de créer un tel groupe de professeurs titulaires, différent pour chaque classe si vous le souhaitez (vous pouvez aussi n'en créer qu'un seul ou quelques-uns). Vous disposez ainsi d'une liberté complète pour octroyer des droits précis à chacun.

Pour ajouter ou enlever des utilisateurs à un groupe de professeurs existant, la procédure la plus simple consiste à éditer "à la main" le fichier */etc/group*. (Les noms des utilisateurs membres du groupe sont séparés par des virgules).

Quelques autres tâches à accomplir sur le serveur (en tant que 'root') :

A) Image de fond pour le bureau de chaque groupe :

Dans le répertoire personnel de chaque élève, on peut placer l'image (obligatoirement de type bmp) qui servira de fond d'écran pour le bureau. Son nom doit impérativement être *background* (ce nom a en effet été défini dans la stratégie système de l'utilisateur par défaut – voir page 14), mais cela peut aussi être un lien symbolique portant ce nom et pointant vers une image située ailleurs. Le script ***accounts.py*** crée automatiquement un tel lien symbolique, lequel pointe vers un fichier qui s'appelle lui aussi *background* et qui est censé se trouver dans le répertoire attribué à la classe (*/home/classe4*, par exemple). Si vous souhaitez que tous les élèves d'une même classe partagent la même image pour leur fond d'écran, il vous suffit donc de copier cette image dans le répertoire de la classe, et de la renommer *background* :

```
cp  /dos/graphics/bmp/img055.bmp  /home/classe4/background
```

B) Contenu de /home/classe4/bureau :

On place ici les icônes que l'on veut faire apparaître sur le bureau des membres du groupe. Si l'on se connecte sur une machine locale en tant que 'chef de groupe' (c.à.d. En utilisant le nom du groupe comme identifiant), on pourra disposer de l'explorateur et mettre en place ou enlever des icônes par simple glisser-lâcher. Vérifier que les droits d'accès à ces icônes soient à '755' pour que les autres membres du groupe puissent les utiliser).

C) Contenu du répertoire /home/classe4/apps : ("classe4" pour l'exemple)

Placer dans ce répertoire les raccourcis (Window$) pointant vers les applications autorisées pour les membres du groupe. Ces raccourcis peuvent aussi pointer vers des répertoires.

D) Contenu de /home/classe4/smenu :

On peut placer ici les raccourcis pointant vers les applications réservées au 'chef de groupe' (généralement l'explorateur, la fenêtre MSDOS, etc.). Pour un ordinateur fonctionnant en machine isolée, ces icônes devraient se trouver dans le répertoire '\windows\Start menu'.

E) Contenu de /home/classe4/start :

Placer ici les raccourcis pointant vers les applications à lancer automatiquement lors du logon de l'utilisateur (par exemple l'utilitaire WinPopup.exe)

Note : Pour créer tous ces raccourcis, la procédure la plus simple consiste à créer des répertoires **apps**, **bureau**, **smenu** et **start** dans le répertoire /root du serveur, puis se connecter en tant que *root* sur un PC Window$, et pour y créer tous les raccourcis souhaités, lesquels seront alors automatiquement mémorisés dans le répertoire personnel de *root* sur le serveur (/root). Lorsque l'on examine ensuite les sous-répertoires de /root indiqués plus haut, on y trouve donc les raccourcis en question. On leur attribue à tous des droits "755", et on en redistribue des copies dans les sous-répertoires apps, bureau, smenu, etc. de chaque groupe-classe.

Résultat : Comment les choses se présentent-elles pour les utilisateurs ?

Chaque utilisateur pourra trouver sur le serveur (via le "voisinage réseau") un répertoire à son nom, lequel est accessible également sous le nom générique "homes" (Ce qui autorise de nombreux automatismes, comme vous le comprendrez plus loin). Les données qu'il y placera désormais seront tout à fait inaccessibles aux autres utilisateurs.

Tous les élèves d'un même groupe verront apparaître sur leur écran un même bureau commun (même image de fond, mêmes icônes, etc.)

Lorsque le professeur responsable du groupe désire apporter des modifications au bureau commun, aux applications accessibles, etc., il se connecte lui-même en utilisant le nom du groupe comme identifiant (mot de passe = 'abcde'). Il peut alors modifier le bureau, ajouter ou retirer des icônes, etc. Les élèves n'ont pas ce pouvoir.

Si le partage correspondant a été défini correctement dans */etc/smb.conf*, les élèves ont également accès à un répertoire personnel dans /home/siteweb. Cela pour leur permettre d'installer des pages Web personnelles que les autres utilisateurs de l'intranet puissent consulter.

Logon scripts

Les *'logon scripts'* doivent être des fichiers de type '.BAT' situés sur le serveur, mais dont le lancement est déclenché automatiquement par Samba sur la machine Window$ qui se connecte. Ces scripts doivent être créés de préférence sous Window$ (pour que les séquences <cr><lf> soient correctes), mais ils doivent être placés dans un répertoire partagé sous le nom "netlogon" sur le serveur principal. Si le nom de celui-ci est Boss, ils doivent donc se trouver dans le partage : \\Boss\netlogon (Rappel :Nous avons décidé précédemment que ce partage correspondrait au répertoire /home/netlogon sur notre la machine Linux - voir aussi les lignes correspondantes dans le fichier /etc/smb.conf).

On peut configurer Samba de façon à ce que le script exécuté ainsi soit un fichier '.BAT' dont le nom soit formé à partir du nom de l'**utilisateur** qui se connecte (par exemple 'jules.bat'), ou bien à partir du nom de la **machine** qui se connecte (par exemple 'PC28.bat'), ou encore à partir du nom de **groupe** de l'utilisateur qui se connecte (par exemple 'Classe5.bat'). C'est cette dernière possibilité qui a notre préférence (Nous préférons éviter d'avoir à définir un *logon script* différent pour chaque élève !). Voir plus loin les lignes correspondantes dans /etc/smb.conf

Exemple de *logon script* typique :

```
@CALL COMMUN.BAT                (= appel d'un autre script éventuel)
NET TIME \\BOSS /SET /YES
          (= synchronisation de l'horloge interne avec celle du serveur)
NET USE T: \\BOSS\programs\Scala
          (= définition d'un lecteur réseau)
DEL C:\WINDOWS\CONFIG.POL
          ( = effacement d'un fichier Config.Pol local éventuellement présent
DELTREE /Y C:\WINDOWS\PROFILES\*.*
          ( = effacement des profils locaux éventuels)
ATTRIB -S -H C:\WINDOWS\TEMPOR~1\*.*
DELTREE /Y C:\WINDOWS\TEMPOR~1\*.*
DEL C:\WINDOWS\TEMP\*.* /Y
          (= effacement du contenu de répertoires temporaires)
\\BOSS\NETLOGON\identd.exe -q
          (= lancement d'un serveur d'identification - voir "Configuration de Squid", page 34)
```

Tâches à effectuer sur les postes clients

Ne pas oublier de partager les ressources qui doivent l'être. En particulier, il est bien utile de partager l'entièreté du disque dur local, avec protection par un mot de passe connu seulement de l'administrateur réseau. (Ne pas utiliser le mot de passe root du serveur !) Ainsi l'administrateur pourra aisément effectuer toute une série de tâches sur cette machine au départ du serveur (copier, modifier, effacer des fichiers) en réalisant un "smbmount".

Exemple :

```
smbmount  //Boss/public  /mnt   -U  georges  -P  password
```

Messages d'erreur au logon des PC Window$

Les messages d'erreur délivrés par Window$ sont souvent assez fantaisistes.

Lorsqu'un poste qui tente de se connecter reçoit le message "*Impossible d'accéder à ce dossier. Le chemin est trop long*", cela peut signifier en fait que l'utilisateur ne dispose pas des droits d'accès requis pour des répertoires demandés sur le serveur. Vérifier en particulier que la section [homes] dans smb.conf comporte bien le paramètre 'browseable = yes'

Si le message reçu est "*Le mot de passe est incorrect. Réessayez*", c'est que l'utilisateur lui-même n'est pas trouvé dans la base de données du système.

Si c'est seulement le mot de passe qui est erroné, le message doit être : "*Le mot de passe du domaine que vous avez fourni n'est pas correct, ou l'accès au serveur de session a échoué*".

Si le message reçu est "*Aucun serveur de domaine n'était disponible pour valider votre mot de passe. Vous ne pourrez peut-être pas avoir accès à certaines ressources réseau.*", pensez à vérifier que le nom de domaine entré dans le champ ad hoc est bien orthographié correctement. Si le problème persiste, le serveur est peut-être réellement inaccessible pour diverses raisons (vérifiez les connexions, câbles, cartes réseau, etc.)

Le répertoire [netlogon] défini dans smb.conf doit être rendu accessible à tous (en lecture), sinon des messages du type "*Le partage est introuvable. Assurez-vous que le type a bien été entré correctement.*" vont être envoyés aux utilisateurs (autres que root) qui se connectent.

Fichier /etc/smb.conf typique, commenté

```
#==================================
# Fichier de configuration de Samba

# Voir le manuel "L'intro / SAMBA" édité chez CampusPress
# pour informations complémentaires.
# Les lignes commençant par # ou ; sont ignorées.
#
# NOTE: Après chaque modification dans ce fichier, veuillez lancer la
# commande "testparm" pour vérifier si vous n'avez pas commis de faute(s)
# de syntaxe, et aussi pour voir l'ensemble des paramètres (y compris ceux
# qui ont simplement leur valeur par défaut).

#========= Paramètres globaux =========
[global]

# workgroup : Indiquer ici le Domaine Windows NT
  workgroup = CFORM

# Commentaire décrivant le serveur (apparaîtra dans "voisinage réseau")
# %v est une variable système qui sera remplacée automatiquement par
# le numéro de version de Samba
  server string = Linux Samba Server %v

# Hôtes acceptés.
# Cette option est importante pour la sécurité du réseau.
# La ligne ci-dessous restreint l'accès au machines du réseau
# de classe C 192.168.0. ainsi qu'à l'interface "loopback"
# On pourrait ajouter d'autres réseaux.
  hosts allow = 192.168.0.  127.

# Si vous voulez charger automatiquement votre liste d'imprimantes
# plutôt que de les installer manuellement, il vous faut ceci :
  printcap name = /etc/printcap
  load printers = yes

# Il ne devrait pas être nécessaire de préciser la méthode d'impression
# sauf si celle-ci est spéciale (bsd et cups sont les plus classiques)
  printing = bsd

# Activez ceci si vous voulez préciser un compte "invité" autre
# que "nobody" (ce qui est le choix par défaut mais non idéal)
  guest account = ftp

# Ce qui suit force Samba à utiliser un fichier d'historique
# (log file) différent pour chaque machine qui se connecte :
  log file = /var/log/samba/log.%m
  lock directory = /var/lock/samba
  locking = yes
  strict locking = no
  share modes = yes

# Ceci limitera la taille des fichiers d'historique (en kb) :
  max log size = 50

# Mode de gestion de la sécurité. En général on préférera le mode "user".
# (Voir littérature ou fichier Security_level.txt pour les détails)
  security = user

# L'option ci-dessous est à utiliser seulement en conjonction avec le
# mode "security = server" (Cette option s'utilise dans le cas où le
# présent serveur est un serveur secondaire - voir littérature):
;   password server = <Nom du serveur principal>

# Les options ci-dessous permettent de préciser combien de caractères
# doivent être lus sans tenir compte de la casse (majusc./minusc.)
# dans le mot de passe entré par l'utilisateur
;  password level = 8
;  username level = 8
;  null passwords = yes          (à éviter !)

# Si vous avez des postes clients Win98/Me et/ou Win NT, vous souhaiterez
# probablement conserver l'encryptage des mots de passe (que pratiquent
# ces systèmes par défaut - Win95 envoie ses mots de passe "en clair").
# Attention : ceci n'est pas toujours simple. Lisez le fichier
# ENCRYPTION.txt ou bien une bonne documentation de Samba en cas de problème.
  encrypt passwords = yes
  smb passwd file = /etc/private/smbpasswd

# Le fichier défini ci-dessous sert à la conversion des noms d'utilisateur
# "longs" (Windows accepte des noms assez longs pouvant contenir des
# espaces, alors que les systèmes Unix n'acceptent que 8 caractères :
  username map = /etc/smbusers
```

```
# (il faudra évidemment créer le fichier correspondant)

# La ligne ci-dessous vous permettrait de personnaliser la configuration
# de Samba différemment pour chaque machine qui se connecte.
# La variable %m sera automatiquement
# remplacée par le nom Netbios de la machine :
;   include = /etc/smb.conf.%m

# Cette option améliore la vitesse de Samba dans la plupart des cas
# Voir le fichier speed.txt et les pages de "man" pour les details.
   socket options = TCP_NODELAY

# Cette option configure Samba pour plusieurs interfaces
# Si vous avez plusieurs interfaces réseau, vous devez les lister ici.
;   interfaces = 192.168.12.2/24 192.168.13.2/24

# Options concernant le rôle de Samba comme contrôleur de domaine :
# ----------------------------------------------------------------
# Si vous NE SOUHAITEZ PAS que Samba soit le contrôleur principal
# du domaine sur votre réseau (cas d'un poste client), choisissez "no".
# Si vous choisissez "yes", les règles d'élection habituelles
# seront d'application.
  local master = yes

# Le niveau d'OS indique l'importance de ce serveur en tant que
# candidat au rôle de contrôleur principal lorsqu'une élection
# est provoquée. Choisissez zéro si vous ne voulez pas que Samba
# soit contrôleur principal (cas d'un poste client).
  os level = 33

# L'option ci-dessous définit Samba comme le Contrôleur de domaîne
# principal (maître). Ceci permet à Samba de collationner les listes
# de partages entre les sous-réseaux. N'utilisez pas ceci si vous
# disposez déjà d'un serveur NT pour effectuer ce travail.
  domain master = yes

# L'option ci-dessous indique à Samba de forcer une élection de contrôleur
# de domaine au démarrage, et lui donne ainsi une petite chance de gagner
# lors de cette élection
  preferred master = yes

# Utilisez ceci seulement si vous avez déjà sur votre réseau un serveur NT
# configuré comme contrôleur principal de domaine (cas d'un poste client)
;   domain controller = 192.168.0.100

# Activez ce qui suit si vous voulez activer des "logon scripts"
# lorsque les utilisateurs se connectent sur des postes
# Win95, 98, Me ou NT :
  domain logons = yes

# Les "logon scripts" peuvent être définis sur la base du nom de
# l'utilisateur, du nom du groupe, ou même du nom de la machine
# qui se connecte. Les variables %U , %G et %m seront automatiquement
# "remplies" par Samba :
# Exemples :
# un logon script spécifique pour chaque machine :
;   logon script = %m.bat
# un logon script spécifique pour chaque utilisateur :
;   logon script = %U.bat
# notre choix : un logon script spécifique pour chaque groupe d'utilisateurs :
  logon script = %G.bat

# Où faut-il placer les profils itinérants des utilisateurs (Windows) ?
# (par défaut, ils sont mémorisés dans le rép. perso. de chaque utilisateur)
# Note : vous préférons personnellement nous passer de ces "profils" qui
# consomment les ressources du système à un taux inacceptable
# (la variable %L contiendra le nom NetBIOS de ce serveur
#  la variable %U contiendra le nom d'utilisateur)
# Vous devrez également activer le partage [Profiles]
# (voir plus loin)
  logon path = \\%L\Profiles\%U


# Tous les noms NetBIOS doivent être résolus en adresses IP.
# La ligne ci-dessous indique l'ordre dans lequel les différents mécanismes de
# résolution de noms doivent être invoqués. Par défaut, cet ordre est :
# "host lmhosts wins bcast".
# "host" désigne ici la fonction système Unix gethostbyname() qui utilisera
# /etc/hosts ou DNS ou NIS pour la résolution des noms, suivant les choix
# définis dans /etc/host.config, /etc/nsswitch.conf et /etc/resolv.conf .
# "host" désigne donc une méthode qui dépend de la configuration système.
# Ce paramètre est utilisé le plus souvent pour éviter les appels à DNS
# pour la résolution des noms NetBIOS. Utiliser avec prudence !
# L'exemple ci-dessous empêche la résolution des noms pour les machines qui
# NE font PAS partie du réseau local -OU- ne sont pas à rechercher
# délibérément via lmhosts ou via WINS.
```

```
  name resolve order = wins lmhosts bcast

# Section WINS (Windows Internet Name Serving Support) :
# Un des composants de Samba peut fonctionner comme serveur WINS.
# On l'active à l'aide de l'option ci-dessous :
  wins support = yes

# Activez ce qui suit si votre réseau dispose déjà d'un serveur WINS
# (par exemple une machine Windows NT server). Samba fonctionnera alors
# comme un client WINS. Attention : Samba peut fonctionner comme serveur WINS
# ou client WINS, mais non les deux à la fois !
# Il faut bien évidemment préciser ici l'adresse IP # de ce serveur WINS :
;   wins server = w.x.y.z

# WINS Proxy :
# L'option ci-dessous force Samba à répondre aux demandes de résolution
# de noms émanant de clients autres que ceux qui utilisent le protocole
# WINS. Attention : il faut qu'au moins un serveur WINS soit actif !
# (le choix par défaut est 'no')
  wins proxy = yes

# DNS Proxy :
# Samba peut aussi fonctionner comme serveur DNS. Ceci n'est cependant
# plus guère recommandé :
  dns proxy = no

# Le respect de la casse (Maj/minuscules) peut être utile (le choix système
# par défaut est de ne pas la respecter : preserve case = no)
# NOTE: ceci peut être activé sur une base individualisée (suivant machine)
  preserve case = yes
  short preserve case = yes

# La casse par défaut est normalement upper (MAJ.) pour les fichiers DOS
  default case = lower
# Soyez très prudent si vous activez la sensibilité à la casse ! :
;  case sensitive = no

#======================= Définition des partages ========================
# Indiquer entre crochets le nom de partage, ensuite toutes ses propriétés.
# S'inspirer des exemples ci-dessous, ou bien lire la documentation !
#========================================================================

# Le partage ci-dessous apparaîtra comme répertoire personnel
# (et donc à son nom) pour l'utilisateur qui se connecte au serveur.
# (Samba remplacera automatiquement 'homes' par le nom de l'utilisateur) :
[homes]
   comment = Répertoire personnel
   browseable = yes
   writable = yes
   directory mask = 2770
   create mode = 0770
   public = no
   root  preexec  /bin/backprint.py  %U  %m

# la dernière ligne ci-dessus provoque le lancement automatique du logiciel
# /bin/backprint.py sur le serveur, au moment où l'utilisateur accède
# à ce partage. Le logiciel est exécuté avec des droits de root et deux
# paramètres (nom d'utilisdateur, nom de machine) lui sont transmis
# voir à ce sujet les notes concernant backprint.py

# Dans ce répertoire, on placera les "logon scripts", le fichier
# définissant les stratégies système (config.pol), ainsi que d'autres outils
# concernant la gestion des utilisateurs.
# Rem : ce partage doit être "browseable" et accessible à tous pour que le
# mécanisme de connexion fonctionne correctement (Ne pas restreindre l'accès
# par l'option "valid users = nn mm", par exemple).
# Il faudra donc s'assurer (par les droits d'accès Linux) que les élèves ne
# puissent pas modifier ou effacer les fichiers (logon scripts par ex.)
# qui se trouvent ici.
[netlogon]
   comment = Network Logon Service
   path = /home/netlogon
   writable = yes
   browseable = yes

# Activez le partage ci-dessous si vous désirez que les profils utilisateurs
# "itinérants" de Windows soient mémorisés à cet emplacement précis plutôt
# que dans les rép. perso. de chaque utilisateur.
# Vous pouvez inhiber la mémorisation de ces profils en indiquant ici un
# répertoire inaccessible (ce que nous faisons personnellement) :
[Profiles]
   path = /home/profiles
   browseable = yes
   guest ok = yes
   public = yes
   writable =yes
```

```
# NOTE: Si vous utilisez le système d'impression de type BSD il n'est pas
# nécessaire de définir individuellement chaque imprimante
# Activez <public = yes> pour autoriser l'impression par tous.
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   public = yes
   guest ok = yes
   writable = no
   printable = yes
   create mode = 0777

# Définition d'un partage destiné à recevoir les pages web d'Intranet créées
# et gérées par les élèves. Chacun d'eux disposera ici d'un sous-répertoire
# personnel pour y placer ses pages.
[siteweb]
   comment = site web d'intranet
   path = /home/siteweb
   browseable = yes
   writable = yes
   create mode 0775
   directory mask = 2775

# ======== Autres exemples. ============
# Imprimante utilisable seulement par l'utilisateur 'Jules'. Les données de
# spool seront envoyées dans son répertoire personnel.
[Julesprn]
   comment = imprimante de Jules
   valid users = Jules
   path = /homes/Jules
   printer = Jules_printer
   public = no
   writable = no
   printable = yes

# Un partage privé, utilisable seulement par Jules et Fernande.
[JuFern]
   comment = Chez Jules & Fernande
   path = /home/JF/private
   valid users = Jules, Fernande
   public = no
   writable = yes

# Un partage qui correspond à un répertoire différent pour chaque machine
# qui se connecte (la variable %m est automatiquement remplacée par le nom
# netbios de la machine). Ceci vous permet d'affiner une foule de
# choses pour les postes clients qui se connectent. Vous pourriez également
# faire en sorte que ce répertoire soit différent pour chaque utilisateur qui
# se connecte, en remplaçant la variable %m par la variable %u.
[pchome]
  comment = PC Directories
  path = /usr/pc/%m
  public = no
  writable = yes

# Le partage ci-dessous est accessible à tous (full access puisque les droits
# d'accès aux fichiers et aux répertoires que l'on y crée sont 777).
# On peut évidemment définir d'autres masques si l'on désire (par exemple)
# que les fichiers et/ou les répertoires créées ici restent privés.
[Commun]
   comment = Repertoire public
   path = /home/commun
   public = yes
   writable = yes
   browsable = yes
   directory mask = 2777
   create mode = 0777

# Partage accessible à tous, mais en lecture seule.
# Les membres du groupe "staff" sont les seuls autorisés à y enregistrer qqc.
[documentation]
   comment = Public Stuff
   path = /home/archives
   public = yes
   writable = yes
   write list = @staff
```

Installation de quotas disque :

Le noyau standard SuSE est déjà compilé pour prendre en compte les quota. Il suffira donc de :

- installer le paquetage *quota* de la série *ap*
- éditer le fichier */etc/fstab* : ajouter l'attribut *usrquota* pour toutes les partitions qui seront contrôlées par le gestionnaire de quotas.
  Exemple : /dev/sda5 /home ext2 defaults,usrquota 1 2
- Sous *Yast*, activer les quotas dans le fichier /etc/rc.config : **START_QUOTA = yes**
- rebooter (c'est nécessaire, pour une fois : le système de fichiers lui-même est modifié.)
- lancer la commande *quotacheck* pour initialiser le fichier de description des quotas (lequel sera situé dans /home/aquota.user) . Lancer les 2 commandes ci-dessous pour SuSE 7.2 :
  `quotacheck -acF vfsv0 (! : dernier caractère = zéro)`
  `quotacheck -amv`
  Note : Ne pas lancer *quotacheck* si les quotas sont actifs (voir *quotaoff -av* ci-dessous)
  Pour SuSE <7.2, lancer seulement la commande : **`quotacheck -avug`**

Utilisation :

- On peut activer/désactiver les quotas avec : **`quotaon -av / quotaoff -av`**
- Pour établir les quotas de l'utilisateur *trucmuche*, entrer :
  `edquota -u trucmuche`
  Cette commande provoque l'appel d'un logiciel éditeur appelé **vi** , dont le maniement est assez rébarbatif aux yeux d'une personne plutôt habituée à l'univers DOS/Windows. La page qui apparaît décrit la situation des quotas actuels pour l'utilisateur choisi.
  - La ligne ***blocks in use*** précise (en blocs de 1Ko) la place déjà occupée par les fichiers appartenant à l'utilisateur, ainsi que deux limites (***soft*** et ***hard***), également exprimées en blocs de 1Ko. Le paramètre *hard* fixe la limite supérieure absolue que l'utilisateur ne pourra pas dépasser. Le paramètre *soft* spécifie une limite "douce" qui peut être dépassée pendant un certain temps (mais dont les messages d'avertissement destinés à l'utilisateur n'apparaissent pas correctement sur les postes clients Windows).
  - La ligne ***inodes in use*** décrit les limites fixées en termes de nombres de fichiers. Vous pourrez donc également définir une limite de ce type si cela vous paraît nécessaire.

  En définitive, le seul paramètre véritablement important est le paramètre *hard*. S'il est laissé égal à zéro, aucune limite n'est imposée. Si vous souhaitez établir une limite de 20 méga-octets, par exemple, il faut donner à ce paramètre la valeur 20000.
  Note : dans l'éditeur vi, il faut d'abord utiliser la touche "i" *(= insertion) pour pouvoir effectivement éditer le texte. Pour quitter l'éditeur et enregistrer les modifications, il vous faudra enfoncer successivement les touches* "escape", ":", "w" (= écrire) & "q" (= quitter).
- Chaque utilisateur peut connaître son statut courant avec la commande : **`quota`**
- L'administrateur peut visualiser tous les quotas à l'aide de la commande : **`repquota -av`**
- On peut attribuer à l'utilisateur *trucmuche* les quotas de l 'utilisateur *machin* par :
  `edquota -p machin -u trucmuche` . Il est donc possible de définir un utilisateur "prototype" et d'automatiser ensuite l'attribution de ses quotas à tout un groupe d'utilisateurs. C'est cette méthode qui est utilisée dans le script "accounts.py" (Voir + loin).

Notes:

- Le fichier contenant les quotas est /home/aquota.user (ou /home/quota.user - anc. version) relancer **`quotacheck -avm`** de temps à autre pour contrôler et régénérer ce fichier
- Le script *accounts.py* attribuera automatiquement des quotas à tous vos élèves. Avant de l'utiliser, il faut créer d'abord quelques utilisateurs "prototypes" auxquels vous attribuez des quotas de référence. Créez par exemple des utilisateurs **quota10**, **quota20**, **quota50**, etc., auxquels vous attribuez des quotas d'espace disque de 10 Mb, 20Mb, 50 Mb, ...

Sauvegarde et restauration de l'intégralité d'une partition Window$

Si les postes de travail sont équipés d'un disque dur de taille suffisante, on aura grand intérêt à y installer une petite partition Linux (même si l'on ne souhaite pas utiliser ces machines sous Linux en temps ordinaire). A partir de Linux, on peut en effet sauvegarder l'intégralité de la partition Window$ de la machine, sous forme comprimée, à l'aide du logiciel d'archivage **tar** .

Exemple : Supposons que le point de montage de la partition Window$ doit /dos, et que l'on désire effectuer la sauvegarde sous le nom *winbak* dans le répertoire */archives* (lequel peut bien entendu être situé sur une autre machine, et rendu accessible par NFS). Il suffira de faire :

```
cd  /dos
tar  -cvzf  /archives/winbak *
```

pour effectuer la sauvegarde.

Pour la restauration, il suffira de faire :

```
cd  /dos
rm  -rf  *                    (pour effacer intégralement la partition existante)
tar  -xvzf  /archives/winbak     (pour rétablir la partition à partir de sa
sauvegarde)
```

Attention !!! La commande **`rm -rf *`** est très dangereuse : vérifiez trois fois plutôt qu'une que vous vous trouvez bien dans le bon répertoire, avant de la lancer !!!

Si la sauvegarde est effectuée sur une machine distante (le serveur principal, par exemple), et si l'on dispose d'un certain nombre de postes de travail identiques, il sera extrêmement pratique d'utiliser cette méthode pour effectuer une maintenance régulière de ces machines. Si les machines sont toutes différentes, par contre, il sera peut-être préférable d'effectuer des sauvegardes locales (mais il faut alors disposer d'un espace disque suffisant pour la partition Linux : compter que la sauvegarde comprimée occupe environ la moitié de l'espace occupé par la partition originale).

Utilisation de la disquette TomsRtBt

TomsRtBt est une micro-distribution de Linux qui réussit l'exploit de tenir tout entière sur une seule disquette. Vous pouvez vous procurer cette distribution sur le site officiel *TomsRtBt* : http://www.toms.net. Nous en avons préparé une version modifiée à votre intention sur notre site : http://ulg.ac.be/cifen/inforef/swi. Dans cette version modifiée, nous avons enlevé le support PCMCIA pour récupérer un peu d'espace disque et pouvoir ainsi ajouter un vrai logiciel d'archivage tar (version GNU) ainsi que quelques pilotes de cartes réseau supplémentaires (Via-Rhine II, RealTek 8139).

Considérons donc que vous disposez d'une disquette *TomsRtBt* fonctionnelle. A titre d'exercice pour apprendre à nous en servir, nous allons effectuer une sauvegarde complète de l'installation de Window$ existant sur un PC quelconque, et l'expédier sous forme comprimée dans un répertoire d'archivage situé sur une autre machine, à savoir un serveur Linux accessible via le réseau local.

A) Préparation du serveur d'archivage

Avant toute autre chose, il faut vous assurer que le serveur distant soit prêt à recevoir les fichiers que vous désirez lui envoyer. Connectez-vous en tant que "root" sur ce serveur, et créez-y le répertoire *ad hoc* (par exemple un sous-répertoire de /home) :

```
cd  /home
mkdir  winarch
chmod  777  winarch     ("full access" pour contourner quelques difficultés)
```

De l'extérieur, vous accéderez à ce répertoire à l'aide d'un montage NFS. Il faut donc configurer le serveur NFS pour qu'il accepte le "partage" de ce répertoire. Pour ce faire, il faut éditer le fichier de configuration /etc/exports déjà mentionné page 10. Ouvrez donc le fichier /etc/exports à l'aide d'un éditeur quelconque, pour y ajouter une ligne telle que :

```
/home/winarch    (rw, no_root_squash)
```

rw indique que le répertoire sera accessible en lecture/écriture, **no_root_squash** indique que la connexion distante en tant que "root" devrait (en principe) être acceptée.

Après avoir sauvegardé votre modifications, vous devez faire en sorte qu'elle soit prise en compte par le serveur NFS. Il suffit de relancer celui-ci par :

`/etc/init.d/nfsserver restart` ou encore par : **`rcnfsserver restart`**

B) Démarrage de Linux sur le poste Window$ à archiver

Insérez la disquette *TomsRtBt* dans le lecteur et redémarrez la machine. Durant le processus de démarrage, vous avez la possibilité de choisir la résolution de l'écran (choisissez 3, par exemple), et surtout de choisir la disposition de votre clavier. Guettez l'apparition de cette invite, sinon vous vous retrouverez par défaut avec la disposition des touches d'un clavier US.

Lorsque l'initialisation est terminée, vous disposez d'un Linux très complet entièrement situé en RAM. (Vous pouvez désormais enlever la disquette du lecteur). Durant l'initialisation, votre carte réseau a peut-être déjà été détectée et son pilote déjà installé automatiquement (cas des cartes 3COM, DEC "Tulip", D-Link DFE-500TX, NE 2000, Intel EEpro100 ...). Sur notre version de la disquette, nous avons ajouté les pilotes de deux cartes courantes, que vous pouvez installer à l'aide d'une des commandes ci-dessous :

```
insmod  via-rhine       pour Via-Rhine II (ou D-Link DFE-530TX)
insmod  rtl8139.        Pour RealTek 8139
```

Pour configurer l'interface réseau et le routage vers le réseau local, entrez les commandes :

```
ifconfig  eth0  192.168.0.200                  (-> choix de l'adresse IP)
route  add  -net  192.168.0.0                  (choix du réseau local)
```

Dans l'exemple ci-dessus, la machine sur laquelle on travaille reçoit l'adresse IP 192.168.0.200

Pour vérifier si la connexion réseau est bien établie, vous pouvez bien évidemment lancer une commande **ping** en direction de votre serveur distant (admettons pour l'exemple que l'adresse de celui-ci soit 192.168.0.100) :

`ping 192.168.0.100` -> etc. (interrompre par CTRL-C)

C) Montage de la partition Window$ - Montage distant par NFS

Vous disposez donc à présent d'un système Linux tout à fait opérationnel, mais qui ne connaît encore de la machine locale que sa RAM. Pour passer à l'exploration du disque dur, vous devez effectuer un montage de l'une ou l'autre de ses partitions dans l'arborescence des répertoires. En règle générale, la partition qui contient Window$ est la première partition du premier disque (/dev/hda1). Vous pouvez donc y accéder en lançant les commandes :

```
cd /
mkdir  dos                              (création du point de montage)
mount  -t  vfat  /dev/hda1  /dos        (montage proprement dit)
```

D'une manière similaire, vous effectuez le montage du répertoire d'archivage, via NFS :

```
mount  -t  nfs  192.168.0.100:/home/winarch  /mnt
```

Vous aurez remarqué que nous utilisons pour ce dernier le point de montage "classique" : **/mnt**

D) Archivage à l'aide du logiciel tar

A ce stade, votre partition Window$ locale se trouve donc dans le répertoire **/dos** ,
et le répertoire d'archivage de votre serveur distant se trouve dans **/mnt**

Vous pourriez donc effectuer une simple copie de l'un dans l'autre à l'aide de la commande **cp** (en veillant toutefois à inclure de manière récursive tous les sous-répertoires dans la copie), mais vous occuperiez ainsi un espace disque excessif. Vous pouvez faire mieux à en faisant appel au logiciel **tar** (*tape archiver*). Lancez les commandes :

```
cd  /dos
tar  -cvzpf  /mnt/pc25.tgz  *
```

Parmi les arguments fournis, **c** signifie : création (d'une archive) ; **v** signifie : verbeuse (le logiciel doit afficher à l'écran l'évolution du travail) ; **z** signifie qu'il faut comprimer l'archive suivant l'algorithme zip ; **f** indique que ce qui suit est le nom du fichier d'archivage. L'astérisque est utilisée pour signifier à **tar** qu'il faut archiver récursivement tout ce qui se trouve dans le répertoire courant. L'extension **.tgz** accolée au nom de fichier choisi pour l'archivage est facultative, mais conseillée (elle indique une archive de type "tar zipée").

E) Restauration de l'image archivée

Si tout s'est bien passé, vous disposez maintenant d'une image comprimée de votre partition Window$, sous la forme d'un fichier situé dans /home/winarch/pc25.tgz sur votre serveur.

Pour effectuer la restauration de cette image, vous devrez démarrer la machine concernée en répétant les étapes B et C ci-dessus.

La restauration sera alors effectuée à l'aide des commandes :

```
cd  /mnt
ls -l                        (pour vérifier que l'archive est bien accessible)
cd  /dos                 (pour se positionner sur la partition cible)
ls                       (pour vérifier que l'on est bien au bon endroit !)
rm  -rf  *               (pour effacer tout de manière récursive  !!! Attention
!!! )
tar  -xvzf  /mnt/pc25.tgz        (pour restaurer l'archive)
```

C'est tout.

Installation de SQUID 2.3 (serveur proxy)

Squid est un serveur *proxy* extrêmement perfectionné qui vous permet de contrôler et d'optimiser le trafic réseau entre vos postes de travail et l'internet. Sa fonction de cache accélère très fortement l'accès aux pages *web* qui sont demandées par plusieurs utilisateurs, et ses diverses fonctions de filtrage permettent de réglementer efficacement les connexions.

Installer le paquetage squid (série n)

Infos utiles : Les messages de log seront envoyés dans /var/squid.
Squid lui-même est installé dans /usr/sbin/squid.
Le cache sera installé dans /var/squid/cache.

Pour configurer Squid, il suffit d'éditer le fichier /etc/squid.conf

Pour démarrer/arrêter/redémarrer Squid, utiliser l'une des commandes :

```
/etc/init.d/squid  start
/etc/init.d/squid  stop
/etc/init.d/squid  restart
```

Pour configurer le démarrage automatique de Squid, comme celui de la plupart des autres services, on utilisera de préférence Yast ou Yast2. Exemple sous Yast2 :

Divers → Editeur RC-Config → Start-variables → Start-Network → Start Squid = Yes

Il faut bien entendu s'assurer aussi que la machine sur laquelle on installe Squid soit elle-même capable de dialoguer avec l'internet. Configuration sous Yast2 :

Réseau/Avancé → Routage → Passerelle par défaut : 192.168.0.75 (adresse du routeur Internet)

Réseau/Base → HostName & DNS : 212.166.2.11 212.166.3.106 (adresses de serveurs DNS)

Editer le fichier /etc/squid.conf

Ce fichier est fort long car très documenté. Ne soyez pas effrayé par la multitude d'options disponibles (vous explorerez peut-être tout cela petit à petit), car vous pouvez conserver la valeur par défaut pour la plupart d'entre elles. Voici cependant un petit guide pour vous y retrouver (à consulter aussi : le site "Linux pour les lycées luxembourgeois" **http:/www.lll.org.lu** qui consacre un certain nombre de pages à la configuration de Squid)

A) Numéro de port utilisé – taille du cache et des paquets

Par défaut, Squid écoute les requêtes envoyées par ses clients (c.à.d. les browsers Web : Netscape, etc.) sur le port 3128. Les browsers proposent souvent par défaut le port 8080. Choisissez l'un ou l'autre, mais comprenez bien que le serveur et ses clients doivent être d'accord sur ce point !

Attention: ne pas utiliser le N° de port 80 qu'il est préférable de réserver plutôt à *Apache*.

Si vous choisissez le port 8080, il faut activer une ligne telle que :

```
http_port  8080
```

Vous pouvez également modifier les tailles définies par défaut pour les deux caches : celui qui est situé en mémoire vive et celui qui sera créé sur le disque dur du serveur (tout ceci en fonction des possibilités de votre machine, bien entendu). Recherchez les lignes similaires aux suivantes dans /etc/squid.conf, modifiez-les en fonction de vos propres critères, et activez-les :

```
cache_mem  32 MB
maximum_object_size 4096 KB
cache_dir  ufs  /var/squid/cache  200  16  256
```

La seconde ligne signifie que les "objets" de plus de 4 Mb ne seront pas "cachés". Vous avez d'ailleurs probablement intérêt à limiter la taille des fichiers téléchargeables, en activant la ligne :

```
reply_body_max_size  2000 KB
```

B) Filtrage des accès.

Le mécanisme utilisé par *Squid* pour filtrer l'accès à l'internet consiste à analyser les informations contenues dans l'en-tête de la requête envoyée par le browser client, et à accepter ou rejeter cette requête en fonction de règles bien définies. On peut établir des règles d'accès de tous types dans le fichier de configuration /etc/squid.conf. Ces règles sont toujours composées de deux parties :

- la première consiste à définir d'abord des "étiquettes" qui correspondent à la description d'une catégorie particulière d'informations contenues dans la requête : les lignes correspondantes commencent par le mot-clé **acl** (pour *accesss control list*)
- la seconde consiste à établir les règles d'accès proprement dites s'appliquant à ces étiquettes. Les lignes correspondantes commencent par le mot-clé **http_access** .

Dans le fichier de configuration d'origine, vous trouverez déjà en place un certain nombre de lignes **acl** et **http_access** prédéfinies. Laissez-les telles quelles, et veillez à ajouter ou activer celles que nous décrivons ci-après.

Certaines **acl** désignent des groupes particuliers de machines. Il faut d'abord bien distinguer les machines clientes, ou "sources", c.à.d. les PC du réseau local qui vont faire appel à *Squid* pour se connecter à l'internet, et les machines "destinataires", c.à.d. les ordinateurs que les clients cherchent à joindre via *Squid* et l'internet. Pour désigner toutes ces machines, on peut utiliser des combinaisons adresseIP/masque, ou toute une série d'autres conventions. Chaque ligne **acl** associe un nom, un type, et une ou plusieurs données. Par exemple :

```
acl  all  src  0.0.0.0/0.0.0.0
```

cette ligne associe l'étiquette **all** à toutes les adresses IP possibles pour des machines clientes (src). Comme vous le verrez un peu plus loin, nous nous servirons de cette étiquette pour bloquer l'accès à toute machine **autre** que celles que nous aurons explicitement autorisées.

```
acl  admis  src  "/home/netlogon/machines_admises"
```

Cette ligne définit le groupe de machines qui seront autorisées à se connecter au proxy. Elle associe l'étiquette **admis** au type **src** (clients) et à une liste de noms ou d'adresses IP se trouvent dans le fichier indiqué. Ces machines seront évidemment celles de notre propre réseau scolaire. Nous préférons placer cette liste dans un fichier externe plutôt que directement dans /etc/squid.conf, parce que nous souhaitons pouvoir modifier aisément cette liste, à tout moment en fonction des besoins (par exemple pour restreindre temporairement l'accès internet à une seule salle de classe). Ce sera plus facile avec un fichier indépendant de /etc/squid.conf, et qui ne contient rien d'autre. Remarquez au passage que nous plaçons ce fichier dans le même répertoire (/home/netlogon) que les *logon scripts*, afin de centraliser nos outils de gestion.
Ce fichier contiendra des lignes telles que :

```
192.168.0.125-192.168.0.130
192.168.0.171
192.168.0.102-192.168.0.110
```

Nous pourrons nous faciliter la gestion en (ré)générant ce fichier à la demande à l'aide d'un script (éventuellement avec interface graphique). Voir par ex. le script *squidconf.py* de G.Swinnen.
Pour l'instant, continuons notre examen des autres lignes **acl** de /etc/squid.conf :

```
acl  explicit myip     192.168.0.100
```

Cette ligne désigne l'adresse du serveur proxy lui-même. Elle est destinée à distinguer les requêtes que les clients auront envoyées au proxy de manière explicite (elles devraient l'être toutes), de celles qui peuvent dans certains cas être envoyées de manière implicite via un serveur Web.

```
acl  ident    ident    REQUIRED
acl  punis    ident    "/home/netlogon/utilisateurs_punis"
```

On définit ici deux **acl**, l'une pour établir que l'identification des utilisateurs sera requise, et l'autre pour signaler que la liste des utilisateurs interdits d'accès se trouve encore une fois dans un fichier externe bien déterminé (celui-ci contiendra un identifiant par ligne, en caractères minuscules). Notez bien que ce mécanisme de contrôle d'accès par les noms d'utilisateurs ne pourra fonctionner que si un logiciel serveur d'identification a été mis en service sur les postes clients (voir plus loin).

```
acl  censure1 url_regex     ^http://www.porno.com
acl  telechar      urlpath_regex  -i  \.exe$  \.zip$  \.mp3$  \.mpg$  \.avi$
acl  FTP      proto         FTP
acl  censure2 dst           213.193.0.30
```

Ces lignes sont des exemples des restrictions d'accès que vous pouvez mettre en place, non plus en fonction des clients qui se connectent, mais bien en fonction des destinations que ces clients cherchent à joindre (et que vous souhaitez censurer). Vous pouvez ainsi définir des URL ou des morceaux d'URL interdites, des protocoles (FTP), ou encore des adresses IP.

Dans la deuxième ligne, par exemple, on demande à *Squid* d'utiliser la technique des "expressions régulières" Unix (*regex = regular expressions*) pour détecter toutes les URL qui se terminent par **.exe** , **.zip** , **mp3** , etc. (Le $ indique qu'il faut chercher seulement à la fin de la chaîne, le ^ qu'il faut chercher à partir du début, l'anti-slash signifie que le caractère qui suit n'est pas "spécial", l'option **-i** stipule qu'il ne faut pas tenir compte de la casse des caractères).

Note : Si vous souhaitez mettre en place de nombreuses restrictions, il sera peut être préférable d'installer en complément de *Squid* le logiciel redirecteur *Squidguard*.

C) Mise en place des règles d'accès proprement dites :

On commence d'abord par rejeter toutes les connexions interdites :

```
http_access  deny  !explicit          # (ne pas oublier le ! qui signifie "tout,
sauf")
http_access  deny  punis
http_access  deny  telechar
http_access  deny  censure1
http_access  deny  censure2
http access  deny  FTP
```

On continue en acceptant seulement les connexions explicitement autorisées :

```
http_access  allow  admis
```

On termine en rejetant tout le reste :

```
http_access  deny  all
```

Attention : l'ordre de ces instructions est important : *Squid* les lit l'une après l'autre jusqu'à ce qu'il trouve une correspondance. Il applique alors la règle directement, sans lire les lignes suivantes.

Note importante : la requête envoyée à *Squid* par un *browser* ne contient aucune indication du nom de l'utilisateur de ce *browser*. Pour pouvoir identifier cet utilisateur (et éventuellement lui interdire l'accès), *Squid* doit lui-même envoyer une requête à la machine cliente, et il faut donc qu'un logiciel soit mis en place sur celle-ci pour répondre à la demande d'identification. Les clients Window$ qui se connectent doivent avoir lancé le logiciel **identd.exe** , lequel peut assurer le même service d'identification de l'utilisateur que celui pris en charge par le "daemon" **identd** sur les machines Linux. Vous pourriez installer ce logiciel sur chaque poste Window$, mais il est bien plus simple de l'installer dans le partage **netlogon** du serveur, en s'assurant que les PC Window$ soient forcés de l'exécuter au moment où ils se connectent, ce qui peut être obtenu aisément en insérant la ligne suivante dans leur logon script :

```
\\Boss\Netlogon\identd.exe -q
```

D'autre part, il faut s'assurer que *Squid* effectue cette demande d'identification (il ne le fait pas par défaut). Pour ce faire, le fichier /etc/squid.conf doit contenir une ligne telle que :

```
ident_lookup_access  allow  admis
ident_lookup_access  deny all
```

Les deux lignes sont nécessaires pour préciser que ces demandes d'identification ne concernent que les machines de votre réseau local.

D) Accès à l'internet via un fournisseur d'accès qui a lui-même installé un proxy (cas de Profor, fournisseur d'accès de la Communauté Française) :

Activer le 'cascading proxy' avec la ligne :

```
cache_peer 193.191.224.7 parent 3130 no-query
```

et plus loin :

```
never_direct  deny  local_servers
never_direct  allow  all
```

Ces deux dernières lignes sont nécessaires pour que les requêtes que l'on envoie (lorsque l'on remplit un formulaire, par exemple) soient routées correctement.

E) Installation du redirecteur SquidGuard

Installer le paquetage squidgrd (série n).

Dans /etc/squid.conf, remplacer la ligne : **`redirect_program  none`** par :

```
redirect_program  /usr/bin/squidGuard
```

Editer le fichier /etc/squidguard.conf :

```
src  postes_eleves  {
     ip  192.168.0.0/24
}

dest  blacklist  {
     domainlist  blacklist/porn_domains
     urllist  blacklist/sjb_urls
}

acl  {
     postes_eleves  {
          pass  !blacklist  all
          redirect  http://...
}
```

Reconfigurer squid avec la commande : **`squid  -k  reconfigure`**

Il semble que l'on ne puisse mentionner qu'une seule domainlist, et une seule urllist.

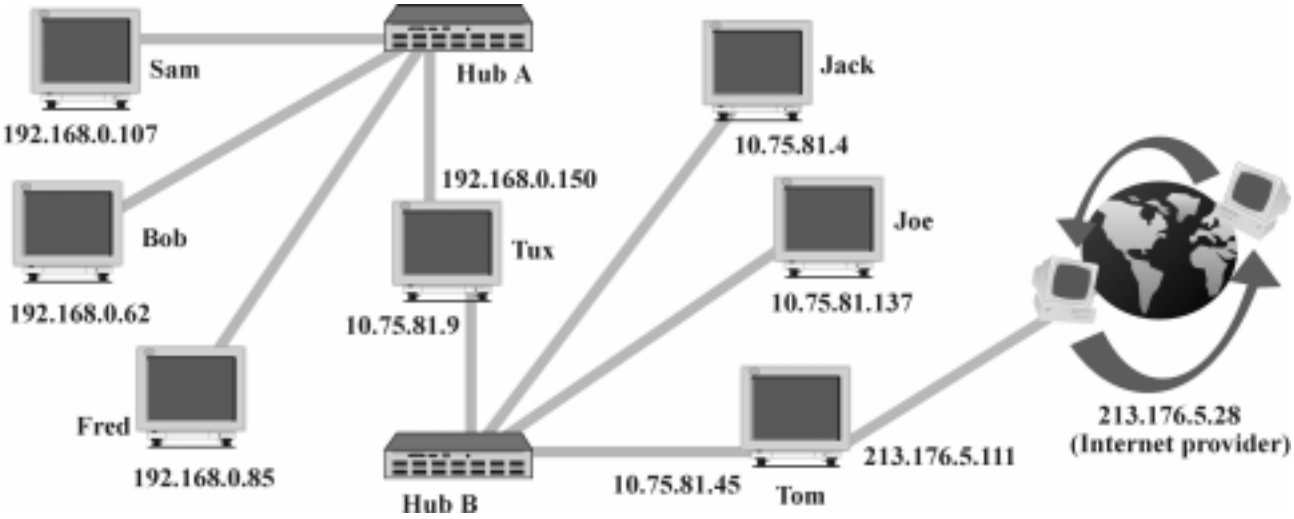
La redirection doit pointer une vraie adresse url http (et non un chemin du genre : file://...)

En cas de problèmes, penser à consulter les fichiers de log (/var/squidGuard/logs)

Ne pas oublier de concéder des droits d'accès sur les fichiers de la blacklist

Installation d'une passerelle pour réunir deux réseaux

Dans la suite de nos explications, nous considérerons deux réseaux **A** et **B** , chacun d'eux constitué de postes divers reliés aux *hubs* A et B. Les machines du réseau **A** (Sam, Bob, Fred et Tux dans notre schéma) possèdent des adresses IP du type 192.168.0.XXX. ; celles du réseau **B** (Jack, Joe, Tom et Tux dans notre schéma) possèdent des adresses du type 10.75.81.XXX (Il s'agit de 2 réseaux de classe C, et toutes les machines utilisent le même masque de sous-réseau 255.255.255.0)



Le poste **Tux**, au centre de notre schéma, fait partie à la fois du réseau **A** et du réseau **B**. C'est cette machine qui constitue la ***passerelle*** entre les deux réseaux, en effectuant le travail que l'on appelle ***routage des paquets IP***. Elle possède deux cartes réseaux, l'une avec une adresse valide dans le réseau **A** (192.168.0.150) et l'autre avec une adresse valide dans le réseau **B** (10.75.81.9).
Par conséquent, toutes les machines du réseau **A** pourront trouver leur passerelle vers le réseau **B** à l'adresse 192.168.0.150, alors que les machines du réseau **B** pourront trouver leur passerelle vers le réseau **A** à l'adresse 10.75.81.9

Préparation de la machine passerelle

Après l'installation physique des deux cartes réseaux, il faut configurer les interfaces *ethernet* correspondantes, **eth0** et **eth1**. Le plus simple est d'utiliser *Yast* :

```
Administration du système → configurer le réseau → configuration de base du réseau
→ eth0
→ adresse IP: 192.168.0.150 ; masque: 255.255.255.0 ; adresse passerelle: - rien -
→ touche F3 → sélectionner l'interface eth1
→ adresse IP: 10.75.81.9 ; masque: 255.255.255.0 ; adresse passerelle: - rien -
```

Attention : ne pas oublier d'**activer** les 2 interfaces (à l'aide de la touche F4)

Remarque : dans la configuration décrite ci-dessus, vous n'indiquez aucune adresse de passerelle par défaut, parce que vous êtes en train de définir la passerelle elle-même.

Les cartes réseaux et les interfaces *ethernet* sont à présent en place. Il reste à activer le service de routage : Toujours sous *Yast* → administration du système → modifier le fichier de configuration : Rechercher et modifier la ligne : **IP_FORWARD = yes**
Cette variable d'environnement permet à *Yast* d'initialiser la machine comme un routeur au prochain démarrage. Si vous ne voulez pas *rebooter*, vous pouvez activer le routage directement à l'aide de la commande (voir plus loin le paragraphe concernant TomsRtBt) :

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

En principe, vous disposez dès à présent d'une passerelle en ordre de marche.

Vous pouvez vérifier à l'aide des commandes **ifconfig** et **route** (sans argument) que tout est bien en ordre (cfr les pages de **man** ou **info** concernant ces commandes).

Tests :

Activez au moins un poste sur chacun des 2 réseaux (par exemple **Bob** sur le réseau **A** et **Jack** sur le réseau **B**) , et essayez des commandes "ping 192.168.0.62" et "ping 10.75.81.4" depuis la machine passerelle vers chacun de ces postes, pour vous assurer que les connexions soient bien établies.

Reconfigurez l'interface réseau du poste **Bob**, à l'aide de *Yast* si cette machine tourne sous *SuSE Linux*, ou à l'aide du panneau de configuration réseau si cette machine fonctionne sous *Window$*, de manière à y définir l'adresse IP d'une passerelle par défaut. Cette adresse devra être 192.168.0.150 (c.à.d. l'adresse de **Tux** dans le réseau **A**).

D'une manière analogue, vous devez reconfigurer l'interface réseau du poste **Jack**, en y indiquant l'adresse 10.75.81.9 comme étant celle de la passerelle par défaut (et c'est effectivement l'adresse de **Tux** dans le réseau **B**).

En principe, tout est en place. Vous pouvez à présent vérifier que la communication est établie, en lançant des commandes "ping" depuis **Bob** vers **Jack**, et vice-versa. Si cela fonctionne, il vous reste à effectuer le même petit travail (c.à.d. définir la passerelle par défaut) sur tous les autres postes de chacun des deux réseaux.

Remarque très importante :

Pour qu'une communication puisse s'établir entre **Bob** et **Jack**, il est ***indispensable*** que ***chacune*** de ces deux machines dispose d'une adresse de passerelle menant vers le réseau de l'autre, car toute communication TCP/IP est toujours un ***dialogue***. Si l'un quelconque des deux partenaires ne "connaît" pas la passerelle lui permettant d'envoyer ses messages à l'autre, ce dialogue est impossible, et la connexion ne s'établit donc pas.

On constate dans ce cas que **Bob** et **Jack** parviennent chacun à "pinguer" vers chacune des 2 adresses IP de la passerelle, mais ils n'arrivent pas à "pinguer" l'un vers l'autre.

Cas particulier du routeur Internet :

Sur notre schéma, la machine **Tom** fait partie du réseau **B**, et elle a été configurée comme routeur vers l'internet pour ce réseau **B** (vraisemblablement à l'aide d'un service Proxy). Lorsque cette machine a établi sa connexion avec un fournisseur d'accès à l'internet, via son interface PPP, SLIP... , une adresse IP a été attribuée dynamiquement à cette interface (par exemple 213.176.5.111) et ainsi **Tom** est devenu partie intégrante d'un autre réseau : celui dont fait également partie le fournisseur d'accès lui-même (213.176.5.28 dans notre schéma exemple). **Tom** est donc bel et bien une passerelle, au même titre que **Tux**.

Si nous souhaitons pouvoir accéder à l'internet depuis une machine du réseau **A** (**Bob**, par exemple), il faut donc -entre autres choses- indiquer à la machine **Tom** l'adresse de la passerelle menant au réseau **A**. Un petit problème apparaît ici : du fait de sa configuration de routeur vers l'internet, **Tom** dispose déjà d'une adresse de passerelle par défaut (c'est l'adresse IP de son *provider*). Il faut donc fournir à **Tom** une deuxième adresse de passerelle (laquelle ne peut plus être définie comme étant une passerelle "par défaut", car ce statut ne peut évidemment être attribué qu'à une seule adresse).

Si **Tom** est une machine Linux, il suffit d'ajouter une ligne à sa table de routage. Le mieux (pour que la modification ne disparaisse pas au *reboot*) est d'insérer une ligne telle que la suivante dans le fichier /etc/route.conf (il faudra ensuite au moins relancer les services réseau, ou redémarrer) :

```
192.168.0.0   10.75.81.9    255.255.255.0 eth0
```

Ceci signifie que tous les paquets destinés au réseau 192.168.0.0 doivent être expédiés via la passerelle située à l'adresse 10.75.81.9

Si **Tom** est une machine fonctionnant sous un autre OS (Window$ NT, MacOS, ...), on devrait pouvoir faire la même chose, mais j'ignore la marche à suivre. De toute façon, il existe un excellent moyen de contourner la difficulté : le camouflage d'adresses IP (également appelé *"IP masquerading"*), que nous allons décrire ci-après.

Cas particulier d'un réseau dont on n'est pas l'administrateur (CCM, Internet ...)

Supposons à présent que nous ne disposons pas de droits d'administration sur le réseau **B** du schéma, et que nous sommes donc dans l'incapacité de (re)configurer la table de routage des machines de ce réseau : impossible donc d'y ajouter l'adresse d'une passerelle menant au réseau **A**. (C'est évidemment le cas des centres cybermédia fournis aux écoles par le ministère).

Qu'à cela ne tienne ! Le monde de l'informatique libre regorge de ressources et de solutions efficaces. En l'occurrence, nous pouvons résoudre notre problème de manière très élégante grâce à un ensemble de fonctionnalités de camouflage d'adresses et de ports TCP/IP qui sont souvent intégrées aux noyaux Linux récents (c'est le cas des noyaux SuSE).

Dans le cas d'un noyau 2.2, (Attention, tout ce qui suit est assez différent pour un noyau 2.4 !) ces fonctionnalités sont paramétrées et contrôlées à l'aide du programme **ipchains** (le paquetage de même nom doit évidemment avoir été installé : voir la série **sec**).

Le camouflage d'adresses IP fonctionne de la manière suivante :

Supposons que la machine **Fred** du réseau **A** veuille communiquer avec la machine **Tom** du réseau **B**. Chaque paquet émis par **Fred** comporte un en-tête qui contient (entre autres choses) l'adresse IP de l'expéditeur (192.168.0.85) et l'adresse IP du destinataire (10.75.81.45). La table de routage de la machine **Fred** doit contenir l'indication d'une passerelle par défaut : la machine **Tux**. Ainsi le paquet sera d'abord expédié à cette machine, puisque l'adresse de destination n'est pas valide dans le réseau **A**.

Sur la machine **Tux**, on a activé les fonctions de routage (*IP-forward*) et de camouflage (*IP-masquerading*) des paquets IP. L'opération de camouflage consiste à traiter chaque paquet émis vers **B** de manière à y remplacer l'adresse de l'expéditeur original (192.168.0.85) par celle de la passerelle elle-même dans le réseau **B** (10.75.81.9). Pour le destinataire final (**Tom** dans notre exemple), le paquet semble donc provenir d'une machine du réseau **B** (en effet : grâce à sa seconde interface, **Tux** fait partie intégrante de ce réseau). Par conséquent, **Tom** n'éprouve aucune difficulté à envoyer en retour sa propre réponse, mais celle-ci va bien évidemment être adressée à **Tux**, puisque de son point de vue, c'est **Tux** l'expéditeur.

Lors de la réception de cette réponse qui ne lui est pas vraiment destinée, **Tux** va maintenant devoir effectuer l'opération inverse du camouflage effectué à l'aller, c.à.d. traiter le paquet-réponse pour y remplacer sa propre adresse par l'adresse de **Fred** (lors du camouflage à l'aller, cette adresse a été mémorisée dans une table interne). Ainsi le paquet-réponse pourra être routé sans problème vers son destinataire réel.

Préparation de la passerelle pour y activer le camouflage

En principe, il faut d'abord s'assurer que le noyau Linux a été compilé avec les options nécessaires (*IP forwarding & IP masquerading*). C'est déjà fait dans le cas des noyaux SuSE. Il faut également s'assurer que le paquetage **ipchains** a été installé (série **sec**).

Il faut ensuite lancer les commandes :

```
echo "1" > /proc/sys/net/ipv4/ip_forward
ipchains -P forward DENY
ipchains -A forward -s 192.168.0.0/24 -d 0.0.0.0/0 -j MASQ
```

La première ligne active le routage (voir TomsRtBt , rubrique suivante).

La seconde annule tout routage préexistant éventuel

La troisième ajoute la directive de routage qui nous intéresse, c.à.d. : Rediriger tous les paquets qui proviennent du réseau A (-s 192.168.0.0/24), envoyés vers une destination quelconque (- d 0.0.0.0/0) en leur appliquant le *masquerading* (-j MASQ)

Remarque : l'indication réseau/masque 192.168.0/24 utilisée ici est tout à fait équivalente à 192.168.0.0/255.255.255.0 Dans ces 2 formulations, on indique en effet que les 24 premiers bits de l'adresse IP désignent le réseau, et les suivants la machine.

Si vous souhaitez que le routage et le camouflage soient activés automatiquement au démarrage de la machine, vous pouvez créer (à l'aide d'un éditeur quelconque) un petit fichier nommé "masquerade"dans le répertoire **/etc/init.d** , lequel contiendra les lignes suivantes :

```
echo -n "Mise en place du routage avec masquerading ..."
echo  1  >  /proc/sys/net/ipv4/ip_forward
ipchains -P forward DENY
ipchains -A forward -s 192.168.0.0/24 -d 0.0.0.0/0 -j MASQ
echo "done."
```

Vous en modifiez ensuite les permissions d'accès, puis vous installez des liens symboliques vers ce fichier dans les sous-répertoires de /etv/rc.d/init.d qui correspondent au niveaux d'exécution pour lesquels vous souhaitez que ce routage soit actif. Exemple :

```
cd  /etc/init.d
chmod  744  masquerade
cd  rc3.d                                   # ceci pour le run level 3, bien sûr
ln  -s  ../masquerade  s22masquerade
```

(Une autre possiblité est d'insérer une ligne d'instruction pour lancer votre petit script *masquerade*, au bon endroit dans le script /etc/init.d/rc)

Réalisation d'une passerelle à l'aide de TomsRtBt

Une micro-distribution de Linux telle que *TomsRtBT*, qui tient toute entière sur une seule disquette, suffit largement pour réaliser une passerelle. Il est donc possible de réaliser une passerelle à l'aide d'un vieil ordinateur (un PC 386 avec 8 Mb de mémoire suffit, même si son disque dur est "out" !).

Installer les deux cartes réseau dans l'ordinateur, puis lancer les commandes suivantes (en adaptant les adresses IP à votre situation, bien entendu) :

```
insmod rtl8139                      (Installation du (des) pilote(s) éventuel(s) pour les cartes réseau)
ifconfig eth0 192.168.0.150
ifconfig eth1 10.75.81.9
route add -net 192.168.0.0 netmask 255.255.255.0
route add -net 10.75.81.0 netmask 255.255.255.0
echo "1" > /proc/sys/net/ipv4/ip_forward
```

La dernière ligne active la fonction de routage. A ce propos, sachez que cette commande peut être utilisée pour activer le routage sur n'importe quelle version de Linux, à la condition toutefois que la fonction de routage ait été implémentée dans le noyau lors de la compilation de celui-ci. (C'est heureusement le cas pour la plupart des distributions "standard" de Linux).

Cette commande change simplement la valeur d'une variable d'environnement système. Vous pouvez désactiver/réactiver le routage à volonté à l'aide de commandes telles que :

```
echo "0" > /proc/sys/net/ipv4/ip_forward
echo "1" > /proc/sys/net/ipv4/ip_forward
```

De même, vous pouvez savoir si le routage est actif ou non sur une machine quelconque, en visualisant la même variable système, par exemple avec :

```
more /proc/sys/net/ipv4/ip_forward
```

Si vous obtenez 1, c'est que le routage est activé. Vous obtenez 0 sinon. Si vous obtenez un message

d'erreur, c'est que le noyau ne comporte pas la fonctionnalité de routage.

Si tout fonctionne correctement, il vous reste à personnaliser votre disquette *TomsRtBt* pour y ajouter un script qui puisse lancer automatiquement la série de commandes décrite ci-dessus.

Affichage automatique d'informations sur le bureau de l'utilisateur

Comme cela nous a été suggéré lors d'une discussion avec d'autres administrateurs système, nous avons développé un script et une procédure permettant de faire apparaître automatiquement diverses informations directement sur le bureau de l'élève qui se connecte (par exemple en incrustation dans l'image utilisée comme "papier peint").

Au stade actuel du script, les informations affichées sont le nom de l'élève (en clair) et la situation de ses quotas disque. Si vous le souhaitez, vous pourrez aisément modifier ce script pour lui faire afficher autre chose.

Principe :

Si Samba et les PC clients Window$ ont été configurés comme nous l'avons suggéré plus haut, un *logon script* est mis en oeuvre à chaque fois qu'un élève se connecte. Ce script est un fichier de type "BAT" situé dans le répertoire partagé */home/netlogon* du serveur[3], et il correspond au nom du groupe-classe suivi de l'extension **.bat** (Pour rappel, la mise en oeuvre de tout ceci est déterminée dans le fichier de configuration */etc/smb.conf*).

Afin de nous assurer que certaines opérations soient effectuées systématiquement pour tout utilisateur qui se connecte, nous allons veiller à ce que la première ligne du fichier *.bat de chaque groupe contienne toujours l'instruction :

```
@CALL Commun.BAT          ( => appel d'un autre script)
```

Dans ce script *Commun.bat* (que nous placerons lui aussi dans */home/netlogon*), on définit les opérations qui doivent être accomplies par toute machine Window$ qui se connecte, quel que soit son utilisateur (et le groupe-classe correspondant). Parmi ces opérations, nous allons inclure la définition d'au moins un "lecteur réseau". Exemple :

```
NET USE T: \\SERVEUR\Scala
```

Comme vous le savez maintenant fort bien, ce partage doit évidemment avoir été défini dans le fichier de configuration de Samba (fichier */etc/smb.conf*).

Or il faut savoir que Samba est doté d'un dispositif qui permet de lancer automatiquement une application quelconque (sur le serveur ou il est installé), lorsque l'un des postes de travail demande l'accès à un partage quelconque . Il suffit pour cela d'insérer dans la définition de ce partage une ligne telle que :

```
preexec nom_de_l'application_à_exécuter
```

Comme nous l'avons expliqué ci-dessus, nous pouvons nous assurer que certains partages soient toujours "visités" au moment de la connexion (partage [Scala] dans notre exemple) en activant des lecteurs réseau qui pointent vers ces partages, dans un *logon script* commun. Nous disposons donc à présent d'un mécanisme pour lancer une application de notre choix sur le serveur, à chaque fois qu'un utilisateur se connecte. En l'occurrence, l'application que nous allons faire exécuter ainsi est un script *Python* nommé ***backprint.py***

Dans le fichier */etc/smb.conf*, il faut donc repérer la description du partage [Scala], et ajouter à la définition de ce partage une ligne telle que :

root preexec /bin/backprint.py %U %m

3 Rappel : ce répertoire du serveur apparaît dans le "voisinage réseau" des PC Window$ qui se connectent, sous le nom de partage "Netlogon"

L'expression "**root preexec**" précise que l'application devra être exécutée avec des droits d'administrateur. Et comme vous l'aurez certainement compris à la lecture de cette instruction, le script *backprint.py* sera lui-même installé dans le répertoire **/bin** du serveur.

Au démarrage de ce script, Samba lui fournira en outre deux paramètres : l'identifiant sous lequel l'utilisateur s'est connecté, grâce à la variable Samba **%U**, et le nom *Netbios* de son poste de travail, grâce à la variable Samba **%m**.

A partir de ces deux informations, *backprint.py* peut rechercher dans le fichier */etc/passwd* du serveur le nom complet de l'utilisateur et le chemin de son répertoire personnel. Il se charge alors de mémoriser dans un fichier journal, l'identité de la personne qui s'est connectée ainsi que la date et l'heure exactes de cette connexion. Il lance ensuite la commande **quota** pour déterminer la situation de cet utilisateur en matière de quotas d'espace disque , et se charge de placer cette information à deux endroits : dans un petit fichier texte envoyé directement dans le répertoire personnel de l'utilisateur, et en incrustation dans l'image qui sert de "bureau" (fond d'écran) à cet utilisateur (Ceci ne fonctionne toutefois que si la librairie PIL (*Python imaging library*)a été installée sur le serveur.

En fait, le script crée automatiquement une image de fond nommée *background* dans le répertoire personnel de l'utilisateur Si vous souhaitez que le texte soit incrusté dans une image préexistante, placez celle-ci dans le répertoire de la classe, et donnez-lui le nom *papierpeint*. (Si cette image n'existe pas, le texte sera incrusté dans un fond gris uniforme).

Installez donc le script *backprint.py* dans le répertoire /bin du serveur, ainsi que les fichiers de polices qui l'accompagnent. N'oubliez pas de rendre ce script exécutable, par exemple en entrant :

```
chmod  755  backprint.py
```

Vous pouvez en outre installer aussi le script complémentaire *mouchard.py* dans un répertoire de votre choix, par exemple dans /root (répertoire réservé à l'administrateur), ou encore dans /bin (ce répertoire fait en effet partie du chemin par défaut (variable PATH), ainsi les logiciels/scripts qu'on y place peuvent être lancés de partout).

N'oubliez pas de rendre ce script exécutable, lui aussi.

backprint.py mémorisera les connexions dans un fichier journal nommé *userlogons* (qui sera créé automatiquement), dans le répertoire /var/log/samba (Vérifiez l'existence de ce répertoire, et au besoin créez-le).

Le script *mouchard.py* vous permet d'analyser ce fichier de manière confortable. Si vous avez installé le service proxy Squid sur votre serveur, il vous permet en outre de rechercher n'importe quelle chaîne de caractères dans les URL visitées par les élèves sur l'internet, et de retrouver qui a effectué cette connexion. Si votre proxy est autre (situé sur un serveur NT, par exemple), vous ne pourrez malheureusement pas tirer profiter de cette fonctionnalité.

Si vous ne souhaitez pas utiliser mes scripts, vous pouvez aussi simplement inclure la ligne suivante dans /etc/smb.conf, sous la définition d'un partage "toujours" visité parce que connecté à un lecteur réseau (tel [Scala] dans mon exemple)[4].

---

4 Vous pouvez utiliser aussi le partage [Profiles]. En principe, ce partage est toujours visité au moment de la connexion (il contient la mémorisation des profils itinérants de Window$). Il faut alors veiller à ce que les lignes suivantes soient définies dans la section globale de /etc/smb.conf :

```
domain logons = yes
logon path = \\%L\Profiles\%U
```

Veillez aussi à définir le partage "Profiles" lui-même :

```
[Profiles]
  root preexec /bin/backprint.py %U %m
  path =  /home/profiles
  etc.
```

(Si vous ne souhaitez pas que les profils itinérants soient réellement mémorisés (c'est mon cas), rien ne vous

```
root  preexec  = echo "%T : poste %m Utilisateur %U" >>
                                                  /var/log/samba/userlogons
```

On utilise simplement ici la commande echo, dont on redirige la sortie vers un fichier journal.

Configuration d'un ordinateur "bastion" (passerelle⇔ Internet)

Installation de la carte RNIS avec les outils SuSE :

PC utilisé : 486 avec 24 Mb de RAM (il m'a été impossible d'installer les modules I4L sur un PC ne comportant que 8 Mb de RAM : j'en déduis qu'il faut un minimum de 16 Mb).

Si la carte RNIS utilisée était une PCI, elle serait détectée automatiquement (en principe).
La carte utilisée ici est une ISA PNP (il n'y a pas de slots PCI sur les vieux 486 !!)
Il faut donc dans ce cas installer le paquetage isapnp (série ap)
puis lancer la commande :

```
pnpdump -r > /etc/isapnp.conf
```

A la suite de cette commande, le programme pnpdump analyse la machine, détecte les cartes ISA PNP présentes et stocke l'information trouvée dans le fichier cible /etc/isapnp.conf

On peut ensuite éditer ce fichier /etc/isapnp.conf, notamment pour y spécifier une autre irq et/ou une autre plage d'adresse i/o que celles établies par défaut. Dans le cas de la carte Eicon Diva 2.01 utilisée, j'ai choisi l'irq 7 et la plage d'adresses i/o 0x220. Ne pas oublier la ligne ACT = Y

(Au redémarrage de la machine, la carte devrait être installée automatiquement)

Configuration du système avec YAST

→ Administration du système → Intégrer le matériel dans le système → Configurer le matériel ISDN → [x] Démarrer i4l, Euro-ISDN(Edss1), Eicon Diva 2.01 ISA, IRQ 7, E/S = 220
→ Démarrer → Si ça marche → Mémoriser.

→ Administration du système → Configurer le réseau → Configuration de base du réseau →
→ Sélectionner la 2e ligne (la 1e concerne l'interface eth0) → F5 pour choisir un périphérique réseau ISDN SyncPPP → F6 → Adresse IP de votre machine : 192.168.0.99 (obligatoirement),
[x] adresse IP dynamique, masque 255.255.255.255 (obligatoirement), adresse de la passerelle par défaut = 192.168.0.1 (obligatoirement), adresse de l'autre machine point à point = 192.168.0.1 (obligatoirement) → F4 (activer) → F10 (mémoriser) →
Retourner à "Configuration du matériel ISDN" → Paramètres ISDN →

Votre n° local (MSN) : entrer le n° de téléphone attribué à la ligne RNIS (ex: 042540741)
N° à appeler : entrer le n° de téléphone du fournisseur d'accès Internet
[x] Seulement numéros spécifiés autorisé
[x] Numérotation auto (si l'on souhaite cet automatisme)
période d'inactivité : 120
n. de tentatives d'appel : 2 Serveurs DNS : par ex. 212.166.2.11 – 212.166.2.25
config callback ISDN : off
Nom login ppp : entrer l'identifiant fourni par le provider
Mot de passe du login ppp : entrer le mot de passe → Mémoriser

Quitter YAST

Editer le fichier **/etc/isdn/isdn.conf** :

COUNTRYCODE = 32
AREACODE = 4 (pour Liège) Ne pas mettre de zéros.

---

empêche d'indiquer à la ligne "path" ci-dessus un répertoire qui n'existe pas !)

Rebooter.

On peut vérifier si ça marche en essayant un ping vers une adresse internet connue, par exemple les adresses IP des serveurs DNS

On peut tout de suite surfer sur le Web en mode texte (ce qui ne manque pas d'intérêt : c'est ultra-rapide !) à la condition d'avoir installé aussi le logiciel **lynx** (voir paquetage du même nom) :

```
lynx  hhtp://www.google.com
```

Debug :

Lancer la commande **ifconfig** pour afficher l'état des interfaces réseau.

On peut aussi lancer **netstat** avec les options -M , -i ou -s (voir **man netstat**)

La commande **isdnctrl** (dont vous pouvez trouver la documentation complète dans les pages de man) permet de mettre en route, d'arrêter, de surveiller la connexion établie.

Exemple :

```
isdnctrl dial ippp0          => établit la connexion
isdnctrl status ippp0        => indique si la connexion est établie ou non
isdnctrl hangup ippp0        => coupe la connexion
isdnctrl secure ippp0
isdnctrl list all            => montre les paramètres actuels
etc.
```

Considérons par exemple que la connexion refuse de se couper automatiquement lorsque les consultations web ont cessé (le cas se produit par exemple si votre fournisseur d'accès vous envoie sans cesse des petits paquets de contrôle toutes les x secondes). Pour résoudre ce problème, on peut (à l'aide d'un éditeur quelconque) créer un fichier **/bin/isdnstop** qui contient les lignes suivantes :

```
#!/bin/sh
isdnctrl  hangup  ippp0
```

et appeler ce script périodiquement à l'aide du service **cron**. Ainsi vous serez assuré que la connexion internet sera interrompue systématiquement toutes les 15 minutes, par exemple, si vous ajoutez la ligne suivante dans le fichier /etc/crontab :

```
2,17,32,47  *  *  *  *  root  /bin/isdnstop
```

On doit encore configurer cet ordinateur bastion comme passerelle effectuant le *masquerading*.

Pour ce faire, voir page 40 : "Mise en place d'une passerelle avec masquerading".

Le fichier **/etc/rc.d/init.d/masquerade** contiendra cependant :

```
echo -n "Mise en place du routage avec masquerading ..."
echo  1  >  /proc/sys/net/ipv4/ip_forward
ipchains -P forward DENY
ipchains -A forward -s 192.168.0.100 -i ippp0 -j MASQ
echo "done."
```

L'avant-dernière ligne indique que l'on masque vers l'interface ippp0, et que la seule machine autorisée à utiliser la passerelle est la machine 192.168.0.100 (celle ou se trouve le proxy).

Pour sécuriser la passerelle, il faut y supprimer tous les services inutiles (ne conserver éventuellement que **ssh** (pour pouvoir agir sur cette machine à distance)

****** à compléter ****** (entre autres : désactiver le service inetd)

Installation d'un service NIS (ou YP)

Ce service permet de centraliser les informations concernant les comptes utilisateur pour les machines clientes fonctionnant sous Linux.

Installer les paquetages **ypserv** et **ypbind** (sur le serveur) et **ypbind** (sur les postes clients).

A l'aide de ***yast***, éditer le fichier de configuration générale **rc.config**

Sur le serveur et les clients, il faut que la variable YP_DOMAINNAME contienne la même chose (Ce nom de domaine YP peut être choisi arbitrairement)

Sur les postes clients, il faut activer le démon **ypbind** (START_YPBIND = yes), et désigner le ou les serveurs YP présent(s) (YP_SERVER = 192.168.0.100 192.168.0.101)

Il faut également veiller à inclure dans leur fichier **/etc/fstab** une ligne telle que :

```
192.168.0.100:/home   /home   nfs   defaults  0  0
```

afin que ces clients effectuent automatiquement un montage NFS du répertoire **/home** du serveur sur leur point de montage **/home** local. Ainsi les répertoires personnels de chacun (situés sur le serveur) sont rendus accessibles à l'endroit habituel dans la hiérarchie locale des répertoires

Sur le serveur, il faudra bien entendu mettre en route les démons **ypserv** et **yppasswd** :

(START_YPSERV = yes , START_YPPASSWDD = yes , START_YPBIND = no).

Dans le fichier **/etc/exports** , prévoir une ligne telle que :

```
/home    *(rw)
```

afin que le répertoire personnel des utilisateurs soit accessible via NFS.

Redémarrer les services réseau, puis créer les tables (maps) de la base de données. Pour ce faire, il faut se placer dans le répertoire **/var/yp** et lancer la commande : **make** (après avoir éventuellement édité le fichier **/var/yp/Makefile** pour établir certaines options : changer notamment MinUID = 500, puisque les ID des utilisateurs "normaux" commencent à 500).

Afin que des mises à jour de la base de données NIS soient effectuées régulièrement, il faudra aussi placer la ligne suivante dans **/etc/crontab** :

```
*/10 * * * * root make -s -C /var/yp
```

(ou bien lancer de temps en temps "à la main" la commande : `make -s -C /var/yp`)

On peut en outre mettre en place un ou plusieurs serveurs "esclaves" (voir doc.)

Pour vérifier si cela fonctionne, essayer sur un poste client la commande : **ypwhich**
qui doit vous retourner le nom du serveur NIS actif

ypwhich -m doit retourner la liste des tables (maps) accessibles
ypcat <nom_d'une_table> doit permettre de visualiser le contenu de la table choisie

Vérifiez qu'il existe bien un "**+**" à la fin de vos fichiers **/etc/passwd** et **/etc/group**

Vous pouvez utiliser le serveur NIS pour la résolution de noms (exportation du fichier **/etc/hosts** du serveur). Pour ce faire, ajoutez une ligne "**order hosts nis bind**" dans votre fichier **/etc/host.conf**

NIS n'autorise pas le *login* en mode graphique directement. Il faut se connecter d'abord en mode texte. On lance ensuite le serveur graphique ensuite à l'aide de la commande **startx**

Se connecter à distance sur un autre poste Linux

En mode texte, il suffit d'utiliser la commande **telnet** , ou mieux encore la commande **ssh** suivie du nom ou de l'adresse IP de la machine distante. C'est tout (à la condition toutefois que ces services aient été installés, ce qui est généralement le cas). Ça marche aussi avec l'internet (pour autant que la machine distante accepte la connexion). Il vous faudra bien évidemment posséder un compte sur la machine distante, qui vous demandera de vous identifier.

En mode graphique c'est un tout petit peu plus compliqué.

Le mode graphique étant actif sur la machine cliente, commencez par y autoriser l'affichage des images qui proviendront du serveur, à l'aide de la commande (dans une fenêtre de terminal) :

xhost nom_de_la_machine , ou encore :
xhost 192.168.0.100

Vous autorisez ainsi la machine désignée à accéder à votre serveur graphique (en d'autres termes, vous l'autorisez à vous expédier des images).

Ensuite, connectez-vous par **telnet** ou **ssh** à la machine distante :

ssh 192.168.0.100 ou bien :
ssh jules@192.168.0.100 (si vous voulez vous connecter avec une identité particulière)

Entrez le mot de passe demandé. C'est prêt. Il vous suffit maintenant de lancer n'importe quel logiciel graphique en entrant son nom (toujours dans la même fenêtre de terminal).

Lorsque vous avez terminé, il faut penser à entrer la commande :

xhost -

afin de réinterdire l'accès de votre serveur graphique à toute autre machine.

Note importante :

Si vous voulez utiliser **ssh** pour une connexion en mode graphique comme expliqué ci-dessus, il faut d'abord éditer le fichier **/etc/ssh_config** de votre machine, pour y activer la ligne :
ForwardX11 = yes

Sinon votre machine refusera l'accès de votre serveur graphique à la machine distante.

Connexion FTP

Ce mode de connexion à distance permet d'échanger facilement des fichiers, même si le service NFS n'est pas activé.

Pour se connecter à un serveur (y compris sur internet), il suffit d'entrer une commande telle que :

ftp <nom ou adresse IP du serveur> (il faudra bien entendu s'identifier)

cd <répertoire>	change de répertoire sur le serveur
pwd	indique dans quel répertoire on se trouve (sur le serveur)
binary	passe au mode de transfert binaire
lcd	change de répertoire chez le client
get <fichier>	transférer un fichier du serveur au client (download)
put <fichier>	expédier un fichier vers le serveur (upload)
quit	terminer la session ftp

(Voir la description des autres commandes dans la littérature)

Configuration de Apache

Editer le fichier ***/etc/httpd/httpd.conf***

Il faut surtout y indiquer quel(s) répertoire(s) sont accessibles .

Le répertoire par défaut pour les pages html dans la distribution SuSE est */usr/local/httpd/htdocs.* Pour en choisir un autre, il faut éditer la ligne :

```
DocumentRoot  "/usr/local/httpd/htdocs"
```

en remplaçant le chemin indiqué par un autre de votre choix (par ex. */home/siteweb*).

Les répertoires auxquels on souhaite que Apache puisse accéder doivent être signalés, dans des descriptions telles que celle-ci :

```
<Directory  "/home/siteweb">
     Options Indexes +FollowSymLinks +Includes Multiviews
     AllowOverride None
     Allow from all
</Directory>
```

Méthode encore plus simple :
Si vous ne souhaitez pas changer le répertoire racine déjà établi par SuSE, vous pouvez tout simplement y ajouter un lien symbolique pointant vers le répertoire de votre choix :

```
cd  /usr/local/httpd/htdocs
ln  -sn  /home/siteweb  siteweb
```

Il faudra cependant veiller alors à ce que la description de ce répertoire racine , dans le fichier */etc/httpd/httpd.conf* , comporte bien l'option +FollowSymLinks (-FollowSymLinks par défaut).

Ne pas oublier de relancer Apache à l'aide de la commande :

```
rcapache restart
```

ou bien :

```
/etc/init.d/apache restart
```

Installation de Linux sur les postes de travail élèves

Réaliser une installation type aussi complète que possible sur une machine quelconque. Cette configuration doit inclure les clients NIS et NFS.

Note : *KDE* est une interface très, très "gourmande". On peut s'en passer sans problème et se contenter de *WindowMaker* qui est assez chouette, occupe beaucoup moins d'espace disque et est nettement plus rapide. Avec *SuSE 7.2*, choisir l'installation minimale avec X11 mais sans KDE. Ajouter les paquetages : mc, joe, identd (série ap), nfsutils, ypbind (série n), nedit, netscape (série xap), python, python-imaging (série d).

Sauvegarder sur disquette le choix des paquetages retenus au cours de l'installation (Yast).

Sur les machines cibles, il suffira d'installer :
Source d'installation → CDROM
Déterminer/démarrer l'installation
Charger la configuration → frapper F9 (Disquette) →
Cocher la case correspondant au fichier de mémorisation sur la disquette → Ajouter
Démarrer l'installation → laisser Yast installer les paquetages.

Ensuite (toujours sous Yast, sur chaque machine) :

Installation de la carte réseau, nom d'hôte, adresse IP, etc. →
Administration système → Modifier le fichier de configuration →
YP_DOMAINNAME = CFORM (Nom choisi pour le domaine NIS)
YP_SERVER = 192.168.0.100 (adresse IP du serveur NIS)
START YPBIND = yes

Ne pas oublier d'éditer le fichier /etc/fstab pour y ajouter une ligne telle que :

```
192.168.0.100:/home   /home   nfs   defaults  0  0
```

afin que ces clients effectuent automatiquement un montage NFS du répertoire **/home** du serveur sur leur point de montage **/home** local. Ainsi les répertoires personnels de chacun seront accessibles à l'endroit habituel dans la hiérarchie des répertoires

Rebooter.

Les élèves peuvent maintenant se connecter avec leur identifiant et leur mot de passe habituels. Ils trouveront leurs données personnelles dans un sous-répertoire de /home

Installation du SGBD MySQL (SuSE 7.2)

MySQL est un système de gestion de bases de données (SGBD) très puissant, dont on peut installer la partie serveur sur un ordinateur central, et la partie cliente sur de nombreux postes de travail (Linux ou Window$). On pourra ainsi disposer d'un vrai système de base de données centralisé pour l'intranet de l'école.

Toutes les opérations d'installation décrites ci-dessous sont effectuées en tant que root :

- Installer les 3 paquetages (série ap)
- Lancer le script de démarrage prévu par SuSE, à l'aide de la commande :
  rcmysql start
  La première fois qu'elle est lancée, cette commande crée un sous-répertoire **mysql** dans **/var/lib**, et y installe toute une série de choses. C'est dans ce répertoire que se créeront les bases de données gérées par **MySQL**. (En cas de gros problème nécessitant une réinstallation, il suffira d'ailleurs d'effacer ce répertoire et son contenu).
- Vérifier que le serveur fonctionne, par exemple en entrant la commande :
  `mysqladmin version (-> affichage d'infos diverses)`
- Vérifier ensuite si l'on peut arrêter et redémarrer le serveur à volonté, à l'aide des commandes :
  `rcmysql stop`
  `rcmysql start`
- Créer le mot de passe de l'administrateur (une seule fois !) :
  `mysqladmin -u root password xxxxxx`
- Si l'on souhaite que le serveur soit automatiquement mis en marche à chaque redémarrage du système, lancer Yast → Administration du système → Modifier le fichier de configuration → **START_MYSQL = yes**.

Tout est désormais en place.

Pour accéder à une base de données sur le serveur lui-même, on utilise le logiciel client ***mysql***.
Par exemple, pour accéder à la base de données d'administration (laquelle s'appelle tout simplement *mysql*), le patron (*root*) pourra se connecter à l'aide d'une commande telle que :

```
mysql -u root mysql -p          (-> son mot de passe lui sera demandé)
```

L'une des premières tâches d'administration à accomplir consistera à désigner les utilisateurs autorisés à se connecter, avec quels droits précis. Tout se gère à l'aide de commandes SQL.
Il est bon de désigner tout de suite un utilisateur particulier (autre que *root*) qui sera l'administrateur effectif de *MySQL* pour la suite des opérations (on évitera ainsi de devoir fréquemment se connecter en tant que *root*). Voici comment désiger cet utilisateur (on suppose ici que l'administrateur *root* est toujours connecté à la base de données ***mysql*** - le nom choisi pour l'utilisateur est quelconque : il ne doit pas nécessairement exister dans le système Linux au préalable) :

```
grant all privileges on *.* to freddy@localhost identified by 'abcde'
      with grant option;
grant all privileges on *.* to freddy@"%" identified by 'abcde' with grant option;
```

Explication : *.* désigne toutes les tables de toutes les bases de données – freddy@localhost indique que l'utilisateur peut se connecter directement sur la machine serveur – freddy@"%" indique qu'il peut aussi se connecter depuis n'importe quel poste (les 2 instructions sont nécessaires).
On peut dès à présent se déconnecter, puis se reconnecter en tant que "freddy" pour effectuer la suite des opérations :

```
\q                              (fermer la connexion en tant que 'root')
mysql -u freddy mysql -p        (se reconnecter en tant que 'freddy')

grant select, insert, update, create (-> liste des droits octroyés)
          on banksys.*                (toutes les tables de la base banksys)
          to frank@"%.CFORM"          (frank ne peut se connecter qu'à
          identified by 'stupid' ;     partir d'un poste du domaine CFORM)
```

Note : toutes les informations ainsi introduites se trouveront dans la base ***mysql***. Il sera donc facile (pour l'administrateur) de les retrouver et de les modifier éventuellement par la suite.

Création d'une base de données intitulée "club" par l'utilisateur autorisé "freddy" :
(Attention : les commandes SQL sont insensibles à la casse, mais il faut faire attention à celle-ci pour le nom des bases de données, car il s'agit en fait de noms de fichiers) :

```
mysql -u freddy -p             (Connexion initiale : le mot de passe est requis)
create database club;
use club;                      (Note : point-virgule facultatif pour cette commande)
create table membres
     (ref int auto_increment not null, nom varchar(20), prenom varchar(20),
      phone bigint default 0, d_naiss date, primary key (ref),
     unique(nom, prenom));
insert into membres
     (nom, prenom, phone, d_naiss)
     values ('Dupont', 'Charles', 78541369, '1965-5-21');

     ... etc ...

update membres set nom ='Dupond' where nom ='Dupont';
select * from membres order by nom;
show tables;

etc, etc ...
```

On peut accéder à une base de données *MySQL* au départ de n'importe quel poste de travail, à l'aide de clients appropriés. Sous Linux, on peut évidemment utiliser le client **mysql**. Exemple :

```
mysql  -h  192.168.0.125  -D  club  -u  freddy  -p
```

Sur un poste de travail Window$, on peut aussi utiliser un client *mysql* en mode texte. Un meilleur choix consiste à utiliser plutôt le client MySQLGUI.exe (disponible sur le site internet de *MySQL*), qu'il n'est pas nécessaire d'installer sur chaque poste (il suffit qu'il soit accessible sur un serveur de fichiers).
On peut aussi installer un pilote *ODBC* (également disponible sur le site internet de *MySQL*).
Il devient alors possible d'accéder à la base de données en utilisant des outils de bureautique classique, tels M$-Access ou StarOffice.
Attention : les tables de la base de données doivent disposer chacune d'au moins une clé primaire (primary key), sinon l'accès ODBC à ces tables n'aura lieu qu'en lecture seule.

Exemple d'instructions pour ajouter une clé primaire à une table existante :

```
alter table membres change nomvarchar(50) not null;
alter table membres add primary key (nom);
```

Note : avec M$-Access, il faut cocher l'option "Return matching rows" en activant le pilote.
Il est également parfaitement possible d'accéder à un serveur *MySQL* distant ou local à l'aide d'un programme Python, que celui-ci fonctionne sous Linux ou sous Window$ sur la machine cliente (voir notes de cours sur Python).
Backups :
La technique la plus simple consiste à sauvegarder l'intégralité du répertoire /var/lib/mysql.
Sauvegarde :

```
rcmysql stop                          (On commence par arrêter le service)
cd  /var/lib/mysql
tar  -cvzpf  /ggg/mysqldump.tgz  *
```

Restauration :

```
cd  /var/lib/mysql
rm  -rf  *                    (On efface tout. Prudence !!!)
rcmysql  start                (Reconstruction d'une nouvelle structure)
tar  -xvzpf  /ggg/mysqldump.tgz
rcmysql  stop                 (Enregistrement de la structure modifiée)
rcmysql  start                => Tout devrait fonctionner à nouveau
```

Accès à MySQL depuis Python (le paq. python-mysql doit avoir été installé !) :

Les modules devraient se trouver dans /usr/lib/python/site-packages/
Exemple d'accès à la base de données "club" :

```
import MySQLdb
db = MySQLdb.Connect(user="freddy", passwd="klich", db="club"
                        [, host="192.168.0.100", port=.. , unix_socket=..)

# informations diverses :
print db.get_host_info()         # info machine serveur
print db.get_server_info()       # info logiciel serveur MySQL (version)
print db.stat()                  # activité du serveur
print db.affected_rows()         # nombre d'enregistrements

# création du curseur :
cursor = db.cursor()

# affichage des champs présents dans la table active :
for cursorFieldName in cursor.desription:
    print cursorFieldName

# requête type :
ref = cursor.execute("SELECT * FROM membres")
# ref contiendra le n° de référence du dernier enreg. trouvé

# autre exemple avec une variable python :
max_price = 5
cursor.execute("""SELECT spam, eggs, sausage FROM breakfast
            WHERE price < %s""", (max_price,))

# pourquoi un tuple pour s% ?
# parce que les paramètres attendus par la DB API sont des séquences

# transfert du résultat de la requête de sélection dans un tuple de tuples :
res = cursor.fetchall()

# tables accessibles (c'est une requête comme une autre) :
cursor.execute("show tables")
print cursor.fetchall()          # il s'agira d'un tuple

# fermetures :
db.commit()        # termine la transaction
db.rollback()      # annule la transaction (pas toujours possible)
cursor.close()
db.close()

Fetch row as dictionary :
not standard; alternate cursor class DictCursorprovides a dictionary
interface,keys are "column" or "table.column" if there are two columnswith the
same name; use SQL AS to rename fields.
```

Installation de PostgreSQL (SuSE 7.2) :

PostgreSQL est un SGDB (système de gestion de bases de données) très puissant *(et libre !).*

- Installer les paquetages postgresql, postgresql-odbc, postgresql-python, postgresql-server (série ap), pygresql (série d) *Note : les binaires s'installeront d'eux-mêmes dans /usr/bin*
- Sous Yast : → Administration du système → Modifier le fichier de configuration → **`START_POSTGRES = Yes`** (Ainsi le démon *postmaster* sera mis en route automatiquement à chaque démarrage système)
- Sous *root*, changer le mot de passe de l'utilisateur *postgres* (lequel a été créé automatiquement durant l'installation), de manière à pouvoir se connecter sous ce compte.
  Vérifier dans /etc/passwd que le rép. personnel de l'utilisateur postgres soit bien /var/lib/pgsql
- Entrer la commande : **`/etc/init.d/postgresql start`** → démarrage du service
- Se connecter en tant que *postgres* : cet utilisateur système est le "patron" du SGBD
- Entrer la commande : **`createuser -P`** → création de comptes utilisateurs pour le SGDB
  Le paramètre -P est nécessaire si vous souhaitez que des mots de passe spécifiques à PostgreSQL soient créés pour ces utilisateurs. Ces mots de passe seront sauvegardés dans la table /var/lib/pgsql/pg_shadow . Penser à créer un utilisateur wwwrun si vous envisagez de gérer vos bases de données via des scripts CGI associés à des pages Web (il semble qu'il faille donner à cet utilisateur les droits de création de bases de données ?bug?)
  Modifier le fichier */var/lib/pgsql/pg_hba.conf* → définition de divers droits d'accès. Si vous avez créé des mots de passe PostgreSQL pour vos utilisateurs, vous pourrez par exemple utiliser les 2 lignes de configuration suivantes :

```
local    all                                  crypt
host     all  192.168.0.0  255.255.255.0      crypt
```

  N'activez ceci qu'après avoir créé les utilisateurs et leurs mots de passe. Si vous avez du mal à reprendre le contrôle en tant que root, restaurez les options d'origine pour l'accès local ('trust' au lieu de 'crypt' dans la première ligne ci-dessus).
- Connectez-vous à présent en utilisateur quelconque. Pour créer une nouvelle base de données, il vous suffira d'entrer la commande :

```
createdb -U utilisateur nom_d'une_base_de_données
```

  Le nom d'utilisateur fourni doit être celui d'un utilisateur créé avec *createuser*. Son mot de passe vous sera demandé. La base de données ainsi créée est la propriété de cet utilisateur. C'est lui seul qui pourra définir les droits des autres utilisateurs sur cette base de données (à l'aide de la requête GRANT - voir plus loin)
- La méthode expliquée ci-dessus ne permet pas de gérer des droits d'accès indépendamment pour chaque base de données (chaque utilisateur défini comme ayant tous les droits sur une base de données possède les mêmes droits sur toutes les autres). De plus, le système de cryptage des mots de passe ne fonctionne pas pour des clients qui se connectent via *ODBC*.
  Voici donc une méthode plus élaborée :
  Pour chaque base de données indépendante, créez un fichier d'utilisateurs (et de mots de passe) distinct, à l'aide de l'utilitaire *pg_passwd*. Exemple :

```
pg_passwd  /var/lib/pgsql/data/pwd_club
     Username: jules
     New password: xxxxx
     Re-enter New password: xxxxx
```

  Vous créez ainsi une table d'utilisateurs/mots de passe tout à fait indépendante, dans un fichier dont vous choisissez vous-même le nom et l'emplacement. Ce fichier sera de préférence installé dans le répertoire principal de *Postgres*, c.à.d. */var/lib/pgsql/data*. Il faudra aussi s'assurer que ce fichier ne soit accessible qu'à l'utilisateur *postgres* (attributs = 600).
  Ensuite, éditer le fichier */var/lib/pgsql/pg_hba.conf* pour y indiquer la correspondance entre les bases de données et les fichiers décrivant les utilisateurs autorisés à s'y connecter :

```
local    club                                  password  pwd_club
host     club     127.0.0.1     255.255.255.0 password  pwd_club
host     club     192.168.0.0   255.255.255.0 password  pwd_club
local    bd2                                   password  pwd_bd2
```

```
host      bd2       192.168.0.0   255.255.255.0 password  pwd_bd2
etc.
```

- Un logiciel client simple est à votre disposition pour vous connecter en ligne de commande.
  Entrer : **psql -U *utilisateur* -d *nom_de_la_base_de_données***
  → mot de passe, puis requêtes SQL, etc.
  Note : ne pas oublier de terminer les requêtes SQL par un point-virgule !
- Si vous en avez le droit (défini par createuser), vous pourrez modifier les paramètres d'un utilisateur (mot de passe, par exemple), à l'aide de la requête *SQL* ALTER USER :
  exemple : **ALTER USER freddy WITH PASSWORD 'dqbf457';**
- La requête GRANT permet de définir les droits exacts de chacun :
```
CREATE GROUP demogroup WITH USER jules,ernest;
GRANT [select,insert,update,delete,rule | ALL]
ON object1, object2, ... TO [GROUP demogroup | jules | PUBLIC];
```
- Pour créer un champ à numérotation automatique (utilisable comme clé primaire), il vous suffit de le déclarer comme étant du type SERIAL. Exemple :
```
CREATE TABLE membres (
     ref  SERIAL,
     nom  VARCHAR(30), ... etc ... ,
     CONSTRAINT ref_pk PRIMARY KEY (ref);
```
  Attention : il semble que ceci ne soit pas possible qu'au moment de la création de la table (cela ne marche pas d'ajouter après coup un champ de type SERIAL à une table existante).
- Les données d'administration se trouvent dans la base de données *template1*.
- Sous X, vous pouvez utiliser l'outil graphique **pgaccess** (ressemble un peu à Access)
- Sauvegarde d'une base de données : **pg_dump *nom_de_la_base* > *fichier_destination***
- Restauration d'une base : **psql *nom_de_la_base_à_restaurer* < *fichier_source***
- En cas de gros pépin, détruire le répertoire *data* dans */var/lib/pgsql*, puis recommencer avec */etc/init.d/postgresql start*, *createuser*, etc.
- Pour sortir la structure des tables (ou le résultat de requêtes diverses) sur imprimante :
  - entrer la commande de redirection de psql :
    \o |lp (toutes les sorties sont désormais redirigées vers lp)
  - entrer les commandes de listage (ou les requêtes) :
```
\d  table1
\d  table2    ... etc
```
  - terminer (quitter psql) pour que le contenu du buffer soit transféré :
    \q ==> l'impression commence.

Quelques requêtes SQL utiles

Définition/modification des contraintes d'intégrité référentielle :

```
ALTER TABLE nom_table1 ADD FOREIGN KEY (champA, champB, ...)
     REFERENCES nom_table2
     ON UPDATE CASCADE ON DELETE [SET NULL | RESTRICT];
```

Requêtes SELECT utilisant des expressions régulières :

```
SELECT * FROM friends WHERE firstname ~ '^D'
         (begins with D)

SELECT * FROM friends WHERE firstname ~* '^D'
     (contains D, case insensitive)

SELECT * FROM friends WHERE firstname !~* '^D'
         (does not contain D)

SELECT * FROM friends WHERE firstname ~ '^D.*e.*f'
     (begins with D, contains e and f)

SELECT * FROM friends WHERE firstname ~* '[A-D]'
     (contains A, a, B, b, C, c, D or d)
```

```
SELECT * FROM friends WHERE firstname ~ '[Aqzp]'
     (contains A, Q, z, or p)

SELECT * FROM friends WHERE firstname ~ 'G *$'
     (ends with G, with optional trailing sp.)
```

Utilisation de M$-Access comme logiciel client (via ODBC)

Le pilote *ODBC* est disponible sur le site de *PostgreSQL*. Il s'installe facilement.

Procédure à suivre sous *Access* pour transférer une base de données *.mdb* → *PostgreSQL* :

Ouvrir la base de données source (mdb).
Cliquer du bouton droit sur une zone vide de la fenêtre blanche.
Dans le menu qui apparaît, sélectionner "Lier les tables". Une nouvelle fenêtre "Attacher"apparaît.
Dans le bas de cette fenêtre, choisir le type de fichier : "ODBC databases ()" → Une fenêtre "Sélectionner la source de données" apparaît.
Cliquer sur "nouveau" → La fenêtre "Créer une nouvelle source de données" apparaît.
Choisir PostgreSQL , cliquer sur "Suivant". Choisir un nom pour le fichier d'interface.
Cliquer sur "Terminer" → La fenêtre "PostgreSQL connection" apparaît.
Entrer les informations demandées. Ne pas oublier de cliquer sur le bouton "Connection" dans l'encadré "Options (advanced)" pour décocher la case "read only", et éventuellement cocher la case "Show columns" dans "OID Options" → Retour à la fenêtre "Sélectionner la source de données".
Sélectionner la rubrique nouvellement créée → bouton OK → Connexion → Une fenêtre "Attacher les tables" apparaît.
Sélectionner la ou les tables à lier → OK → Sélectionner un champ pour la clé primaire.
A ce stade, les bases de données locale (mdb) et distante (odbc) sont toutes deux accessibles.
Sélectionner l'onglet "Requêtes".
Cliquer sur "Créer une requête en mode création".
Ajouter UNE table de la base de données locale (source).
Dans la petite fenêtre qui s'ouvre pour montrer les champs de cette table, double-cliquer sur chacun des champs à transférer.
Repérer l'icône "Type de requête" dans la barre d'outils. (Vers le centre de la barre. Icône avec deux tables superposées et une petite flèche indiquant la disponibilité d'un menu déroulant).
Cliquer sur la flèche accompagnant l'icône → Choisir "requête ajout" → Choisir le nom de la table destinataire (odbc) Dans la fenêtre de travail, vérifier que chaque champ de la table source possède son équivalent dans la destination. Au besoin, double-cliquer sur les cases vides pour y ajouter ce qui faut. Quand tout est OK, fermer la fenêtre pour enregistrer la requête.
Il suffira ensuite d'effectuer un double clic sur cette requête pour l'exécuter.

Accès depuis un script Python (le paquetage pygresql (ou postgresql-python dans la distribution SuSE 8.0) doit avoir été installé !) :

Le module principal devrait se trouver dans /usr/lib/python/site-packages/pg.py
Exemple d'accès à la base de données "music" :

```
from pg import DB                    # la classe DB contient tout le nécessaire
db = DB('music', user ='freddy', passwd ='dbx302')          # connexion
print db.get_tables()                # affichage des tables présentes
print db.query("select * from compos")    # requête type
# extraction de données sous la forme d'une liste de dictionnaires (chacun d'eux
# correspondant à une ligne de la table) :
dd = db.query("select * from compos where dnaiss >1800").dictresult
```

Accès à une base de données PostgreSQL via un script CGI dans une page Web :

La page web elle-même doit contenir un (ou plusieurs) formulaire(s) <FORM> :

```
<P><H3>Exercice : page Web gérée par un script Python</H3></P>
<P>Recherche dans la base de données "music" <BR>
Les informations compositeur sont dans la table "compos".</P>
<FORM method =POST ACTION="http://localhost/cgi-bin/website.py">
<P>Entrez ci-dessous votre requête SQL :</P>
<P><TEXTAREA NAME=requete ROWS=3 COLS=50></TEXTAREA></P>
<INPUT type="submit" NAME=send VALUE="Envoyer la requête">
</FORM>
```

Le script CGI lui-même (ici *website.py*) doit se trouver dans le répertoire réservé à ce genre de chose, c.à.d. pour *Apache*, tel qu'installé dans la distribution SuSE : /usr/local/httpd/cgi-bin

Dans le fichier de configuration d'Apache : /etc/httpd/httpd.conf , il faut activer des scripts CGI : (voir dans la partie <IfModule mod_alias.c>

```
    <Directory "/usr/local/httpd/cgi-bin">
        AllowOverride None
        Options +ExecCGI
        Order allow,deny
        Allow from all
    </Directory>
```

Pour que les élèves puissent créer eux-mêmes des sripts CGI, ajouter un alias :

```
ScriptAlias /python/ "/home/siteweb"
<Directory /home/siteweb>
  AllowOverride none
  Options +ExecCGI -Includes
  SetHandler cgi-script
</Directory>
```

Le fichier /var/log/httpd/error_log contiendra la description des erreurs de script

Exemple de script Python (*website.py*):

```
#! /usr/bin/python

# Exemple de script pour la gestion d'un site Web
from pg import DB
import os, cgi, sys

# Connexion à la base de données postgreSQL - instanciation :
db = DB('music')
# Acceptation des entrées utilisateur :
formulaire = cgi.FieldStorage()
# La méthode FieldStorage de l'objet cgi retourne un dictionnaire
# (contenant dans cet exemple un seul élément qui est une requête SQL) :
req = formulaire['requete'].value

# Affichage d'une page Web en réponse à la requête :
# Attention : la ligne ci-dessous est indispensable :
print "Content-Type: text/html \n"
print "<HTML><HEAD><TITLE>Réponse à une requête SQL</TITLE></HEAD><BODY>"
print "<H3>Formulaire bien reçu</H3>"
print "La requête introduite était : <BR>", req, "<BR>"
print "<H4>Résultat :</H4>"
res = db.query(req).dictresult()

for u in res:
   print u['ref'],u['prenom'],u['nom'],u['dnaiss'],u['dmort'],"<BR>"

print "</BODY></HTML>"
```

Notes diverses, en vrac :

WindowMaker

Thèmes et images de fond

On peut en trouver des dizaines sur internet. Si on souhaite les rendre accessibles à tous les utilisateurs, il faut les "décompresser" dans /usr/X11/share/WindowMaker/Themes (ou /usr/X11/share/WindowMaker/Backgrounds pour les simples images)

Les utilisateurs qui souhaitent en profiter doivent modifier leur fichier ~/GNUstep/Defaults/WMRootmenu : Sous la rubrique ("Apparence",("Themes",OPEN_MENU,"-noext etc., remplacer le chemin existant par /usr/X11/share/WindowMaker

La manière dont l'image de fond est installée sur le bureau (centrée, étirée, répétée en mosaïque, etc.) est définie dans le fichier **style** associé au thème : dans l'option WorkspaceBack, on aura par exemple :

- **cpixmap** pour une image centrée (*centered*) avec conservation de ses dimensions originales
- **spixmap** pour une image étirée (*stretched*) aux dimensions de l'écran (avec modification éventuelle du rapport hauteur/largeur)
- **mpixmap** pour une image maximisée, c.à.d. redimensionnée de manière à être la plus grande possible et entièrement visible, sans modification du rapport hauteur/largeur
- **tpixmap** pour une image répétée en mosaïque (*tiled*)

Configuration de l'image de fond d'écran lorsqu'elle est chargée sans passer par un thème :

Dans le fichier ~/GNUstep/Defaults/WMRootmenu , éditer la fin de la ligne "images" (sous la rubrique "Appearance"), et modifier les options de wmsetbg. Par exemple,

```
WITH wmsetbg -u -e          pour obtenir une image centrée
```

(Voir à ce sujet les pages de man : **wmaker** et **wmsetbg**).

Démarrage de Linux depuis WINDOW$ 95 ou 98, avec Loadlin

Note : *ceci ne marche pas avec Window$ ME. Si vous comptez utiliser Window$ ME et Linux sur la même machine, il vous faut installer LILO comme gestionnaire de boot.*

Dans le fichier config.sys de la machine :

```
[Menu]
MenuItem = WIN, Windows 98
MenuItem = LINUX, Linux SuSE 6.3
MenuDefault = LINUX, 30

[LINUX]
SHELL = C:\LOADLIN\LOADLIN.EXE C:\LOADLIN\VMLINUZ ROOT=/dev/hda7 MEM=96M
```

(l'indication MEM = ... est à utiliser dans le cas ou Linux ne détecte pas correctement la totalité de la mémoire installée, ce qui se produit parfois lorsqu'on utilise Loadlin comme lanceur de Linux)

```
[WIN]
device = ...
country = ... etc.
```

Créer un répertoire \LOADLIN dans la partition Windows. Y copier **Loadlin.exe** (que l'on peut trouver sur le CD d'origine, répertoire dosutils) et **vmlinuz** (lequel fichier contient le véritable noyau de Linux, qui devrait se trouver dans le répertoire /boot en fin d'installation. On pourrait aussi

recopier celui qui s'est placé sur la disquette de démarrage, si l'on en a créé une durant l'installation)

Pour effectuer la copie, il faut travailler sous Linux, avec par exemple :

```
cp /boot/vmlinuz-2.0.36-0.7 /dos/loadlin/vmlinuz
```

Eventuellement :

Modifier MSDOS.SYS : => BootGUI = 0 Logo = 0
Dans autoexec.bat, prévoir WIN à la fin

Réglage de l'heure et de la date

Il est vivement conseillé de synchroniser automatiquement l'heure et la date des postes clients avec celles du serveur (voir à ce sujet l'exemple de *logon script*). Pour remettre à l'heure et à la date le serveur Linux lui-même, utiliser la commande :

`date MMJJHHmm` avec MM = mois, JJ = jour, HH = heure, mm = minutes

Envoi d'un message WinPopup via Samba :

Sur les poste clients Win95 ou Win98, activer C:\Windows\Winpopup.exe
Lancer le petit script /usr/doc/packages/samba/examples/misc/wall.perl
Attention : les deux premières lignes font référence à des répertoires erronés.
Remplacer par « /usr/bin/smbstatus » et « /usr/bin/smbclient -M ».
Lancer le script en fournissant le nom du poste cible en paramètre.

Annulation de l'encryptage des mots de passe sur machines Win98 :

Si vous préférez désactiver l'encryptage automatique des mots de passe (Cfr. Discussion de ce problème dans la doc. de Samba), voici la procédure à suivre :

- Utiliser le CDROM de Win98. A l'aide de l'explorateur, repérer le fichier **D:\tools\mtsutil\Ptxt_on.inf**
- Cliquer dessus à l'aide du bouton droit de la souris -> choisir l'option Install.
- *Rebooter (il s'agit de Windows !).*

Installation de StarOffice en réseau

Le montage NFS mentionné plus haut dans le fichier /etc/fstab est nécessaire sur chaque poste où l'on souhaite faire fonctionner une version "réseau" de StarOffice.

Pour installer la partie "serveur" de StarOffice, lancer " ./setup /net " dans le répertoire temporaire où l'on a effectué la "décompression" de l'archive originale. Le répertoire choisi pour l'installation doit être celui qui est défini dans /etc/fstab ci-dessus comme devant être monté automatiquement par les postes clients à chaque démarrage.

Ensuite, sur chaque poste client, on se connecte en tant qu'utilisateur (autre que 'root'). Si le montage défini ci-dessus dans /etc/fstab est correct, il ne reste plus qu'à entrer dans ce répertoire partagé sur le serveur, passer au sous-répertoire /bin, et lancer le setup.

Lecteurs Zip

Un lecteur interne (ATAPI) est reconnu automatiquement comme "device" hdc4 (ou hda4, hdb4, etc. voir les messages du noyau au démarrage).

Pour un lecteur sur port parallèle, il faut essayer la commande :

`modprobe ppa` (ou, en cas de message d'erreur : **`modprobe imm`**).

Ajouter ensuite celle de ces 2 commandes qui est acceptée sans erreur, à la fin du script de démarrage contenu dans /etc/rc.d/rc.local

Installation d'une carte réseau ISA PNP compatible NE2000

(Ce cas s'est présenté à l'institut Don Bosco à Verviers)

Dans le BIOS, vérifier que l'option PNP OS installed est positionnée à "yes"

Sous Windows (ou en utilisant les outils fournis avec la carte), déterminer la plage d'adresses I/O (et l'IRQ ?). Cas présent : I/O addr. = 0240-025F

Sous Linux :

Installer le paquetage isapnp (série ap)

Sous root, lancer la commande **pnpdump -c > /etc/isapnp.conf**

Installer la carte réseau de la manière habituelle avec YAST. En paramètre, fournir l'adresse de départ de la plage I/O (pas l'irq). Ex: **io = 0x240**. Rebooter.

Installation de la carte 3COM Etherlink III 3c509 (carte ISA)

Certains BIOS ne parviennent pas à prendre en compte correctement l'automatisme PNP présent sur cette carte. Il faut alors reconfigurer la mémoire présente cette carte, à l'aide d'un programme de configuration disponible sur le site Web de la société 3Com (www.3com.com) et qui fonctionne sous DOS. A l'aide de ce logiciel, il faut désactiver le support PNP et configurer l'adresse de départ de la plage mémoire réservée à la carte ainsi que son numéro d'irq (nous avons choisi io = 0x300, irq 11).

Dans le BIOS de la machine, il faut en outre spécifier que l'irq choisie (11, en l'occurrence) sera utilisée par une carte ISA.

Création d'une disquette de boot à partir du CD d'origine (SuSE)

Formater une disquette (rejeter celles qui présentent des défauts) :

```
fdformat /dev/fd0u1440
```

Monter le premier CD de la distribution, par exemple au point de montage /cdrom

Copier l'image disque souhaitée (il y en a plusieurs. Consulter le fichier README qui se trouve dans le répertoire /cdrom/disks) :

```
dd if=/cdrom/disks/yast2 of=/dev/fd0 bs=8k
```

Problème de clavier avec SuSE 7.1 (touche AltGr inactive)

supprimer le répertoire **/usr/X11R6/lib/X11/xkb**
établir un lien symbolique vers /etc/X11/xkb à sa place :

```
ln  -s  /etc/X11/xkb  /usr/X11R6/lib/X11/xkb
```