

Installation de PC diskless

Alban Créquy

Été 2003

Table des matières

I	Sujet du stage	7
II	Description de la problématique	7
1	Clients lourds	7
2	Terminaux X	8
3	PC diskless	8
4	Besoins du LAOG	9
4.1	Population du laboratoire	9
4.2	Environnement	9
5	Historique des PC diskless au LAOG	9
5.1	Avant	9
5.2	État actuel	10
5.3	Ce qui reste à faire	10
III	Solutions de boot	10
6	Qu'est-ce que le boot ?	10
7	Boot sur la carte réseau PXE	11
7.1	Principe du boot en réseau	11
7.2	Installation et configuration du "bootloader" : pxelinux	12

8	Boot avec PxeGrub	13
8.1	Principe	13
8.2	Avantages	13
8.3	Inconvénient	13
8.4	Conclusion	14
8.5	Liens	14
9	Démarrage d'un PC diskless à partir du disque dur	14
9.1	Installation de Grub	14
9.2	Fichier de configuration de Grub	15
9.3	Copie du noyau du diskless	16
9.4	Résumé	16
10	CD bootable	16
10.1	El Torito	16
10.1.1	Avec l'image d'une disquette	16
10.1.2	Émulation	17
10.1.3	Problèmes	17
10.1.4	Conclusion	17
10.1.5	Liens	17
10.2	Introduction à Isolinux	17
10.2.1	Principe	17
10.2.2	Mode non-émulation	18
10.2.3	Étapes	18
10.2.4	Fichier de configuration d'isolinux	18
10.3	Méthode utilisée avec isolinux	19
10.3.1	Création des fichiers du CD	19
10.3.2	Créer l'image du CD	20
10.3.3	Graver l'image du CD	20
10.3.4	Avantages	21
10.3.5	Inconvénient	21
10.3.6	Liens utiles	21
10.4	Comment faire des CD génériques avec isolinux	21
10.4.1	Quel est le problème?	22
10.4.2	Comment résoudre le problème : première idée	22
10.4.3	Solution du problème	22
10.4.4	Mise au point	24
10.4.5	Conclusion	24
11	Procédure pour créer les CD bootable	24
11.1	Principe	24
11.2	procédure	24
11.3	Copier le noyau linux	25
11.4	Fichier de configuration d'isolinux	25

11.5	Créer l'image bootable du CD pour le PC diskless	26
11.6	Graver l'image	27
11.7	Démarrer le PC diskless sur le CD	28
11.8	Résumé	28
IV	Installation du serveur de diskless : Manuel utilisateur	28
12	Installation et configuration des services du serveur	28
12.1	Configuration du serveur DHCP	29
12.2	Configuration du serveur NFS	29
12.3	Configuration du serveur TFTP	30
13	Installation dans /diskless	30
13.1	Création des répertoires de /diskless	30
14	Installer le noyau du diskless	31
14.1	Faire d'abord une sauvegarde	31
14.2	Construire le noyau et les modules	32
15	Création des diskless : modèle et clones	33
15.1	Création du modèle	33
15.2	Fichiers de configuration propres au diskless	33
15.3	Faire du ménage	36
15.4	Placer les liens durs	36
15.5	Reconfigurer le serveur	36
15.6	Cloner un diskless	36
V	Partage des fichiers systèmes entre les diskless	37
16	Arborescence UNIX	37
17	Idée de base	38
18	Principe général	38
19	Méthode de mise en commun	38
20	Les fichiers d'authentification	39
21	Les fichiers utilisateurs	40

22 Les modules du noyau	40
22.1 Modules dans /diskless/lib/modules	40
22.2 Modules dans /diskless/lib/modules et dans les répertoires des diskless	41
22.3 Modules dans /diskless/lib/modules et liens durs	41
22.4 /lib en lien dur	42
23 Manipulation des liens durs	42
23.1 Fonctionnement	42
23.2 dkl-liens-durs.sh	43
23.3 dkl-liens-durs-1diskless.sh	43
23.4 dkl-llien-dur-diskless.sh	43
23.5 dkl-llien-dur-1diskless.sh	43
23.6 dkl-suppr-llien-dur-diskless.sh	44
23.7 dkl-suppr-llien-dur-1diskless.sh	44
VI Installation de logiciels sur les PC diskless	44
24 Qu'est-ce que RPM et URPMI?	44
24.1 Principe de URPMI	44
24.2 Comment installer les sources la première fois	45
24.3 Mises à jour automatiques	45
24.4 Liens utiles sur URPMI	45
25 Problèmes	45
25.1 NFS lock	45
25.2 Partages entre diskless	47
25.2.1 Détails du problème	47
25.2.2 Une piste : URPMI-parallel	47
25.2.3 Une autre piste : installation dans un chroot	48
25.2.4 Une autre piste avec quelques scripts	48
26 Procédure d'installation de paquets	48
26.0.1 Préparation	49
26.0.2 Partage des bases de données RPM et URPMI	49
26.0.3 Première étape : sur un diskless	49
26.0.4 Seconde étape : sur le serveur	50
26.0.5 Résolution de problèmes RPM	50
VII Mises au point	51
27 XFree	51

28 Problèmes à l'arrêt du diskless	51
28.1 Où se situe le problème	51
28.2 Qu'est-ce qu'un démontage paresseux?	52
28.3 Ce qui se passe exactement	52
28.4 Première idée	52
28.5 La solution	53
28.6 D'autres problèmes	53
28.7 Conclusion	54
29 Problème des pilotes NVidia	54
30 Conflits entre serveurs DHCP	55
31 NFS lockd	57
32 Message "chgrp : invalid group name : utmp"	58
33 Impossible de trouver les modules dans /lib/modules/2.4.21-01	59
VIII Utilisation du disque dur du diskless	59
34 Introduction	59
35 Installation sur le serveur	60
36 Installation sur le diskless	60
IX Annexes	61
A Glossaire	61
B Arborescence du serveur de diskless	62
C Machines de tests	62
D Fichiers XF86Config-4	63
D.1 Fichier du diskless gag220	63
D.2 Fichier du diskless gag220	66
E Ce qu'il reste à faire	69
E.1 Vitesse de KDE	69

F	Scripts	69
F.1	dkl-init-serveur.sh	69
F.2	dkl-init-diskless.sh	71
F.3	dkl-diskless-menage.sh	72
F.4	dkl-liens-durs.sh	73
F.5	dkl-liens-durs-1diskless.sh	74
F.6	dkl-1lien-dur-diskless.sh	74
F.7	dkl-1lien-dur-1diskless.sh	74
F.8	dkl-suppr-1lien-dur-diskless.sh	75
F.9	dkl-suppr-1lien-dur-1diskless.sh	75
F.10	dkl-tftp-start.sh	76
F.11	dkl-graver-cdrom.sh	76
F.12	dkl-syslogd.sh	77
F.13	dkl-paquetage-install.sh	78
F.14	dhcpd.conf-exemple.txt	79
F.15	exports-exemple.txt	80
F.16	fstab-exemple.txt	80
F.17	isolinux.cfg	81
G	À propos de ce document	82

Première partie

Sujet du stage

Des méthodes de boot pour des PC Linux sans disque ont été mises au point. Je dois faire une synthèse de ces méthodes, rechercher de nouvelles méthodes et les mettre au point, puis éditer un manuel utilisateur de toutes ces méthodes.

Deuxième partie

Description de la problématique

Un parc informatique contenant plusieurs dizaines/centaines/milliards (rayer les mentions inutiles) de PC ne peut pas être géré sans méthode.

Le but de l'administrateur est de gérer les différentes ressources sur le réseau. Ces ressources sont :

1. Espace de stockage de données (disques durs)
2. Puissance de calcul (processeurs)
3. Capacité du réseau (bande passante)
4. Serveurs (machines placées dans la salle des machines)
5. Terminaux (machines utilisées directement par l'utilisateur)

Il existe plusieurs grands modèles pour gérer ces ressources. Je vais décrire brièvement trois catégories : celle des clients lourds, celle des terminaux X et celle des PC diskless (PC sans disque).

1 Clients lourds

Principe : installation du système GNU/Linux sur les disques durs de chaque PC.

Inconvénients :

1. L'administration est très fastidieuse. En effet, il est nécessaire de passer sur chaque machine pour mettre à jour le système.
2. Par conséquent, les systèmes installés sur les clients lourds ne sont jamais à jour.
3. Les systèmes sont hétérogènes. L'administration est donc plus difficile.

Avantage : Les applications s'exécutent en local sur le PC ; les serveurs ne sont donc pas surchargés.

2 Terminaux X

Principe : la machine est un terminal X ou PC qui sert juste à afficher les fenêtres des programmes de l'utilisateur sur un écran. La machine ne fait quasiment aucun calcul et tout est centralisé sur les serveurs.

Inconvénients :

1. Cela surcharge des serveurs car toutes les applications s'exécute sur les serveurs. Les applications semblent donc tourner moins vite.
2. Les terminaux X n'ont aucune puissance de calcul utilisable. Pour le même prix, un PC est plus puissant.

Avantages :

1. L'administration est centralisée. C'est donc plus pratique que les clients lourds.
2. Un terminal X ne fait aucun bruit.
3. Un terminal X ne chauffe quasiment pas.

3 PC diskless

Principe : Un PC sans disque ou "PC diskless" est un poste de travail dont le disque s'il en a un, n'est pas dédié au système Linux. Le PC diskless peut démarrer par différents moyens que j'expliquerai par la suite (disquette, CD, carte réseau PXE). Le système d'exploitation est récupéré sur un serveur de diskless (voir glossaire page 62).

Inconvénient :

1. Il faut trouver des méthodes de boot. Cela fait partie de mon stage.
2. La méthode n'est pas encore tout à fait au point.

Avantages :

1. Toutes les applications s'exécutent sur le PC diskless (les serveurs sont donc peu chargés). Utiliser des PC diskless est donc plus rapide qu'utiliser des terminaux X.
2. De plus, toutes les données sont centralisées sur les serveurs. L'administration des PC diskless est donc plus facile que l'administration des clients lourds.
3. C'est moins coûteux car on n'a pas besoin de disque.
4. Moins bruyant car il n'y a pas de disque.
5. Les PC diskless chauffent moins car il n'y a pas de disque.

6. Les PC diskless peuvent être utilisés comme grille de calcul (voir glossaire page 61)

4 Besoins du LAOG

4.1 Population du laboratoire

Les besoins de l'ensemble de la population du laboratoire est homogène, hormis l'administration du laboratoire qui travaille sur des plateformes dédiées à la bureautique. La population, pour 80% du laboratoire a besoin d'un poste de travail entièrement géré par l'administrateur : les doctorants, la majorité des chercheurs, les stagiaires et invités et les étudiants de l'école doctorale. Seules les machines dédiées au pilotage de manips ou au développement d'instruments sont gérées par les ingénieurs concernés.

40% de la population se renouvelle régulièrement (stagiaires, invités, étudiants, doctorants) et les équipements des bureaux sont amenés à bouger, ce qui nécessite de poser des postes de travail banalisés et non encombrants.

4.2 Environnement

La nécessité est d'avoir des postes légers, non bruyants, banalisés, configurés à distance par l'administrateur, dont le nombre doit évoluer avec l'augmentation du nombre de personnes du laboratoire : une centaine actuellement. Le type des postes évolue selon le marché et la technologie. Les répertoires des utilisateurs (\$HOME) sont centralisés sur une machine serveur NFS, concentrés sur une baie RAID, donc indépendants des postes de travail. La mise à jour par l'administrateur, du système centralisé entraîne une mise à jour de tous les postes de travail à la fois.

Tous les logiciels hors licences doivent tourner en local, et être accessibles via NFS sur un serveur d'applications. Les grands espaces disques nécessaires aux simulations, modélisations, dépouillement de données sont servis par des serveurs de RAID.

5 Historique des PC diskless au LAOG

5.1 Avant

Les PC bootaient sur une disquette contenant LILO (voir glossaire page 61) et le noyau Linux. Tous les systèmes de fichiers étaient montés par NFS (voir glossaire page 62). La disquette allait donc chercher les fichiers système sur le serveur. Mais depuis, les noyaux Linux sont devenus trop gros pour être mis sur une disquette. Chercher à réduire l'espace utilisé par le noyau sur la disquette n'est pas une solution car le noyau risque de continuer à grossir au fil du temps. Cette méthode n'est donc plus utilisée.

5.2 État actuel

Les PC diskless démarrent sur le réseau en utilisant le protocole de démarrage PXE (voir glossaire page 62).

PXE permet de booter sur le réseau. Concrètement, au démarrage du PC diskless, le BIOS (glossaire page 61) donne la main au programme PXE qui se trouve sur la carte réseau. Ce programme fait une requête DHCP (glossaire page 61) pour trouver un serveur qui lui donnera son adresse IP (glossaire page 61) et le nom du serveur qui lui permettra de télécharger le noyau avec TFTP (glossaire page 62).

Ce protocole s'est bien développé : une bonne partie des cartes réseaux du LAOG supportent PXE. Toutes les nouvelles cartes réseaux le supporte.

5.3 Ce qui reste à faire

Mais toutes les cartes réseaux ne supportent pas PXE. Il faut donc trouver d'autres méthodes pour ces PC.

De plus, la méthode utilisée pour administrer les diskless doit être améliorée pour corriger certains problèmes et faciliter l'administration.

Troisième partie

Solutions de boot

6 Qu'est-ce que le boot ?

Lorsqu'un PC démarre, il doit trouver son système d'exploitation, le charger et l'exécuter. C'est ce qu'on appelle le boot.

Dans le cas de GNU/Linux, la méthode la plus employée est la suivante :

1. Le BIOS (glossaire page 61) possède plusieurs options de démarrage : disque dur, CD, etc. La valeur la plus courante de cette option est de démarrer sur le disque dur.
2. Le BIOS cherche donc sur le disque dur une zone qu'on appelle le MBR (pour Master Boot Record). Cette zone contient le chargeur de démarrage. Il s'agit de GRUB ou de LILO. Le BIOS donne la main à ce programme.
3. Le chargeur de démarrage trouve le noyau Linux sur le disque dur, le charge en mémoire puis lui donne la main en lui passant des paramètres.
4. Linux démarre et tient compte de ces paramètres pour savoir que faire.

Mais dans le cas d'un diskless, le disque ne doit pas être utilisé pour héberger le système d'exploitation. Il faut donc trouver d'autres méthodes de boot.

La méthode utilisée au LAOG actuellement est la méthode de boot par réseau avec le protocole PXE (glossaire 62).

Je vais d'abord décrire cette méthode puis les différentes méthodes que j'ai trouvées.

7 Boot sur la carte réseau PXE

Une documentation sur PXE existe sur :

<http://clic.mandrakesoft.com/documentation/pxe/>

Ce paragraphe est tiré de :

http://sari.inpg.fr/rubriques/themes/zone_publicque.groupe_linux/diskless/serveur-diskless.html

7.1 Principe du boot en réseau

Le boot réseau consiste pour une machine cliente à obtenir auprès d'une machine serveur son identité réseau (son adresse IP, son adresse de diffusion, son masque réseau, son nom de machine, son nom de domaine, les noms des serveurs DNS, l'adresse de la passerelle) et des informations sur le serveur de boot (l'adresse IP du serveur et le nom du fichier de boot).

La machine cliente doit être raccordée au réseau et positionnée de manière à booter d'abord sur son interface réseau (UNDI/PXE) : ceci est à positionner dans le BIOS du PC. Le protocole PXE indique les étapes nécessaires pour le démarrage effectif de la machine client :

1. Émission par le client d'une requête DHCPDISCOVER de demande de son identité réseau ; cette requête contient son adresse MAC. Réception par le serveur dhcp de la requête qui la traite en recherchant une correspondance de l'adresse MAC dans son fichier dhcpd.conf, puis envoie au client une réponse DHCPOFFER contenant son identité réseau si elle a été trouvée.
2. Réception par le client et positionnement de ces paramètres (équivalent ifconfig) puis envoi au serveur une requête DHCPREQUEST demandant les informations de boot
3. Réception par le serveur de la requête DHCPREQUEST qui la traite en recherchant les infos demandées dans son fichier dhcpd.conf, puis envoie au client une réponse DHCPACK contenant l'adresse IP du serveur TFTP et le nom du fichier de boot (pxelinux.0).
4. Réception par le client de la requête DHCPACK qui établit une connexion au serveur TFTP (port 69) et demande le chargement du fichier indiqué.

Après le chargement, exécution de ce fichier qui, toujours par une connexion TFTP :

1. chargera un fichier de configuration se trouvant dans un répertoire pxelinux.cfg ; celui-ci contient le nom du noyau et les options de démarrage de celui-ci ;
2. chargera ensuite le noyau et les autres fichiers nécessaires au boot (fichier initrd).

Le noyau est alors démarré... Le fichier initrd est un fichier système de fichiers compressé ; le système de fichiers contient les modules noyau et toute autre chose utile.

7.2 Installation et configuration du "bootloader" : pxelinux

Le programme de boot (bootloader) est celui qui chargera en fait le noyau linux en mémoire. Pour PXE, il faut un bootloader "intelligent" capable de charger un noyau et des images initrd, de les décompresser.

Après la création de /diskless/tftpboot dans lequel on a construit les systèmes nécessaires aux diskless clients de ce serveur, il faut créer un répertoire pxelinux.cfg (ceci est normalement déjà fait par le script dkl-init-serveur.sh, cf. partie Installation).

1. cd /tftpboot
2. mkdir pxelinux.cfg

pxelinux se trouve dans l'outil syslinux (la version utilisée est la 1.72). Installer syslinux-1.72.tar.gz sous /usr/local, depuis <http://www.kernel.org/pub/linux/utils/boot/>. Je l'ai ajouté dans l'archive diskless-kit.tar.gz (cf. partie Scripts en annexe).

1. cd /tftpboot
2. cp /usr/local/syslinux-1.72/pxelinux.0 /tftpboot

Dans le répertoire /tftpboot/pxelinux.cfg, sera édité un fichier par PC diskless qui bootera par PXE. Chaque fichier a pour nom le numéro hexadécimal de l'adresse IP du PC (Ex. C3DC4F0C : C3 pour 195,DC pour 220, 4F pour 79 et 0C pour 12).

Ex. de fichier "bootloader" :

```
#gagXXX
DEFAULT bzImage-diskless-pxel
APPEND root=/dev/nfs hdg=ide-scsi devfs=mount ip=: :: :: :dhcp nfsroot=n0-IP-serveur-nfs:/
PROMPT 1
TIMEOUT 50
```

La première ligne donne le nom du noyau linux (bzImage...) qui doit être placé sous /tftpboot. La seconde ligne donne les options du noyau.

Pour les PC avec graveur de CD, l'option hdg=ide-scsi ==> le noyau ne reconnaîtra plus le graveur en tant que périphérique IDE (et donc plus en tant que /dev/hdg mais en tant que /dev/scd0).

Le noyau a été créé spécifiquement pour le diskless à partir du fichier de configuration qui a servi à créer le noyau du serveur.

Pour visualiser le contenu d'une image :

1. `mv initrd.img initrd.img.gz`
2. `gunzip initrd.img.gz`
3. `mount /tmp/initrd.img /mnt -t ext2 -o loop`
4. `ls /mnt`

8 Boot avec PxeGrub

8.1 Principe

Il n'est plus possible de mettre le noyau sur une disquette car les nouveaux noyaux sont devenus trop gros. Une solution est donc de mettre le noyau sur le serveur et de trouver un moyen de télécharger ce noyau par le réseau. Une solution est d'utiliser PxeGrub (page 62).

Le diskless démarre sur une disquette. Cette disquette contient PxeGrub, mais pas le noyau linux. PxeGrub va télécharger le noyau sur un serveur. Puis ce noyau est démarré et les systèmes de fichiers sont montés par NFS.

8.2 Avantages

1. Permet de démarrer un noyau à partir d'une disquette, même si le noyau ne tient pas sur la disquette car PxeGrub va télécharger le noyau sur le serveur de diskless.
2. Pas de contrainte sur la taille du noyau.
3. Pas besoin de carte réseau compatible PXE.
4. PxeGrub est lancé, soit en mode interactif (l'utilisateur choisit s'il veut démarrer sur le réseau ou sur le disque local, ou autres) soit en mode automatique (PxeGrub démarre sur le réseau sans rien demander à l'utilisateur). On peut donc proposer un menu à l'utilisateur.

8.3 Inconvénient

La carte réseau doit être reconnue par PxeGrub. Or, une partie des cartes réseaux du LAOG n'est pas reconnue par PxeGrub. Toutes les cartes réseaux reconnues par Linux ne sont pas reconnues par PxeGrub.

La liste des cartes réseaux reconnues par PxeGrub n'est pas la même que la liste des cartes compatibles PXE (ca n'a rien à voir). Pour qu'une carte réseau soit compatible PXE, il faut que le protocole PXE soit programmé dans la puce de la carte réseau. Selon les modèles, une carte réseau peut ou pas être compatible PXE. Mais pour qu'une carte soit reconnue par PxeGrub, il faut que les développeurs du logiciel PxeGrub écrivent le pilote de cette carte.

pour PxeGrub. PxeGrub est en cours de développement et tous les pilotes ne sont pas disponibles.

8.4 Conclusion

Cette méthode n'est pas retenue car les PC diskless doivent marcher sur tous les PC du LAOG.

8.5 Liens

Liste des cartes compatibles GRUB :
<http://etherboot.sourceforge.net/db/>

9 Démarrage d'un PC diskless à partir du disque dur

Un "PC diskless" peut très bien avoir un disque dur. PC diskless veut simplement dire que le disque dur n'est pas utilisé pour le système.

Dans cette partie, nous allons utiliser le disque dur pour démarrer le système mais on utilisera toujours NFS-Root (la racine du système sera montée par NFS). L'espace utilisé sur le disque dur du PC diskless sera environ 2 Mo.

Le noyau linux sera placé sur une partition déjà existante (ext2, ext3, ou vfat...) dans le répertoire /Grub. Grub sera installé sur le disque dur et proposera au démarrage de la machine un menu. L'utilisateur pourra choisir entre booter sur le réseau ou démarrer le système présent sur le disque dur, s'il existe. Grub peut ainsi démarrer de nombreux systèmes d'exploitation, dont Linux et Windows.

9.1 Installation de Grub

Avant de commencer, assurez vous qu'un partition est déjà présente sur le disque dur avec suffisamment d'espace libre (2 ou 3 Mo).

1. Démarrer sur le CD d'installation de Mandrake.
2. Taper F1, immédiatement au démarrage puis taper rescue.
3. Choisir "Go to console".
4. Éventuellement, taper loadkeys fr si vous utilisez un clavier français.
5. Monter une partition du disque dur sur /mnt/disque.
6. `cp -a /boot/grub /mnt/disque`
7. Lancer grub

8. Taper `find /grub/stage1`. Noter le résultat de la commande (par exemple `"(hd0,0)"`). La commande `find` indique où sur quelle partition se trouve le fichier `/grub/stage1` selon la notation Grub. Grub lit toutes les partitions de tous les disques durs pour indiquer le résultat. Aucune modification n'est effectuée à cette étape.
9. Taper `"root (hd0,0)"` (ou adapter selon le résultat de la commande `find`). La commande `root` indique que le fichier de configuration se trouve sur la partition `(hd0,0)`
10. Taper `"setup (hd0)"`. La commande `setup` installe Grub sur le premier disque `(hd0)`.
11. Taper `"quit"` pour sortir de Grub.

Grub est maintenant installé.

9.2 Fichier de configuration de Grub

Il faut maintenant changer son fichier de configuration. Celui-ci se trouve sur `/mnt/disque/grub/menu.lst`

Voici un exemple de fichier de configuration :

```
timeout 30
default 0

title Demarrer en tant que diskless
kernel (hd0,0)/grub/vmlinuz-diskless root=/dev/nfs devfs=mount ip:::::dhcp nfsroot=/di

title Demarrer a partir du disque dur
kernel (hd0,0)/boot/vmlinuz root=/dev/hde1 devfs=mount vga=788
initrd (hd0,0)/boot/initrd.img

title Disquette
root (fd0)
chainloader +1
```

Grub numérote les disques durs à partir de 0 (`hd0`, `hd1`) tandis que Linux les numérote à avec des lettres partir de a (`hda`, `hdb`).

Les partitions sont aussi numérotées à partir de 0 (`(hd0,0)`, `(hd0,1)`) tandis que Linux commence à 1 (`hda1`, `hda2`).

Avec ce fichier de configuration, l'utilisateur aura le choix entre trois démarrages possibles :

1. Démarrer en tant que diskless
2. Démarrer à partir du disque dur
3. Démarrer à partir d'une disquette

9.3 Copie du noyau du diskless

Il faut ensuite copier le noyau du diskless sur `/grub/vmlinuz-diskless`.

Le répertoire `/grub` (sur le disque du diskless) doit donc contenir ces 4 fichiers :

1. `menu.lst`
2. `stage1`
3. `stage2`
4. `vmlinuz-diskless`

9.4 Résumé

Le PC diskless démarre sur le disque dur. Sur ce disque dur, il y a Grub. Celui-ci propose à l'utilisateur de démarrer en tant que diskless. En sélectionnant ce choix, Grub charge le noyau qui se trouve dans le répertoire `/grub` du disque dur. Les modules et la racine du système se trouvent comme d'habitude sur le serveur de diskless.

10 CD bootable

Un CD bootable est un CD, qui, inséré au démarrage de l'ordinateur permet de charger un système d'exploitation. Ce système d'exploitation peut être sur le CD (comme par exemple le système Knoppix) ou il peut être téléchargé sur un serveur (c'est le cas pour un PC diskless).

J'ai étudié plusieurs méthodes pour faire un CD bootable. Ma première piste était de faire un CD au format El Torito. Puis j'ai trouvé un outil bien pratique : `isolinux`.

10.1 El Torito

Première possibilité : faire un CD au format "El Torito".

10.1.1 Avec l'image d'une disquette

Un CD au format El Torito est un CD bootable qui contient un fichier image d'une disquette ou d'un disque dur.

La technique est la même que pour créer une disquette ou un disque bootable. En fait, il faut mettre un fichier image d'une disquette (ou d'un disque) sur le CD et mettre un pointeur vers cette image au début du CD. Cela se fait facilement avec une option du logiciel de création d'image de CD : `mkisofs`.

10.1.2 Émulation

En fait, quand le BIOS doit démarrer sur le CD, cela ne se passe pas comme pour une disquette ou un disque dur. Le BIOS charge l'image de la disquette ou du disque dur qui se trouve sur le CD et boote dessus en émulant une disquette (ou le disque dur). Cette émulation se fait en changeant les fonctions du BIOS qui permettent d'accéder à la disquette ou au disque dur. Ainsi, une fois le système d'exploitation chargé, il croit qu'il se trouve sur une disquette (ou un disque dur).

10.1.3 Problèmes

Mais cette technique est très contraignante : si on utilise une image de disquette, on reste limité aux 1,44Mo (donc impossible d'utiliser de gros noyaux) et utiliser une image d'un disque est assez compliqué.

En effet, bien qu'il existe des outils sous GNU/Linux pour créer un fichier image d'une partition assez facilement (exemple d'outils : dd, mkfs, mount -o loop), je n'ai pas trouvé d'outils pour créer un fichier image d'un disque. Lorsque j'essaie de partitionner l'image de mon disque avec fdisk, ce dernier me demande le nombre de cylindre de l'image disque et je n'ai pas la réponse. De plus je ne sais pas comment monter avec mount la partition qui se trouve dans l'image du disque.

10.1.4 Conclusion

Cette première possibilité n'est donc pas utilisée.

10.1.5 Liens

Description du format de CD El Torito :
<http://www.phoenix.com/resources/specs-cdrom.pdf>

10.2 Introduction à Isolinux

Le site de isolinux :
<http://syslinux.zytor.com/iso.php>

10.2.1 Principe

C'est un chargeur de démarrage pour Linux sur CD. Il permet de charger un noyau Linux sans contrainte de taille. Isolinux est utilisé notamment sur les CD d'installation de Mandrake, Red Hat, Debian et sans doute d'autres. Isolinux fait partie du projet Syslinux.

10.2.2 Mode non-émulation

Isolinux fonctionne sans émulation de disquette ou de disque dur. Il n'y a donc pas de limite sur la taille de ce que l'on met sur le CD : le noyau.

10.2.3 Étapes

Voici les Étapes de la méthode que nous allons mettre en place :

1. Créer une image bootable du CD pour le PC diskless
2. Graver cette image.
3. Démarrer le diskless sur le CD.

10.2.4 Fichier de configuration d'isolinux

Quand isolinux est installé sur le CD, son comportement est déterminé par son fichier de configuration qui se trouve dans le répertoire `/isolinux/isolinux.cfg` du CD. Isolinux propose au démarrage un menu avec plusieurs options possibles, tout comme LILO ou GRUB. Ainsi, on peut mettre plusieurs choix possibles au démarrage d'isolinux en mettant les entrées correspondantes dans le fichier de configuration.

Voici un exemple du fichier `isolinux.cfg` :

```
default linux
prompt 0
label linux
KERNEL alt0/vmlinuz
APPEND root=/dev/nfs devfs=mount ip=:::::dhcp \
nfsroot=195.220.79.217:/diskless/tftpboot/195.220.79.220
```

Voici ce que veut dire chaque ligne de ce fichier :

1. Le choix par défaut du système que l'on va démarrer est "linux". On pourrait mettre plusieurs choix et laisser l'utilisateur choisir entre ces systèmes d'exploitation. Le nom "linux" correspond au label correspondant.
2. Avec "prompt 0", on précise qu'on ne veut pas que l'utilisateur puisse choisir. Il faut mettre "prompt 1" dans le cas contraire
3. "label linux" permet de nommer un choix de démarrage. C'est ce que l'utilisateur devra taper pour démarrer.
4. Avec l'option `KERNEL`, on choisit le noyau à démarrer. Le chemin est relatif au répertoire `isolinux`. Ainsi, "alt0/vmlinuz" désigne le noyau `/isolinux/alt0/vmlinuz` du CD. Le nom `alt0` est une convention, on aurait très bien pu prendre autre chose. Cela veut dire "alternative 0". Si on voulait proposer le choix entre plusieurs noyaux, on aurait pu les

mettre dans alt0, alt1, alt2, ... Attention, le nom du fichier noyau ne doit pas dépasser 8 caractères (c'est une contrainte d'isolinux).

5. Enfin, avec l'option APPEND, on choisit toutes les paramètres à passer au noyau linux.

Voici la signification des paramètres passés au noyau linux :

1. root=/dev/nfs : indique que la racine doit être montée par NFS.
2. devfs=mount : activer l'option devfs. Devfs permet de gérer les périphériques en mettant automatiquement les fichiers dans /dev/. Sans devfs, il faudrait créer tous les fichiers de /dev manuellement. C'est donc une option indispensable.
3. ip=::: : dhcp : indique les paramètres réseaux de la machine. Ici, on dit au noyau de faire une requête DHCP pour récupérer ces paramètres.
4. nfsroot=... : indique où trouver la racine du système de fichier. 195.220.79.217 est l'IP du serveur NFS et /diskless/tftpboot/195.220.79.220 est le chemin à monter sur le serveur.

Isolinux va donc nous permettre de démarrer le noyau à partir du CD.

10.3 Méthode utilisée avec isolinux

10.3.1 Création des fichiers du CD

Nous allons mettre nos scripts dans le répertoire /diskless/scripts. Créez le s'il n'existe pas déjà.

Tout ce qui est en rapport avec la création du CD sera mis sur le serveur dans /diskless/CDROM. Voici l'arborescence de /diskless/CDROM :

```
/diskless/CDROM
| - isos : les images des CD seront placees ici
| - root : correspond a la racine du CD
|   | - isolinux : contient le binaire d'isolinux et son fichier de config.
|   | - alt0 : contient le noyau vmlinuz
```

Cette arborescence est créée automatiquement par le script *dkl-init-serveur.sh*. Voici ce qui doit être présent dans /diskless/CDROM/root/ :

1. le fichier isolinux/isolinux.bin issu de l'archive syslinux
2. le fichier de configuration isolinux/isolinux.cfg

Tous ces fichiers sont disponibles, voir la partie *Scripts* en annexe.

Le script /diskless/scripts/dkl-init-diskless.sh a déjà créé les répertoires dans /diskless/CDROM décrits ci-dessus (voir la partie Installation du serveur).

/diskless/CDROM/root/ sera la racine de notre CD.

Copier le noyau du diskless dans /diskless/CDROM/root/isolinux/alt0. Par convention, on appelle le noyau vmlinuz car le nom est limité à 8 caractères.

```
cp -a /diskless/tftpboot/bzImage-diskless-pxe1 \
      /diskless/CDROM/root/isolinux/alt0/vmlinuz
```

Modifier le fichier /diskless/CDROM/root/isolinux/isolinux.cfg pour mettre à jour l'IP du serveur comme expliqué dans la partie *Introduction à isolinux*.

10.3.2 Créer l'image du CD

Le script /diskless/scripts/dkl-graver-cdrom.sh permet ensuite de créer l'image de notre CD bootable avec mkisofs. Voici ce qu'il contient :

```
mkisofs -o /diskless/CDROM/isos/iso1.iso \
        -b isolinux/isolinux.bin \
        -c isolinux/boot.cat \
        -no-emul-boot -boot-load-size 4 \
        -boot-info-table \
        /diskless/CDROM/root/
```

1. -o permet de choisir où mkisofs va mettre le fichier image (glossaire page ??) du CD bootable.
2. -b permet d'indiquer à mkisofs où se trouve le fichier binaire isolinux.bin par rapport à la racine du CD. C'est ce fichier qui sera exécuté au démarrage du CD
3. -c permet d'indiquer à mkisofs où il doit créer le fichier boot.cat. Ce fichier n'existe pas, c'est normal : mkisofs va le créer et une fois le CD gravé, on le trouvera sur le CD.
4. -no-emul-boot permet de graver l'image sans émuler une disquette ou un disque pour démarrer le CD. En effet, isolinux fonctionne sans émulation.
5. -boot-load-size 4 et -boot-info-table sont des options pour créer l'image bootable. Je n'ai pas compris ce qu'elles faisaient exactement mais elles semblent indispensable pour créer l'image bootable.
6. Enfin, /diskless/CDROM/root/ indique ce qui doit être gravé.

10.3.3 Graver l'image du CD

Une fois le script dkl-graver-cdrom.sh lancé, l'image se trouve dans /diskless/CDROM/isos/iso1.iso. Il ne reste plus qu'à graver l'image.

Cette image de CD peut être gravée comme n'importe quelle autre image. Par exemple, on peut utiliser cdrecord :

```
cdrecord dev=0,0,0 blank=fast -eject /diskless/CDROM/isos/iso1.iso
```

1. L'option `blank=fast` permet d'effacer le CD avant de graver s'il s'agit d'un CD réinscriptible.
2. L'option `eject` permet de sortir le CD du lecteur après la gravure.
3. `/diskless/CDROM/isos/iso1.iso` indique l'image à graver.

10.3.4 Avantages

1. Pas de contrainte sur la taille du noyau.
2. Toutes les cartes réseaux compatibles linux marcheront avec cette méthode.

10.3.5 Inconvénient

Il faut trouver une méthode pour faire un CD générique. En effet, il serait fastidieux de créer un CD différent pour chaque diskless ! Pour l'instant, on a écrit sur le CD l'adresse IP du diskless et du serveur dans le fichier `isolinux.cfg`. Le CD ne sera donc utilisable que sur ce diskless. Ce problème est résolu dans la partie suivante

10.3.6 Liens utiles

1. Le répertoire `/isolinux` du CD d'installation de Mandrake.
2. La page du manuel de `mkisofs` (`man mkisofs`)
3. Le site d'`isolinux` <http://syslinux.zytor.com/iso.php>
4. Description du format El Torito <http://www.phoenix.com/resources/specs-cdrom.pdf>
5. Documentation sur la création de CD bootables <http://tldp.org/HOWTO/Bootdisk-HOWTO/cd-roms.html>
6. `Isolinux` peut être téléchargé sur : <http://www.kernel.org/pub/linux/utils/boot/syslinux/>
7. La dernière version est `Syslinux 2.04` : <http://www.kernel.org/pub/linux/utils/boot/syslinux/syslinux-2.04.tar.bz2>

10.4 Comment faire des CD génériques avec `isolinux`

Un CD bootable générique est un CD qui marchera sur tous les PC diskless du LAOG.

Nous avons vu comment créer un CD bootable qui permet de démarrer un PC diskless en utilisant un serveur de diskless. Cependant, cette méthode a un gros défaut : il faut faire un CD différent pour chaque diskless !

10.4.1 Quel est le problème ?

Avec la méthode précédente, il fallait un CD différent par diskless.

Le seul paramètre qui change d'un diskless à l'autre dans notre création de CD se trouve dans le fichier de configuration d'isolinux (isolinux.cfg) :

```
default linux
prompt 0
label linux
KERNEL alt0/vmlinuz
APPEND root=/dev/nfs devfs=mount ip=:::::dhcp \
      nfsroot=195.220.79.217:/diskless/tftpboot/195.220.79.220
```

L'adresse IP 195.220.79.220 est l'adresse IP du client. Celle-ci diffère sur chaque diskless. Le point d'où doit être montée la racine de notre diskless n'est pas constante.

10.4.2 Comment résoudre le problème : première idée

Le chemin à accéder sur le serveur (/diskless/tftpboot/195.220.79.220) doit être écrit sur le CD. Il faut donc :

1. Mettre un chemin unique sur le CD
2. Le serveur NFS doit différencier les clients selon leur IP et donner des répertoires différents.

Malheureusement, le serveur NFS de Linux ne permet pas ça... Cette idée n'est donc pas applicable !

10.4.3 Solution du problème

Le diskless doit trouver le chemin sur le serveur pour sa racine en fonction de son adresse IP. Mais cette adresse IP ne doit pas être écrite sur le CD car elle est différente pour chaque diskless. La solution se trouve en fait dans la documentation de Linux sur :

`/usr/src/linux/Documentation/nfsroot.txt`

Il s'agit de l'option "nfsroot" que l'on passe au noyau. Il est dit :

```
If there is a "%s" token in the string, the token will be
      replaced by the ASCII-representation of the client's IP
      address.
```

Le fichier de configuration isolinux.cfg doit donc contenir :

```

default linux
prompt 0
label linux
KERNEL alt0/vmlinuz
APPEND root=/dev/nfs devfs=mount ip:::::dhcp \
      nfsroot=195.220.79.217:/diskless/tftpboot/%s

```

Ainsi, le noyau va remplacer %s par l'adresse IP qu'il a obtenue par DHCP. Le chemin sera donc formé correctement.

On peut même aller plus loin : la documentation de Linux dit que, dans l'option `nfsroot=`, si l'IP du serveur NFS n'est pas donnée, l'adresse du serveur DHCP est prise par défaut. Or, dans notre cas, le serveur DHCP et le serveur NFS du diskless sont la même machine.

On peut donc mettre :

```

default linux
prompt 0
label linux
KERNEL alt0/vmlinuz
APPEND root=/dev/nfs devfs=mount ip:::::dhcp nfsroot=/diskless/tftpboot/%s

```

Avec cette dernière solution, voici les étapes du boot qui seront effectuées :

1. Le PC diskless boote sur le CD
2. Isolinux est lancé
3. Isolinux donne la main à Linux en lui passant les paramètres indiqués dans le fichier de configuration `isolinux.cfg`
4. Linux démarre
5. Linux fait une requête DHCP car le paramètre `ip` contient la valeur `:::::dhcp`.
6. Un serveur DHCP lui donne son IP. Linux mémorise son adresse IP ainsi que l'adresse IP du serveur DHCP.
7. Linux demande le montage de sa partition racine au serveur NFS. Le serveur NFS utilisé est celui dont l'IP est celle du serveur DHCP. Le chemin d'accès sur le serveur est `/diskless/tftpboot/195.220.79.xxx/` car Linux a remplacé %s par son adresse IP.

Ainsi, le CD marchera quel que soit le diskless et quel que soit le serveur. Les seules contraintes sont :

1. Le serveur NFS et le serveur DHCP d'un même diskless doivent être la même machine (on ne peut pas utiliser deux machines, chacune étant spécialisée sur un service)

2. Le noyau étant sur le CD, il faudra changer le CD lors des mises à jour du noyau (ce qui ne devrait pas arriver très souvent). On ne peut donc pas mettre à jour le noyau juste sur le serveur, contrairement à la méthode de boot par PXE.

10.4.4 Mise au point

Pour mettre au point la méthode, j'ai dû effectuer de nombreux gravages. Mais utilisant un CD réinscriptible, aucun CD n'a été gaspillé.

10.4.5 Conclusion

Cette méthode marche tout aussi bien que la précédente et permet d'avoir un CD unique pour tout le LAOG.

11 Procédure pour créer les CD bootable

11.1 Principe

Isolinux est un chargeur de démarrage pour Linux sur CD. Il permet de charger un noyau Linux sans contrainte de taille. Isolinux est utilisé notamment sur les CD d'installation de Mandrake, Red Hat, Debian et sans doute d'autres. Isolinux fait partie du projet Syslinux.

Le site de isolinux : <http://syslinux.zytor.com/iso.php>

11.2 procédure

Tout ce qui est en rapport avec la création du CD sera mis sur le serveur dans `/diskless/CDROM`. Voici l'arborescence de `/diskless/CDROM` :

```
/diskless/CDROM
| - isos : les images des CD seront placees ici
| - root : correspond a la racine du CD
|   | - isolinux : contient le binaire d'isolinux et son fichier de config.
|   | - alt0 : contient le noyau vmlinuz
```

Cette arborescence est créée automatiquement par le script *dkl-init-serveur.sh*. Voici ce qui doit être présent dans `/diskless/CDROM/root/` :

1. le fichier `isolinux/isolinux.bin` issu de l'archive `syslinux`
2. le fichier de configuration `isolinux/isolinux.cfg`

Ces deux fichiers ont été copiés par *dkl-init-serveur.sh*. Les deux fichiers sont donc déjà présents. Si vous voulez récupérer la dernière version, téléchargez l'archive de `syslinux` sur <http://www.kernel.org/pub/linux/utils/boot/syslinux/>. Le fichier `isolinux.bin` se trouve dans l'archive.

La création de CD se fait en 5 étapes :

1. Copier le noyau linux.
2. Modifier le fichier de configuration d'isolinux.
3. Créer une image bootable du CD pour le PC diskless
4. Graver cette image.
5. Démarrer le diskless sur le CD.

11.3 Copier le noyau linux

Le noyau se trouve ici : /diskless/tftpboot/bzImage-diskless-pxe1. Il faut le copier dans l'arborescence du CD :

```
cp -a /diskless/tftpboot/bzImage-diskless-pxe1 \
      /diskless/CDROM/isolinux/alt1/vmlinuz
```

Par convention, on appelle le noyau vmlinuz car le nom est limité à 8 caractères.

11.4 Fichier de configuration d'isolinux

Il se trouve sous /diskless/CDROM/root/isolinux/isolinux.cfg. Pas besoin de le modifier, il est correct. Voici un exemple :

```
# Fichier de configuration isolinux.cfg

default linux
# Pour afficher le menu
#prompt 1
# Pour ne pas afficher le menu
prompt 0

#####
# CHOIX 1: "linux"
label linux
# A la racine du CDRM, repertoire isolinux/alt0
KERNEL alt0/vmlinuz
# A ce stade, le noyau est charge du CD en memoire
# Parametres passes au noyau pour booter sur le reseau
APPEND root=/dev/nfs devfs=mount ip:::::dhcp nfsroot=/diskless/tftpboot/%s

#####
# CHOIX 2: "linux-2"
label linux-2
# A la racine du CDRM, repertoire isolinux/alt2
```



```

mkisofs -o /diskless/CDROM/isos/iso1.iso      \
        -b isolinux/isolinux.bin             \
        -c isolinux/boot.cat                 \
        -no-emul-boot -boot-load-size 4     \
        -boot-info-table                     \
        /diskless/CDROM/root/

```

1. -o permet de choisir où mkisofs va mettre le fichier image du CD bootable.
2. -b permet d'indiquer à mkisofs où se trouve le fichier binaire isolinux.bin par rapport à la racine du CD. C'est ce fichier qui sera exécuté au démarrage du CD
3. -c permet d'indiquer à mkisofs où il doit créer le fichier boot.cat. Ce fichier n'existe pas, c'est normal : mkisofs va le créer et une fois le CD gravé, on le trouvera sur le CD.
4. -no-emul-boot permet de graver l'image sans émuler une disquette ou un disque pour démarrer le CD. En effet, isolinux fonctionne sans émulation.
5. -boot-load-size 4 et -boot-info-table sont des options pour créer l'image bootable. Je n'ai pas compris ce qu'elles faisaient exactement mais elles semblent indispensable pour créer l'image bootable.
6. Enfin, /diskless/CDROM/root/ indique ce qui doit être gravé.

Une fois le script *dkl-graver-cdrom.sh* lancé, l'image se trouve ici : */diskless/CDROM/isos/iso1.iso*

11.6 Graver l'image

L'image se trouve ici : */diskless/CDROM/isos/iso1.iso*. Elle peut être gravée par n'importe quel logiciel de gravage. Voici la méthode avec *cdrecord*.

Tout d'abord, taper "*cdrecord -scanbus*" pour savoir où se trouve le graveur de CD. Le résultat doit être ceci :

```
0,0,0    0) 'AOPEN    ' 'CD-RW CRW4048    ' '1.05' Removable CD-ROM
```

On voit donc que le graveur se trouve sur le périphérique "0,0,0".

Graver l'image avec ce commande (remplacer "0,0,0" si nécessaire) :

```
cdrecord dev=0,0,0 -eject /diskless/CDROM/isos/iso1.iso
```

S'il s'agit d'un CD-RW (réinscriptible), il faut utiliser la commande suivante pour effacer le CD avant de graver :

```
cdrecord dev=0,0,0 blank=fast -eject /diskless/CDROM/isos/iso1.iso
```

11.7 Démarrer le PC diskless sur le CD

Le BIOS du PC diskless doit être configuré de manière à booter sur le CD. Une fois ceci configuré, insérer le CD et redémarrer le PC diskless.

11.8 Résumé

1. Le PC diskless boote sur le CD
2. Isolinux est lancé.
3. Isolinux lit son fichier de configuration qui se trouve sur le CD.
4. Isolinux charge le noyau linux qui se trouve sur le CD.
5. Isolinux donne la main à Linux en lui passant les paramètres indiqués dans le fichier de configuration isolinux.cfg
6. Linux démarre
7. Linux fait une requête DHCP car le paramètre ip contient la valeur `:::::dhcp`.
8. Un serveur DHCP lui donne son IP. Linux mémorise son adresse IP ainsi que l'adresse IP du serveur DHCP.
9. Linux demande le montage de sa partition racine au serveur NFS. Le serveur NFS utilisé est celui dont l'IP est celle du serveur DHCP. Le chemin d'accès sur le serveur est `/diskless/tftpboot/195.220.79.xxx/` car Linux a remplacé %s par son adresse IP.

Quatrième partie

Installation du serveur de diskless : Manuel utilisateur

J'ai eu plusieurs machines à ma disposition pour effectuer des tests et mettre au point la méthodes. Pour voir la description de ces machines, allez voir en annexe la partie *Machines de tests*.

Pour faciliter l'installation, j'ai écrit des scripts qui sont accessibles en annexe.

12 Installation et configuration des services du serveur

Le serveur de diskless devra avoir les services suivants :

1. DHCP
2. NFS
3. TFTP (utile seulement pour les boots par PXE ou GrubPxe)

12.1 Configuration du serveur DHCP

DHCPD est un processus tournant tâche de fond qui assigne à toute machine sur un réseau TCP/IP qui le demande une adresse IP. Il fournit aussi certaines informations sur le réseau. L'adresse IP est fournie pour une durée déterminée dans la configuration.

C'est le démon qui est à l'écoute des appels (requête "dhcp" par l'intermédiaire d'un broadcast) sur le réseau, provenant des PC clients. Le serveur tournant dhcpd regarde dans son fichier /etc/dhcpd.conf s'il contient une entrée correspondant à l'adresse ethernet (MAC) du PC. Il lui retourne l'adresse IP indiquée dans la table.

Installer le serveur DHCP. Les paquetages à installer sont :

1. dhcp-server
2. dhcp-common
3. dhcpd
4. dhcp-client

Il faut aussi installer la partie client car, même si le serveur n'en a pas besoin, les PC diskless en ont besoin et le système des PC diskless va être récupéré à partir du serveur.

Un autre fichier - /var/lib/dhcp/dhcpd.leases - doit être créé pour que DHCPD puisse y stocker les informations sur les "locations" d'adresses actuelles. Pour le créer, taper simplement la commande

```
touch /var/lib/dhcp/dhcpd.leases
```

Voici une partie du fichier de configuration /etc/dhcpd.conf pour un diskless :

```
# Diskless de test numero 2
host gag242 {
    allow booting;
    hardware ethernet 00:d0:b7:c4:dc:ba;
    fixed-address 195.220.79.242;
}
```

12.2 Configuration du serveur NFS

NFS veut dire Network File System. Le support du serveur nfs doit être inclus dans le noyau.

Pour le démarrer : /etc/rc.d/init.d/nfs start ==> rpc.portmap et rpc.mountd vérifier que le serveur tourne : ps ex | grep nfsd

Si le nombre de PC diskless à servir est grand, modifier, le fichier /etc/init.d/nfs et passer NFSDCOUNT à 16 (le défaut est 8).

Après le chargement du noyau par tftp ou par CD, un système de fichiers racine doit être fourni. Linux utilise le protocole NFS (qui fait partie

intégrante du noyau qui vient d'être chargé) pour voir le système de fichiers racine.

NFS-Root permet d'utiliser une partition NFS comme partition racine. Cette possibilité est particulièrement utile pour une station dépourvue de disque local (diskless)

Installer et configurer le serveur NFS. Voici une partie du fichier de configuration `/etc/exports` pour deux diskless :

```
# /etc/exports
/diskless      195.220.79.220(rw,no_root_squash,insecure,async)
/diskless      195.220.79.242(rw,no_root_squash,insecure,async)
```

12.3 Configuration du serveur TFTP

TFTP est utile uniquement si certain PC diskless bootent par PXE ou par GrubPxe.

TFTP = FTP sans authentification et tournant sur UDP : transfert bloc par bloc et non par flux comme tcp

TFTP va permettre au PC diskless qui a obtenu son adresse IP par dhcp, de télécharger l'image de son OS, bloc par bloc, jusqu'à ce que tout le fichier soit transféré. Quand l'OS est entièrement chargé, la ROM de démarrage réseau, donne la main à l'image du système d'exploitation au point d'entrée.

Un exemple d'installation, de configuration et de "debug" de tftp :

http://sari.inpg.fr/rubriques/themes/zone_publice.groupe_linux/diskless/tftp-hpa-install

13 Installation dans /diskless

On choisit de recopier l'image complète du système linux serveur, dans une partition réservée que l'on nommera /diskless, dans laquelle on va construire l'arborescence racine nécessaire à chaque diskless. Il faut donc prévoir une partition dont la taille sera égale à la taille du système linux serveur plus la taille d'un /usr/local pouvant accueillir les logiciels communs à tous les diskless soit environ 2Go, plus environ 100 Mo par diskless, plus une zone commune système d'environ 100Mo. Il est raisonnable de prévoir une zone de 10 a 20Go pour être à l'aise.

Cette solution va permettre de rendre l'ensemble des diskless indépendants du système serveur, de porter cette partition sur d'autres plateformes qu'un PC linux (par exemple, l'UNIX d'IBM : AIX). L'ensemble des logiciels installés sous /usr/local ou sous /opt seront partageables entre tous les diskless, ainsi que les bibliothèques.

13.1 Création des répertoires de /diskless

Il faut créer /diskless :

```
mkdir -p /diskless
```

Les scripts d'installation seront placés dans /diskless/scripts. Téléchargez-les dans /diskless/scripts (Voir l'adresse en annexe).

Il faut ensuite lire et lancer le script *dkl-init-serveur.sh*.

Ce script prépare l'arborescence /diskless :

1. Prépare /diskless/tftpboot avec quelques fichiers exemples.
2. Prépare /diskless/CDROM pour qu'un CD puisse être gravé.
3. Copie les répertoires /bin, /usr... communs à tous les diskless.
4. Copie les fichiers communs à tous les diskless (/etc/passwd par exemple).

Mais avant de lancer ce script, il faut être sûr d'avoir installé sur le serveur les paquetages qui seront utilisés sur les PC diskless (car les paquetages installés sur les PC diskless viennent de la copie du serveur).

En particulier, il faut :

1. kernel-source
2. dhcp-client

14 Installer le noyau du diskless

14.1 Faire d'abord une sauvegarde

Installer les sources du noyau :

```
urpmi kernel-source
```

Vérifier que les sources se trouvent bien dans /usr/src/linux. Sauvegarder /boot, les modules du serveur et les sources :

```
cp -a /boot /boot.sav
cp -a /lib/modules/2.4.19-16mdk /lib/modules/2.4.19-16mdk.sav
cp -a /usr/src/linux-2.4.19-16mdk /usr/src/linux-2.4.19-16mdk.sav
```

Modifier /etc/lilo.conf ou /boot/grub/menu.lst : remplacer les vmlinuz, initrd, ... qui sont des liens symboliques par le nom réel des fichiers : vmlinuz-2.4.19-16mdk, initrd-2.4.19-16mdk.img, ...

Ajouter une stanza dans lilo.conf ou menu.lst qui permettra de rebooter au cas où le noyau serait endommagé. Par ex. :

```
image=/boot.sav/vmlinuz-2.4.19-16mdk
    label=linux.sav
    root=/dev/sda1
    initrd=/boot.sav/initrd-2.4.19-16mdk.img
    append="quiet devfs=mount"
    vga=788
    read-only
```

Si vous utilisez lilo, n'oubliez pas de taper lilo. Pour Grub, il n'y a rien à faire.

Tester pour voir si ca démarre bien.

14.2 Construire le noyau et les modules

Les sources du noyau avaient été installés sur le serveur et la copie du système sous /diskless, fait que ces sources sont sous /diskless/usr/src.

```
cd /diskless/usr/src
cp -a linux-2.4.19-16mdk linux-2.4.19-16mdk-diskless-pxe1
cd linux-2.4.19-16mdk-diskless-pxe1
cp -a .config .config.serveur
```

On va travailler à partir du .config qui deviendra celui du diskless. Il faudra mettre dans le noyau les paramètres suivants :

1. "Network Device support". intègre dans le noyau toutes les cartes réseaux installées sur les PC du site : le pilote de la carte réseau d'un PC diskless, doit être compilé dans le noyau et non en module.
2. "Networking options". IP : Kernel level autoconfiguration (CONFIG_IP_PNP=y); IP : BOOTP support (CONFIG_IP_PNP_BOOTP=y). Cela permettra le boot sur le réseau.
3. "File systems" / "Network File Systems". NFS file system support (CONFIG_NFS_FS=y) : permet d'activer CONFIG_ROOT_NFS; Provide NFSv3 client support (CONFIG_NFS_V3=y); Root file system on NFS (CONFIG_ROOT_NFS=y). Cela va permettre au noyau linux de monter la racine du système de fichiers par NFS lors du boot.

Pour compiler le noyau :

1. make mrproper
2. Modifier le fichier Makefile (EXTRAVERSION = -16mdk-diskless-pxe1)
3. make xconfig
4. Mettre dans le noyau les paramètres cités.
5. make dep
6. make bzImage
7. make modules
8. make modules_install
9. cp -a arch/boot/i386/bzImage /diskless/tftpboot/bzImage-diskless-pxe1
10. cp -a /lib/modules/2.4.19-16mdk-diskless-pxe1 /diskless/lib/modules
11. cd /diskless/lib/modules/2.4.19-16mdk-diskless-pxe1/
12. rm build

13. `ln -s /diskless/usr/src/linux-2.4.19-16mdk-diskless-pxe1/ build`

Le noyau se trouve maintenant dans `/tftpboot/bzImage-diskless-pxe1`. Selon les méthodes de boot, il devra être copié dans les emplacements suivants :

1. Boot par PXE, GrubPxe : le laisser dans `/diskless/tftpboot/bzImage-diskless-pxe1`
2. Boot à partir d'un CD : le copier dans `/diskless/CDROM/root/isolinux/alt0/vmlinuz`
3. Boot à partir du disque dur : le copier sur le disque du PC diskless dans `/grub/vmlinuz-diskless`

Dans tous les cas, les modules restent sur le serveur de diskless.

15 Création des diskless : modèle et clones

15.1 Création du modèle

Il faut lancer le script `dkl-init-diskless.sh` avec comme premier argument l'adresse IP du PC diskless. Par exemple :

```
/diskless/scripts/dkl-init-diskless.sh 195.220.79.220
```

Voici ce que fait le script :

1. créer la racine `/tftpboot/adresse-IP` du diskless
2. créer l'architecture des répertoires du diskless modèle
3. copier des répertoires propres au diskless et qui ne seront pas communs à tous les diskless
4. créer les répertoires sous `/mnt`

15.2 Fichiers de configuration propres au diskless

Aller dans le répertoire du diskless puis adapter ces fichiers aux paramètres du diskless (adresse IP, modules, disque dur...)

1. `/etc/sysconfig/network` (modifier le `HOSTNAME` -> XXX)
2. `/etc/sysconfig/network-scripts/ifcfg-eth0` (modifier `IPADDR` -> XXX)
3. `/etc/sysconfig/keyboard`
4. `/etc/modules.conf`
5. `/etc/modules`
6. `/etc/X11/XF86Config-4`
7. `/etc/fstab` (modifier l'IP du diskless pour le montage de `"/`)

Voici deux exemples de fichier /etc/sysconfig/keyboard :
Clavier américain :

```
KBCHARSET=iso-8859-15  
KEYBOARD=us_intl  
KEYTABLE=us-latin1
```

Clavier français :

```
KBCHARSET=iso-8859-15  
KEYBOARD=fr  
KEYTABLE=fr-latin1
```

Des exemples de /etc/X11/XF86Config-4 se trouvent en annexe. Voici ce qu'il faut penser à modifier :

1. *Clavier* : Option "XkbLayout" "fr" : Mettre "fr" ou "us_intl" selon le clavier utilisé.
2. *Carte graphique* : Driver "nvidia" : Mettre "nvidia" ou "ati" selon le driver utilisé. On pourra lancer xf86config pour voir la liste complète.
3. *Souris* : Voir les deux exemples ci-dessous.

Souris sans molette :

```
Section "InputDevice"  
    Identifier "Mouse1"  
    Driver "mouse"  
    Option "Protocol" "PS/2"  
    Option "Device" "/dev/psaux"  
    Option "Emulate3Buttons"  
    Option "Emulate3Timeout" "50"  
EndSection
```

Souris avec molette :

```
Section "InputDevice"  
    Identifier "Mouse1"  
    Driver "mouse"  
    Option "Protocol" "IMPS/2"  
    Option "Device" "/dev/psaux"  
    Option "ZAxisMapping" "4 5"  
EndSection
```

Il faut aussi adapter le fichier /etc/fstab du diskless. Voici un exemple :

```

# /etc/fstab du diskless

# 195.220.79.217 est l'adresse IP du serveur
# 195.220.79.220 est l'adresse IP du diskless

195.220.79.217:/diskless/tftpboot/195.220.79.220 / nfs defaults,bg,intr,noac,rsize=8192,
0 0
195.220.79.217:/diskless/shares/bin /bin nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0
195.220.79.217:/diskless/shares/usr /usr nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0
195.220.79.217:/diskless/shares/sbin /sbin nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0
195.220.79.217:/diskless/shares/home /home nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0
195.220.79.217:/diskless/shares/lib /lib nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0
195.220.79.217:/diskless/shares/commun /commun nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0
195.220.79.217:/diskless/shares/opt /opt nfs defaults,bg,intr,noac,rsize=8192,wsiz
0 0

none /dev/pts devpts mode=0620 0 0
none /proc proc defaults 0 0

# DISQUETTES ET CDROM
#none /mnt/cdrom supermount dev=/lib/dev-state/ide/host2/bus1/target1/lun0/cd,fs=auto,ro
#OU: none /mnt/cdrom supermount dev=/dev/cdrom,fs=auto,ro,--,iocharset=iso8859-15,codepage
#none /mnt/cdrom2 supermount dev=/lib/dev-state/ide/host2/bus1/target0/lun0/cd,fs=auto,r
#none /mnt/floppy supermount dev=/dev/fd0,fs=auto,--,iocharset=iso8859-15,sync,codepage=

# DISQUES DURS LOCAUX
#/dev/ide/host2/bus0/target0/lun0/part1 /mnt/disk ext3 defaults 0 0
#/dev/ide/host2/bus0/target0/lun0/part3 /local ext3 defaults 0 0
#/dev/ide/host2/bus0/target0/lun0/part2 swap swap defaults 0 0

```

Remarque :

Le mini-système initialement créé sous la racine /diskless/tftpboot/195.220.79.XXX par dkl-init-diskless.sh est indispensable au boot du PC diskless, à savoir : usr, lib, bin, sbin. Il va être supplanté par le système complet lors du montage NFS des partitions usr, lib, ... indiquées dans fstab et qui elles correspondent au système Linux copié sous /diskless et commun à tous les PC diskless.

Le fait que les fichiers de /bin (par exemple) soient accessibles à la fois par /diskless/bin et par /diskless/tftpboot/195.220.79.xxx/bin n'est pas gênant :

en effet, grâce aux liens durs, cela ne prend pas plus de place.

De plus, il est indispensable de laisser ces fichiers dans ces deux endroits :

1. Dans /usr à cause de rpm.
2. Dans /tftpboot/195.220.79.xxx/usr pour le démarrage quand les montages ne sont pas encore faits.

15.3 Faire du ménage

Faire du ménage avec `dkl-diskless-menage.sh` pour nettoyer /tmp, /root, /var/tmp, etc.

Ne pas lancer ce script quand le PC diskless tourne!

15.4 Placer les liens durs

Il faut ensuite lire et lancer le script `dkl-liens-durs-1diskless.sh` avec comme premier argument l'adresse IP du PC diskless. Par exemple :

```
/diskless/scripts/dkl-liens-durs-1diskless.sh 195.220.79.220
```

Après ce script, tous les fichiers de /usr, ... du PC diskless seront mis en commun par des liens durs avec les autres PC diskless.

15.5 Reconfigurer le serveur

Voici ce qu'il faut penser à faire sur le serveur :

1. Récupération de l'adresse MAC du nouveau diskless
2. MAJ de /etc/dhcp.conf et redémarrer DHCP (/etc/init.d/dhcpd restart)
3. MAJ de /tftpboot/pxelinux.cfg/
4. MAJ /etc/exports et recharger NFS (exportfs -a)
5. MAJ de /etc/hosts et de /diskless/share/commun/hosts
6. Si le diskless utilise auto.master et *.map, configurer les différents serveurs pour accepter le nouveau diskless.

15.6 Cloner un diskless

Le but est de rajouter un nouveau diskless dans /diskless/tftpboot/ à partir d'un modèle.

Si le diskless modèle s'appelle `195.220.79.modele` et si le nouveau diskless s'appelle `195.220.79.242`, tapez :

1. `cp -a /diskless/tftpboot/195.220.79.modele /diskless/tftpboot/195.220.97.242`
2. `dkl-liens-durs-1diskless.sh`

3. `dkl-diskless-menage.sh`

Il faut ensuite reprendre les étapes de la création du modèle mais cette fois pour le nouveau diskless (le diskless clone : 195.220.79.242 par ex.) :

1. Fichiers de configuration propres au diskless
2. Faire du ménage
3. Placer les liens durs
4. Reconfigurer le serveur

Cinquième partie

Partage des fichiers systèmes entre les diskless

16 Arborescence UNIX

Sous Linux, les données systèmes sont classées dans des répertoires dont les noms sont standard. On trouve la même arborescence de base sur tous les PC Linux.

1. `/bin` : contient tous les programmes de base. Aucun fichier utilisateur ou fichier de configuration ne se trouve ici. Ce répertoire peut donc être commun à tous les diskless
2. `/boot` : contient souvent le noyau et quelques fichiers de configuration de boot comme Grub. Mais dans notre cas, on ne démarre pas le PC diskless à partir du disque. Ce répertoire n'existe donc pas sur le PC diskless.
3. `/dev` : contient les fichiers de type périphériques. Ce répertoire est automatiquement rempli au démarrage grâce à `devfs`. Il suffit donc juste de créer un répertoire `/dev` vide pour les diskless.
4. `/etc` : contient tous les fichiers de configuration propre à la machine. Ce répertoire ne peut donc pas être partagé entre les PC diskless car chaque PC diskless peut être différent.
5. `/home` : contient les fichiers des utilisateurs. Ce répertoire doit être commun sur tous les PC diskless car un utilisateur doit pouvoir se connecter sur n'importe quel PC diskless et retrouver ses fichiers.
6. `/lib` : contient les bibliothèques des programmes du système et les modules du noyau. Ce répertoire peut être partagé entre les PC diskless.
7. `/mnt` : permet de monter une disquette ou un CD. Ce répertoire ne contient en fait aucun fichier.
8. `/proc`

9. /root
10. /sbin : comme /bin mais ce répertoire contient normalement les programmes réservés à l'administrateur.
11. /tmp : espace réservé aux fichiers temporaires.
12. /usr
13. /var

Voir :

<http://www.freenix.fr/unix/linux/fsstnd-fr/>

17 Idée de base

L'idée de base est de créer une partition nommée ici "/diskless" sur une plateforme hôte quelconque qui hébergera l'image d'un système linux indépendant de son propre système.

La machine hôte de cette partition (appelée serveur de diskless), exportera par NFS cette partition vers les PC diskless.

Sous /diskless, on va construire l'arborescence nécessaire au boot de chaque client diskless, en limitant au maximum sa taille. Chaque arborescence exportée par NFS depuis le serveur deviendra la racine du client diskless, car "NFS-Root" (glossaire page ??) permet d'utiliser une partition NFS comme partition root (/). Chaque PC diskless bootera à l'aide du protocole PXE de sa carte réseau ethernet ou de son disque dur ou encore un CD. Il obtiendra son adresse IP du serveur de diskless par DHCP et importera par NFS les fichiers nécessaires à la suite de son démarrage.

18 Principe général

Pour gagner de la place sur le serveur, il n'est pas nécessaire de faire une copie de l'arborescence complète de chaque diskless. Par exemple, le répertoire /usr des diskless peut être commun pour tous les diskless.

Voici la liste de ce qui peut être commun : /usr, /bin, /sbin, /home, /opt, et /lib.

Voici ce qui ne peut pas être commun : /dev, /etc, /mnt, /proc, /root, /tmp, et /var

19 Méthode de mise en commun

La méthode utilisée pour partager le répertoire /usr (et les autres cités ci-dessus) pour tous les diskless est la suivante :

1. /usr et les autres répertoires se trouve à un endroit unique sur le serveur : /diskless/usr

2. /diskless/usr est partagé par NFS et est monté sous /usr

De plus, certains fichiers des répertoires non partagés *doivent* être communs. Ces fichiers se trouvent dans /diskless/tftpboot/commun sur le serveur. Le répertoire commun est monté par NFS sur les diskless avec cette ligne du fichier de configuration /etc/fstab :

```
195.220.79.217:/diskless/tftpboot/commun /commun nfs \
defaults,bg,intr,noac,rsize=8192,wsize=8192 0 0
```

Voici la liste de ces fichiers :

1. passwd, group, shadow, sudoers, hosts, printcap, termcap, ld.so.conf, ld.so.cache. Ces fichiers doivent se trouver sous /etc sur le diskless.

Les fichiers de cette liste sont dans /diskless/tftpboot/commun sur le serveur. La première idée est de créer des liens symboliques permettant de les trouver. Prenons le cas de /etc/passwd.

Du point de vue du diskless, le fichier /etc/passwd est un lien symbolique vers /commun/passwd et /commun est le point de montage du répertoire /diskless/tftpboot/commun sur le serveur.

Mais cette méthode pour partager les fichiers et répertoires pour les diskless pose quelques problèmes. En effet, au démarrage du diskless, tous les montages NFS ne sont pas encore montés et les scripts de démarrage ont besoin des fichiers qui se trouvent dans /commun.

En effet, /commun n'est pas encore monté et donc /commun/passwd n'existe pas encore pour le diskless. Le lien symbolique /etc/passwd du diskless n'est donc pas valide.

Une fois /commun monté, il n'y a plus de problème car le lien symbolique devient fonctionnel. Mais avant cela, il y a des problèmes lors de l'exécution des scripts de démarrage.

Pour les fichiers (/etc/passwd du diskless par exemple), la solution est de créer un lien dur sur le serveur entre /diskless/tftpboot/commun/passwd et tous les fichiers /etc/passwd de tous les diskless.

La commande utilisée pour le diskless gag220 est la suivante (bien sûr, cette commande est exécutée sur le serveur) :

```
ln /diskless/tftpboot/commun/passwd /diskless/tftpboot/195.220.79.220/etc/passwd
```

De cette façon, /etc/passwd est directement accessible par le diskless.

20 Les fichiers d'authentification

Les fichiers d'authentification sont : /etc/passwd, /etc/group, /etc/shadow, /etc/sudoers.

Ceux-ci doivent absolument être partagés entre les diskless. En effet, une personne doit pouvoir se loguer sur n'importe quel diskless et retrouver son compte.

De plus, il faut que quand on modifie ces fichiers (nouveau compte, changement de mot de passe...) la modification se fasse automatiquement.

La solution choisie est de lier ces fichiers entre eux par des liens durs. On ne peut pas faire des liens symboliques car cela ne marche pas dans notre cas. En effet, la résolution des liens symboliques se fait du côté du client (diskless) et les fichiers pointés par les liens symboliques ne sont pas accessibles au démarrage du diskless : le montage NFS n'est pas encore fait.

21 Les fichiers utilisateurs

Pas de problème particulier puisque /home n'est utilisé que lorsque la machine a fini de démarrer.

/home est monté par NFS.

22 Les modules du noyau

Les modules du noyau du diskless sont stockés sur le serveur. Ils prennent de la place car :

1. Le noyau est compilé avec un maximum de modules possible. Il n'est pas envisageable de chercher à réduire le nombre de modules car l'ensemble de ces modules sera utilisé sur un parc hétérogène de PC diskless avec du matériel hétérogène et on ne connaît pas à l'avance les composants du diskless. On veut donc un noyau généraliste avec un maximum de fonctionnalités.
2. Les modules sont copiés dans les répertoires de tous les diskless (/diskless/tftpboot/195.220.79.xxx/lib/modules).

Ce deuxième point doit être corrigé. Voici l'historique du placement des modules sur le serveur des diskless.

22.1 Modules dans /diskless/lib/modules

Lors de mes premiers tests, j'ai placé les modules dans le répertoire */diskless/lib/modules/* du serveur. Le répertoire */diskless/lib* du serveur est monté par NFS sur le répertoire */lib* du diskless.

Cela marchait à peu près mais il y avait un gros problème : dans les scripts de démarrage, le diskless lance *depmod*. Ce programme sert à recenser les modules pour pouvoir les charger plus facilement avec *modprobe*.

Mais au moment où *depmod* s'exécute, le répertoire */lib* du diskless n'est pas encore monté par NFS. *depmod* ne trouvait donc aucun module. Sur mon diskless de test, cela pose deux problèmes :

1. *modprobe* ne marche pas (car *modprobe* a besoin de *depmod* dont l'exécution n'a pas marché au démarrage du PC diskless) Cependant, *insmod* marche toujours mais ce n'est pas satisfaisant.

2. Les modules listés dans */etc/modules* ne sont pas chargés automatiquement. En effet, ceux-ci sont chargés par un script au démarrage avec *modprobe* et *modprobe* ne marche pas. Une conséquence de ce problème est que XFree ne marchait pas car le module de NVidia ne se chargeait pas automatiquement.

22.2 Modules dans */diskless/lib/modules* et dans les répertoires des diskless

Pour corriger ces problèmes, j'ai copié les modules dans tous les répertoires de tous les diskless :

```
/diskless/tftpboot/195.220.79.xxx/lib/modules/
```

Ainsi, les modules se trouvent à la fois sous */diskless/lib/modules* et dans tous les répertoires de tous les diskless. En effet, si on ne les met que dans les répertoires des diskless, le diskless ne trouve plus ses modules car le montage NFS */lib* vient cacher les fichiers de */lib*. On doit cependant garder un point de montage NFS pour */lib* car ce répertoire ne contient pas que les modules et il faut que le reste des fichiers soient communs à tous les diskless.

Cela prend donc beaucoup de place. De plus, la maintenance est plus difficile puisqu'à chaque modification des modules on doit les copier dans autant de répertoires qu'il y a de diskless.

Une autre solution doit donc être trouvée.

22.3 Modules dans */diskless/lib/modules* et liens durs

Pour corriger ces nouveaux problème, j'ai proposé une nouvelle solution. Elle corrige le problème de place.

Les modules doivent se placer dans */diskless/lib/modules* et tous les diskless "possèdent" un lien dur vers chacun des modules du noyau.

Ainsi, */diskless/tftpboot/195.220.79.xxx/lib/modules/.../* et */diskless/lib/modules/.../* pointent sur le même inode et désignent donc le même fichier.

Un script se chargera de synchroniser */diskless/lib/modules* et le répertoire de modules de chacun des diskless.

Le besoin de synchronisation est justifié par :

1. Lors de l'ajout d'un diskless, il ne faut pas copier les modules mais faire des liens durs.
2. Lors de l'ajout d'un module, il faut créer tous les liens durs correspondants à chaque PC diskless.
3. Lors de la suppression d'un module, il faut supprimer tous les liens. En effet, le supprimer de */diskless/lib/modules* ne suffit pas à libérer l'espace disque puisque des liens existent encore vers le fichier.

Pour résumer, ce script contient la ligne suivante :

```
cp -al /diskless/lib/modules/ /diskless/tftpboot/195.220.79.xxx/lib/
```

L'option -a permet de garder la structure et les attributs du fichier original lors de la copie. C'est une option indispensable pour les modules.

L'option -l permet de ne pas copier les fichiers mais de faire des liens durs à la place.

En fait, ce script contient d'autres choses mais l'idée est résumée dans cette ligne.

Les tests que j'ai effectués montrent que cette méthode marche correctement. Il reste à tester les performances sur un cas réel : je ne sais pas si ça va aller plus vite ou pas...

22.4 /lib en lien dur

Maintenant, tous le répertoire /lib du PC diskless contient des liens durs "vers" les fichiers correspondant dans /diskless/lib du serveur.

23 Manipulation des liens durs

Tous les fichiers communs entre les diskless se font avec les liens durs. Pour modifier la configuration de ces liens durs, on ne peut pas tout faire à la main car :

1. Il y a trop de diskless.
2. Il y a trop de fichiers.

Les modifications se font à l'aide de scripts :

1. Mise en place de tous les liens durs sur tous les diskless : *dkl-liens-durs.sh*
2. Mise en place de tous les liens durs sur un diskless : *dkl-liens-durs-1diskless.sh*
3. Mise en place d'un lien dur sur tous les diskless : *dkl-1lien-dur-diskless.sh*
4. Mise en place d'un lien dur sur un diskless : *dkl-1lien-dur-1diskless.sh*
5. Suppression d'un lien dur sur tous les diskless : *dkl-suppr-1lien-dur-diskless.sh*
6. Suppression d'un lien dur sur un diskless : *dkl-suppr-1lien-dur-1diskless.sh*

23.1 Fonctionnement

Le fichier *dkl-liens-durs-fonctions.sh* contient des fonctions en shell permettant de modifier les liens durs. Les 6 scripts pré-cités appellent ces fonctions pour effectuer ces modifications.

23.2 dkl-liens-durs.sh

Mise en place de tous les liens durs sur tous les diskless.

Exemple : *dkl-liens-durs.sh* (ce script s'appelle sans argument)

1. Recherche de tous les diskless installés (regarde dans /diskless/tftpboot)
2. Créé les liens durs sur les répertoires /lib, /usr, /bin, /sbin.
3. Créé les liens durs sur les fichiers passwd, group, shadow, auto.master, *.map, ...

23.3 dkl-liens-durs-1diskless.sh

Mise en place de tous les liens durs sur un diskless.

Exemple : *dkl-liens-durs-1diskless.sh 195.220.79.220*

1. Modifie uniquement /diskless/tftpboot/195.220.79.220
2. Créé les liens durs sur les répertoires /lib, /usr, /bin, /sbin.
3. Créé les liens durs sur les fichiers passwd, group, shadow, auto.master, *.map, ...

23.4 dkl-1lien-dur-diskless.sh

Mise en place d'un lien dur sur tous les diskless.

Exemple : *dkl-1lien-dur-diskless.sh /diskless/shares/commun/passwd /etc/passwd*

1. Recherche de tous les diskless installés (regarde dans /diskless/tftpboot)
2. Créé tous les liens durs de /diskless/shares/commun/passwd sur /diskless/tftpboot/195.220.79.*/etc/passwd.

23.5 dkl-1lien-dur-1diskless.sh

Mise en place d'un lien dur sur un diskless.

Exemple : *dkl-1lien-dur-1diskless.sh 195.220.79.220 /diskless/shares/commun/passwd /etc/passwd*

1. Modifie uniquement /diskless/tftpboot/195.220.79.220
2. Créé le lien dur de /diskless/shares/commun/passwd sur /diskless/tftpboot/195.220.79.220/etc/passwd.

23.6 `dkl-suppr-1lien-dur-diskless.sh`

Suppression d'un lien dur sur tous les diskless :

Exemple : `dkl-suppr-1lien-dur-diskless.sh /etc/passwd`

1. Recherche de tous les diskless installés (regarde dans `/diskless/tftpboot`)
2. Détruit tous les fichiers `/diskless/tftpboot/195.220.79.* /etc/passwd`.

23.7 `dkl-suppr-1lien-dur-1diskless.sh`

Suppression d'un lien dur sur un diskless :

Exemple : `dkl-suppr-1lien-dur-1diskless.sh 195.220.79.220 /etc/passwd`

1. Modifie uniquement `/diskless/tftpboot/195.220.79.220`
2. Détruit le fichier `/diskless/tftpboot/195.220.79.220 /etc/passwd`.

Sixième partie

Installation de logiciels sur les PC diskless

Une fois que les PC diskless marchent, ce n'est pas terminé. Il faut pouvoir mettre à jour les logiciels. La méthode habituelle sur Linux Mandrake est d'utiliser les outils rpm et urpmi. Cette méthode est parfaitement adaptée aux ordinateurs avec une installation classique sur le disque dur. Dans le cas des PC diskless, cette méthode pose quelques problèmes et doit être adaptée.

24 Qu'est-ce que RPM et URPMI ?

RPM et URPMI sont les équivalents de dpkg et apt-get respectivement pour Mandrake.

Le but est de faire marcher RPM et URPMI sur les PC diskless de telle façon qu'on puisse installer une fois le paquetage et qu'il soit installé sur tous les PC diskless.

La méthode doit être la plus automatique et la plus simple possible.

24.1 Principe de URPMI

Lorsqu'on demande à URPMI d'installer un logiciel, il cherche ce logiciel dans les sources de paquetages installées sur la machine. Les sources de paquetages peuvent être de différents types : CD-ROM d'installation ou Internet.

Certaines sources sont des sources officielles de Mandrakesoft (main, contrib, updates), d'autres n'ont rien à voir avec Mandrakesoft (plf, textstar).

Certaines sources sont statiques (leur contenu ne change pas), d'autres sont des sources de mises à jour.

Il faut donc installer des sources avant de commencer. A l'installation de Mandrake, les seules sources configurées sont les CD-ROM d'installation. Il faut donc installer les sources Internet.

24.2 Comment installer les sources la première fois

Méthode simple :

1. Aller sur le site <http://plf.zarb.org/nanardon/urpmiweb.php>
2. Sélectionner les sources (prendre des sites proches pour télécharger plus vite)
3. Taper la commande indiquée sur la page.

24.3 Mises à jour automatiques

Il y a deux choses à faire :

1. garder les sources de paquetages à jour
2. Mettre à jour les paquetages

Voici un script qui permet de faire ca :

```
#!/bin/sh
unset ftp_proxy
unset http_proxy
urpmi.update main updates ftp2 2>> /var/log/updates
urpmi --auto-select --auto --no-verify-rpm >> /var/log/updates
```

Attention, ce script n'est pas testé!!

24.4 Liens utiles sur URPMI

Page officielle de URPMI :

<http://www.linux-mandrake.com/cooker/urpmi.html>

25 Problèmes

25.1 NFS lock

Lors de mon premier essai de ces commandes sur un PC diskless, rpm ou urpmi se figent et ne font rien.

Ce problème est maintenant résolu. Voici comment je l'ai résolu. Pour résoudre le problème, je tape la commande suivante :

```
# strace rpm -qa
```

La commande strace permet de montrer tous les appels systèmes faits par la commande "rpm -qa". Cela affiche en particulier les lignes suivantes :

```
open("/var/lib/rpm/Packages", O_RDONLY|O_LARGEFILE) = 3
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
fstat64(3, {st_mode=S_IFREG|0644, st_size=14807040, ...}) = 0
_llseek(3, 0, [0], SEEK_SET) = 0
read(3, "\0\0\0\0\1\0\0\0\0\0\0\0a\25\6\0\7\0\0\0\0\20\0\0\0\10"... , 256) = 256
close(3) = 0
brk(0) = 0x8071000
brk(0x8072000) = 0x8072000
open("/var/lib/rpm/Packages", O_RDONLY|O_LARGEFILE) = 3
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
fstat64(3, {st_mode=S_IFREG|0644, st_size=14807040, ...}) = 0
brk(0) = 0x8072000
brk(0x8073000) = 0x8073000
fcntl64(3, F_SETLK, {type=F_RDLCK, whence=SEEK_SET, start=0, len=0}) = -1 ENOLCK (No lock available)
```

Voici comment interpréter ces lignes :

1. rpm ouvre le fichier /var/lib/rpm/Packages avec l'appel système open (voir la première ligne). Cette fonction renvoie le descripteur de fichier associé. Dans notre cas, c'est 3.
2. rpm veut verrouiller le fichier avec l'appel système fcntl64 (voir dernière ligne). Il s'agit bien du fichier /var/lib/rpm/Packages car le premier paramètre de fcntl64 est 3, c'est à dire le descripteur de fichier correspondant au fichier ouvert.

La raison est donc que rpm veut verrouiller le fichier mais ne peut pas le faire car la fonction renvoie "ENOLCK (No locks available)".

Ce problème est apparu sur le PC diskless, quand le service NFS lockd n'avait pas réussi à démarrer. C'est donc normal que rpm ne puisse pas verrouiller le fichier /var/lib/rpm/Packages car ce fichier est accédé par NFS.

Pour corriger le problème, il suffit donc de démarrer NFS lockd. Pour le démarrer, tapez la commande suivante :

```
/etc/init.d/nfslock start
```

En fait, NFS lockd se lance au démarrage automatiquement. S'il ne le fait pas, allez voir la partie "NFS lockd".

25.2 Partages entre diskless

25.2.1 Détails du problème

Pour mon premier test de urpmi sur un PC diskless, j'ai choisi d'installer ce paquetage : fluxbox-0.1.14-6mdk.i586.rpm. Fluxbox est un gestionnaire de fenêtre basé sur blackbox. Il est très léger et possède plusieurs fonctionnalités intéressantes.

L'installation se fait depuis le PC diskless gag220 et se déroule sans problème avec la commande suivante :

```
urpmi fluxbox
```

Une fois installé, je regarde la liste des fichiers contenus dans le paquetage.

```
rpm -qli fluxbox
```

D'après cette commande, les fichiers sont placés dans :

1. /etc
2. /usr

Ce qui devait arriver arriva :

1. Les fichiers dans /usr sont bien disponibles sur tous les diskless car /usr est commun à tous les PC diskless via un montage NFS.
2. Les fichiers dans /etc ne sont accessibles que sur le PC diskless d'installation, c'est-à-dire, dans cet exemple, gag220.

On a donc un gros problème : les autres PC diskless ne veront pas tous les fichiers.

25.2.2 Une piste : URPMI-parallel

URPMI a une fonctionnalité nommée "-parallel" qui permet de lancer urpmi automatiquement sur plusieurs machines pour installer un paquetage sur plusieurs machines en même temps.

Cette fonctionnalité a été développée pour être utilisée avec Clic mais elle peut en fait être utilisée sur n'importe quelle distribution Linux.

Une documentation de cette fonctionnalité se trouve sur :

<http://clic.mandrakesoft.com/documentation/ch10.html>

Malheureusement, cela ne convient pas pour nos PC diskless du LAOG pour plusieurs raisons :

1. URPMI-serveur lance en fait URPMI sur les clients diskless. Tous les PC diskless qui doivent être mis-à-jour doivent donc être allumés au moment de la MAJ, bien que tout se trouve en réalité sur les serveurs.

2. Il ne faut pas installer 36 fois (s'il y a 36 PC diskless) le même fichier dans /usr si /usr partagé par tous les diskless.
3. Le serveur va demander en même temps à tous les diskless d'accéder en écriture sur un fichier sur son partage NFS. Tous les clients vont donc vouloir accéder en même temps au même fichier sur le serveur.

Cette solution n'est donc pas réaliste.

25.2.3 Une autre piste : installation dans un chroot

Autre solution : installer un paquetage sur tous les PC diskless directement à partir du serveur en lançant la commande urpmi dans un chroot bien choisi (voir man chroot).

Trop compliqué à mettre en oeuvre.

25.2.4 Une autre piste avec quelques scripts

C'est cette solution qui est retenue. Elle est décrite dans la section *Procédure d'installation de paquetages*.

26 Procédure d'installation de paquetages

Si on se contente de faire un urpmi sur un diskless, voici ce qui arrivera :

1. Les fichiers dans /usr sont bien disponibles sur tous les diskless car /usr est commun à tous les PC diskless via un montage NFS.
2. Les fichiers dans /etc ne sont accessibles que sur le PC diskless d'installation.

Les paquetages seront donc installés à moitié...

Je propose une solution en plusieurs étapes :

1. Installer les paquetages sur un diskless avec urpmi.
2. Regarder tous les fichiers installés et les copier si nécessaire (s'il y a des fichiers dans /etc par ex.) sur chacun des diskless.

Cela se fait avec deux scripts :

1. *dkl-paquetage-install-diskless.sh*, à exécuter sur un diskless.
2. *dkl-paquetage-install.sh*, à exécuter sur le serveur.

Quelques commandes utiles :

1. *rpm -ql nom-paquetage* : lister les fichiers d'un paquetage installé.
2. *rpm -qa* : lister tous les paquetages installés.
3. *rpm -qa | grep nom-paquetage* : regarder si un paquetage est installé.
4. *rpm -qa | xargs rpm -ql* : lister tous les fichiers installés par un paquetage.

26.0.1 Préparation

Avant de commencer :

1. *dkl-paquetage-install-diskless.sh* doit être accessible sur le diskless. Placez le par exemple dans `/diskless/shares/commun`.
2. *dkl-paquetage-install.sh* doit être accessible sur le serveur. Placez le par exemple dans `/diskless/scripts`
3. Les sources de paquetages doivent être correctement installées sur les diskless.
4. Les bases de données RPM et URPMI doivent être correctement partagées entre les diskless.

26.0.2 Partage des bases de données RPM et URPMI

Voici les répertoires concernés :

1. `/etc/urpmi` : contient la liste des sources.
2. `/var/lib/urpmi` : contient la liste des paquetages de chaque source.
3. `/var/lib/rpm` : contient la base de données RPM, c.à.d. la liste des paquetages installés.

Ces trois répertoires ne sont pas nécessaires au démarrage du PC diskless. Ils seront donc partagés par des liens symboliques. Voici les liens symboliques qui doivent être placés sur chaque diskless :

1. `/etc/urpmi -> /commun/etc/urpmi`
2. `/var/lib/urpmi -> /commun/var/lib/urpmi`
3. `/var/lib/rpm -> /commun/var/lib/rpm`

Ces 3 liens symboliques vont être mis en place sur le diskless modèle. Comme ce sont des liens symboliques et non des liens durs, ils seront correctement dupliqués sur les autres diskless lors de la commande "cp -a".

26.0.3 Première étape : sur un diskless

Démarrer un PC diskless. Taper "urpmi nom-du-logiciel". Noter les noms des paquetages qu'il installe sur le PC diskless. Par exemple :

```
-bash-2.05b# urpmi vim-X fluxb
(...)
installation de /var/cache/urpmi/rpms/vim-X11-6.1-36mdk.i586.rpm /var/cache/urpmi/rpms/f
```

```
Préparation... #####
  1:vim-X11      #####
  2:fluxbox     #####
-bash-2.05b#
```

Ici, les noms de paquetages sont "vim-X11" et "fluxbox" (indiqué en-dessous de la ligne "Préparation...").

Toujours sur le même diskless, une fois les logiciels installés, tapez :

```
/commun/dkl-paquetage-install-diskless.sh vim-X11 fluxbox
```

Cela va créer le fichier /tmp/MAJ-paquetages.txt qui contient la liste de tous les fichiers de vim-X11 et de fluxbox. Parmi ces fichiers, certains sont déjà partagés entre les diskless (fichiers dans /usr) et d'autres non (fichiers dans /etc).

26.0.4 Seconde étape : sur le serveur

Aller sur le serveur de diskless et taper la commande suivante :

```
/diskless/scripts/dkl-paquetage-install.sh 195.220.79.220 /tftpboot/195.220.79.220/tmp/M
```

Les arguments de dkl-paquetage-install.sh sont :

1. L'adresse IP du diskless sous lequel on a installé les paquetages.
2. Le fichier contenant la liste des fichiers qu'il faut éventuellement copier sur tous les autres diskless

Voici ce que fait ce dernier script :

1. Il lit le fichier MAJ-paquetages.txt généré sur le PC diskless.
2. Pour chaque fichier installé dans /etc/* et /lib/*, il copie ce fichier sur tous les autres diskless.
3. Si ce fichier est déjà présent, il demande à l'administrateur de confirmer la copie car l'option -i est passée à cp. Ainsi, aucun fichier ne sera écrasé par erreur.

26.0.5 Résolution de problèmes RPM

Symptômes : RPM ou URPMI ne répondent plus.

Suivre les règles suivantes :

1. Si rpm ou urpmi ne répondent plus pendant longtemps, vérifier avec "top" qu'il ne fait vraiment rien. S'il utilise des cycles cpu, alors il faut juste attendre.
2. Si rpm ou urpmi ne répond vraiment pas, arrêtez-le avec "Control+C" ou "kill -9". Mais dans ce cas, il faut absolument exécuter les commandes décrites dans l'étape 3 pour réparer les dégats.
3. Pour réparer la base de données RPM : Exécuter : `rm -f /var/lib/rpm/__.db*`
Ces fichiers seront régénérés automatiquement. Puis, si nécessaire, exécuter : `rpm --rebuilddb` Cette commande prend du temps (quelques minutes). Il faut attendre.

Septième partie

Mises au point

27 XFree

XFree a besoin de son fichier de configuration : */etc/X11/XF86Config*.

Ce fichier contient le nom du driver de la carte vidéo (par exemple "ati" ou "nvidia") et d'autres choses.

Il doit donc être modifié sur chaque PC diskless en fonction de son matériel.

D'ordinaire, ce fichier de configuration est écrit automatiquement à l'installation de Linux Mandrake. Mais les PC diskless ne sont pas installés avec le programme d'installation de Mandrake mais en copiant le système du serveur.

Le fichier XF86Config contient donc les bonnes informations pour le serveur mais doit être adapté pour les PC diskless car leur matériel n'est pas le même.

Deux fichiers de configuration XF86Config-4 se trouvent en annexe.

28 Problèmes à l'arrêt du diskless

28.1 Où se situe le problème

Les scripts d'arrêt de Mandrake Linux démonte les systèmes de fichiers NFS avant la fin de l'arrêt du PC diskless. Ensuite, l'arrêt ne peut pas se terminer car le script d'arrêt ne trouve plus les commandes qui se trouvent sur les systèmes de fichiers démontés.

Le script concerné est */etc/init.d/netfs*, ligne 97. Il est appelé avec l'argument "stop" car il y a un lien vers ce script ici :

```
# ls -l /etc/rc.d/rc?.d/???netfs
/etc/rc.d/rc0.d/K75netfs
/etc/rc.d/rc1.d/K75netfs
/etc/rc.d/rc2.d/K75netfs
/etc/rc.d/rc3.d/S25netfs
/etc/rc.d/rc4.d/S25netfs
/etc/rc.d/rc5.d/S25netfs
/etc/rc.d/rc6.d/K75netfs
#
```

À cause de ce script, tous les montages NFS sont arrêtés, y compris la racine. Immédiatement après ce script, l'arrêt se poursuit et lance des programmes, mais ces programmes ne peuvent plus être trouvés.

La commande de démontage est la suivante dans le script (voir ligne 97) :

```
umount -f -l -a -t nfs
```

Les arguments la commande de démontage sont :

1. "-a" démonter les entrées de `/etc/fstab`
2. "-f" forcer le démontage
3. "-l" démontage paresseux (voir paragraphe suivant)
4. "-t nfs" ne démonter que les montages NFS

28.2 Qu'est-ce qu'un démontage paresseux ?

Un démontage paresseux ne démonte pas immédiatement le système de fichiers. Après un démontage paresseux, il n'est plus possible de lire ou de modifier les fichiers du système de fichiers tout comme un vrai démontage.

Cependant, les fichiers ouverts restent ouverts et il est possible de continuer à les lire ou les modifier.

Lorsque tous les programmes auront fermés leurs fichiers du système de fichier concerné, le démontage sera effectif.

28.3 Ce qui se passe exactement

L'option "-t nfs" sélectionne aussi la racine car elle est montée par NFS. Cependant, comme le démontage est paresseux, cela ne prend pas effet immédiatement. `umount` cherche donc à démonter la racine et pour ce faire, il faut démonter d'abord les sous-répertoires de la racine. Ainsi, `/proc` est démonté immédiatement car il n'est pas utilisé à cette étape du script.

Mais dans le script, la ligne qui suit le démontage est (ligne 102) :

```
remaining='LC_ALL=C awk '!/^#/ && $3 ~ /^nfs/ && $2 != "/" {print $2}' /proc/mounts'
```

Donc le script essaie de lire le fichier `/proc/mounts` ; ce fichier n'est plus accessible car `/proc` est effectivement démonté.

Cependant, le système marche toujours (enfin, un peu...) car la racine n'est pas encore démontée car des fichiers sont encore ouverts et le démontage est paresseux.

28.4 Première idée

Une idée serait de commenter l'entrée "root" (`/`) dans `/etc/fstab`. Ainsi, l'option "-a" de `umount` ne sélectionnerait pas la racine et `/` ne se démonterait pas à cette étape du script.

De toute façon, la racine n'est pas montée avec `fstab` mais avec les options `root=/dev/nfs` et `nfsroot=/diskless/tftpboot/%s` passées au noyau.

J'ai fait le test. Le PC `diskless` démarre correctement. `XFree` et `XDM` se lancent correctement mais `KDE` ne démarre plus (il se fige). Je ne sais pas pourquoi.

De toute façon, ce n'est sans doute pas une bonne idée car d'autres programmes pourraient avoir besoin de `/etc/fstab` pour savoir d'où est montée la racine.

28.5 La solution

La solution consiste à modifier le script d'arrêt pour ne pas démonter la racine. Le fichier à modifier est `/etc/init.d/netfs`, ligne 97.

J'ai donc remplacé ceci :

```
if [ "$retry" -lt 3 ]; then
action "Unmounting NFS filesystems (retry): " umount -f -l -a -t nfs
else
action "Unmounting NFS filesystems: " umount -f -l -a -t nfs
fi
```

par ceci :

```
if [ "$retry" -lt 3 ]; then
  for repert in $remaining
  do
    action "Unmounting NFS filesystems $repert (retry): " umount -f -l $repert
  done
else
  for repert in $remaining
  do
    action "Unmounting NFS filesystems $repert: " umount -f -l $repert
  done
fi
```

La variable `$remaining` contient la liste des montages à défaire. Cette variable est déjà remplie correctement par le script (c'est-à-dire elle ne contient pas `/`, la racine), il n'y a donc rien d'autre à faire.

Cela résout le problème : ces montages sont bien démontés et `/proc` est toujours accessible après ces démontages. Le script peut donc continuer.

De plus, cette méthode est assez portable : si on décide de modifier les points de montage NFS, cela marchera toujours.

L'autre solution non portable serait d'énumérer les montages NFS à démonter dans le script. Il faudrait alors modifier le script d'arrêt chaque fois que l'on décide d'ajouter un montage.

28.6 D'autres problèmes

Cependant, le script se bloque plus loin lors de l'arrêt de l'interface réseau `eth0`. En effet, cette interface réseau pour accéder au serveur NFS et est donc nécessaire pour le montage NFS de la racine.

Le problème se trouve dans le fichier `/etc/init.d/network`, ligne 210 environ. Ce fichier est appelé avec l'argument "stop" car des liens se trouvent ici :

```
# ls -l /etc/rc.d/rc?.d/???network
/etc/rc.d/rc0.d/K90network
/etc/rc.d/rc1.d/K90network
/etc/rc.d/rc2.d/S10network
/etc/rc.d/rc3.d/S10network
/etc/rc.d/rc4.d/S10network
/etc/rc.d/rc5.d/S10network
/etc/rc.d/rc6.d/K90network
#
```

J'ai donc mis en commentaire les lignes suivantes (ligne 215) :

```
if ! check_device_down $i; then
    action "Shutting down interface %s: " $i ./ifdown $i boot
fi
```

Depuis, le diskless s'arrête correctement.

28.7 Conclusion

Pour résumer, il faut :

1. Ne pas démonter la racine avec "umount -f -l -a -t nfs" mais plutôt démonter tous les montages NFS sauf la racine qui sera démonté à part.
2. Ne pas débrancher le réseau en cours d'utilisation.

De plus, comme le PC diskless a été arrêté correctement, il ne se plaint pas au démarrage et ne fait pas de fsck (file system check).

29 Problème des pilotes NVidia

Les pilotes NVidia sont un module nommé NVdriver du noyau linux. J'avais copié le fichier NVdriver.o ici :

```
/diskless/tftpboot/195.220.79.220/lib/modules/2.4.21-0.13mdk-diskless-pxe1/kernel/driver
```

Après le démarrage de la machine diskless, ce fichier se trouve au bon endroit, c'est à dire ici :

```
/lib/modules/2.4.21-0.13mdk-diskless-pxe1/kernel/drivers/char/
```

Mais ce module n'était pas chargé, bien que le fichier `/etc/modules` et `/etc/modules.conf` étaient corrects. Donc XFree ne démarrait pas. Pourtant, on pouvait tout à fait taper `"insmod NVdriver"` avec succès. Par contre, `"modprobe NVdriver"` ne marchait pas. Et c'est `modprobe` qui est utilisé dans les scripts de démarrage.

En fait l'explication est la suivante : le répertoire `/lib/modules/.../` de la machine `diskless` est monté à partir d'un export NFS du serveur. Pendant l'exécution des scripts de démarrage, `/lib/` est en fait :

```
/diskless/lib/
```

Ensuite, la partition `/lib/` est montée à partir de :

```
/diskless/tftpboot/195.220.79.220/lib/
```

Les scripts de démarrage fonctionnent ainsi :

1. La partition `/` est montée par NFS
2. `depmod` est lancé pour trouver tous les modules et leurs dépendances.
3. `/etc/modules` est lu pour charger les bons modules
4. La partition `/lib/` et autres sont montées par NFS. Elles cachent les fichiers qui étaient à l'emplacement du montage.

Il faut donc mettre le module `NVdriver.o` dans le répertoire `/diskless/lib/...` pour que le module soit chargé automatiquement. Mais il faut aussi le mettre dans `/diskless/tftpboot/195.220.79.220/lib/...` pour pouvoir charger le module manuellement avec `modprobe/insmod` par la suite.

Au départ, je n'avais mis le module que dans un répertoire, ce qui explique que le module ne pouvait pas se lancer automatiquement et qu'on pouvait le lancer manuellement avec `insmod`. Par contre, on ne pouvait pas lancer le module avec `modprobe` car `modprobe` utilise les calculs de dépendances effectués par `depmod` dans les scripts de démarrage. Et pour `depmod`, il n'y avait pas de module `NVdriver`.

Maintenant, on le module se charge automatiquement et on peut le décharger avec `rmmmod` puis le recharger avec `insmod` ou `modprobe`.

30 Conflits entre serveurs DHCP

Un serveur DHCP permet d'attribuer automatiquement des adresses IP aux clients qui font une requête DHCP. Le serveur peut être configuré pour donner une adresse libre quelconque dans une plage d'adresse ou pour donner une IP fixe aux clients dont l'adresse MAC est connue.

Dans notre cas, chaque PC `diskless` se voit attribuer une adresse IP fixe ; le serveur DHCP connaît donc les adresses MAC de toutes les cartes réseaux des PC `diskless`.

Pour démarrer, le PC diskless a besoin d'un serveur DHCP pour deux raisons :

1. Si le PC diskless boote en utilisant le protocole PXE : quand le PC boote sur le réseau en utilisant le protocole PXE, il émet une requête DHCP pour trouver ses paramètres réseaux. Ces paramètres réseaux vont servir pour demander à un serveur TFTP le noyau du système d'exploitation. Si le PC diskless n'utilise pas PXE, seule la deuxième raison est correcte.
2. Une fois le système d'exploitation lancé, celui-ci fait une requête DHCP pendant les scripts de démarrage pour trouver ses paramètres réseaux. Le réseau sera utilisé notamment pour les montages NFS. Cette seconde requête est nécessaire car le système d'exploitation ne peut pas savoir qu'une requête DHCP a déjà été faite.

Il faut donc un serveur DHCP. Cependant, sur le même réseau physique que mes machines de tests, un serveur DHCP légitime est déjà présent. Il est tout à fait possible d'utiliser le serveur DHCP déjà présent mais je n'ai pas accès à ce serveur. De plus avoir un serveur DHCP pour chaque type d'utilisation est plus facile à administrer.

J'ai donc configuré le serveur DHCP qui va servir au PC diskless. Le problème est de ne pas interférer avec le serveur DHCP légitime déjà en place. La solution est l'option suivante dans le fichier `/etc/dhcp.conf` :

```
deny unknown-clients;
```

Cette option indique au serveur DHCP d'ignorer les requêtes des PC diskless dont l'adresse MAC n'est pas connue.

Voici le fichier de configuration utilisé :

```
# allow booting
# allow bootp
ddns-update-style none;
subnet 195.220.79.0 netmask 255.255.255.0 {
    # default gateway
    option routers 195.220.79.250;
    option subnet-mask 255.255.255.0;

    option domain-name "obs.ujf-grenoble.fr";
    option domain-name-servers 195.220.79.11;

    deny unknown-clients;
}
# Adresses dynamiques
#       range dynamic-bootp 195.220.79.124 195.220.79.125;
```

```

# Adresses statiques affectees aux PC diskless bootant
# a travers PXE
#une directive "host" pour chaque diskless
# l'option next-server donne l'adresse IP du serveur de boot
# filename donne le nom de fichier de boot (attention, pas
# le noyau linux)
host gag220 {
    allow booting;
    hardware ethernet 00:07:95:C6:9F:A8;
    fixed-address 195.220.79.220;
    # Pour la methode de boot avec CDRom, on s'arrete la.
# Pour les autres methodes, il faut rajouter l'IP du serveur
    # next-server 195.220.79.217;
# et le fichier a telecharger:
    # filename "/pxelinux.0";
    # Le chemin est 'relatif' au chemin donne à xinetd/tftpd
}
}

```

Avec cette configuration, les autres clients du réseau ne sont pas dérangés par le nouveau serveur DHCP. car ce dernier ignore leurs requêtes.

Le second problème est que l'ancien serveur DHCP répond aux requêtes du nouveau client. Le nouveau client recoit donc des réponses des deux serveurs DHCP. C'est le plus rapide qui gagne. Heureusement, le bon serveur répond souvent plus rapidement, sans doute parce qu'il est plus près en nombre de switch à traverser.

Il faut modifier la configuration de l'ancien serveur DHCP pour qu'il ignore les requêtes qui ne le concerne pas. Pour cela, on utilise aussi l'option "deny unknown-clients;".

31 NFS lockd

Au démarrage du PC diskless, les scripts de démarrage peuvent afficher :

```
NFS lockd: chdir(/var/lib/nfs) failed: No such file or directory
```

Voici ce que contient /var/lib/nfs :

/var/lib/nfs/etab contains information about what filesystems should be exported to whom at the moment.

/var/lib/nfs/rmtab contains a list of which filesystems actually are mounted by certain clients at the moment.

/var/lib/nfs/xtab is the same information as /proc/fs/nfs/exports but is maintained by nfs-utils instead of directly by the kernel. It is only used if /proc isn't mounted.

/tftpboot/195.220.79.xxx/var/lib est un lien symbolique vers /commun/-lib. Or /commun n'est pas encore monté lorsque NFS lockd est lancé.

Actuellement, le /var/lib/nfs de chaque PC diskless sont communs entre les PC diskless car il se trouve dans /diskless/tftpboot/commun sur le serveur.

Mais il faut que le /var/lib/nfs de chaque PC diskless ne soit pas partagé entre les PC diskless pour deux raisons :

1. NFS lockd doit être démarré avant le montage des répertoires NFS des PC diskless. Or, les fichiers communs ne sont accessibles que par un partage NFS.
2. Le répertoire /var/lib/nfs contient des informations qui par nature ne doivent pas être partagées. En effet, celles-ci renseignent sur l'activité NFS du PC diskless qui ne sera pas la même d'un PC diskless à l'autre.

Mettre un répertoire /var/lib/nfs non partagé entre les PC diskless résoud le problème.

32 Message "chgrp : invalid group name : utmp"

Le message suivant peut apparaître au démarrage de la machine :

```
chgrp: invalid group name: utmp
```

Le problème est maintenant résolu. En suivant la méthode décrite dans la partie Installation du serveur de diskless, vous ne devriez pas avoir ce problème.

Ce problème se produisait lorsque les fichiers d'authentification (/etc/group, /etc/shadow et /etc/passwd) n'étaient pas partagés correctement.

Le problème vient du fichier /etc/rc.d/rc.sysinit, ligne 930. La ligne est :

```
chgrp utmp /var/run/utmp /var/log/wtmp
```

Pour connaître le GID de utmp, le script a besoin de /etc/group. Le fichier /etc/group est commun à tous les diskless. Pour le rendre commun, il se trouve dans un endroit unique : /diskless/tftpboot/commun/.

Des liens symboliques /'étaient faits dans tous les répertoires /etc des diskless vers les fichiers correspondants situés sous /diskless/tftpboot/commun.

Or, /commun n'est pas encore monté par NFS à cette étape du script, sur le diskless. Celui-ci ne peut donc pas résoudre le lien symbolique.

Solution : au lieu de faire un lien symbolique que le diskless est incapable de résoudre, j'ai fait un lien dur. Ainsi, ce sera le serveur qui résoudra le lien.

Document sur les liens et NFS :

http://ou800doc.caldera.com/SDK_sysprog/_Using_Symbolic_Links_with_NFS.html

Mettre des liens durs est aussi utile pour les autres fichiers de /commun pour les mêmes raisons.

33 Impossible de trouver les modules dans `/lib/modules/2.4.21-01`

Les modules du noyau se trouvent normalement dans :

```
/lib/modules/'uname -r'
```

Où `uname -r` représente le numéro de version du noyau.

Ici, les modules du noyau sont cherchés dans le mauvais répertoire : il les cherche dans `/lib/modules/2.4.21-01` au lieu de les chercher dans `/lib/modules/2.4.21-0.13mdk-diskless-pxe1`.

Ce cas se produit lorsque les modules du diskless sont copiés sans utiliser l'option `-a` de `cp`.

L'option `-a` permet de garder la structure et les attributs du fichier original lors de la copie.

Le problème est résolu en recopiant les modules en utilisant l'option `-a` de `cp`.

Huitième partie

Utilisation du disque dur du diskless

34 Introduction

But :

1. Swapper
2. Stockage de fichiers utilisateurs : accès plus rapide
3. Possibilité de d'installer un système complet ultérieurement
4. Démarrer sur le disque dur avec Grub en tant que diskless pour se passer du CDRom de démarrage.

Le script `dkl-fdisk.sh` et son fichier de configuration `partitions.fdisk` sont faits pour cela.

Ces deux fichiers doivent être placés dans `/commun/fdisk`. En exécutant ce script en root à partir du diskless, il va faire les choses suivantes :

1. Effacer toutes les partitions de tous les disques durs.
2. Recréer des partitions et les formater.
3. Indiquer ce qu'il faut ajouter dans `/etc/fstab` pour que le PC diskless puisse monter les nouvelles partitions.

Les partitions créées sont les suivantes :

1. 10 Go - Partition prête pour une utilisation ultérieure.
2. 512 Mo - Swap pour le PC diskless.
3. Reste - Montée sur `/local` pour le PC diskless.

35 Installation sur le serveur

Créer le répertoire /commun/fdisk contenant :

```
/commun/fdisk
| - partitions.fdisk: fichier de configuration de dkl-fdisk.sh
| - dkl-fdisk.sh: script d'installation sur le disque dur du diskless
| - grub
  | - vmlinuz-diskless: noyau du diskless
  | - stage1: fichier binaire de Grub
  | - stage2: fichier binaire de Grub
  | - menu.lst: fichier de configuration de Grub
```

Voici un exemple du fichier menu.lst :

```
timeout 30
default 0

title Demarrer en tant que diskless
kernel (hd0,0)/grub/vmlinuz-diskless root=/dev/nfs devfs=mount ip=:::::dhcp nfsroot=/di

title Demarrer a partir du disque dur
kernel (hd0,0)/boot/vmlinuz root=/dev/hde1 devfs=mount vga=788
initrd (hd0,0)/boot/initrd.img

title Disquette
root (fd0)
chainloader +1
```

36 Installation sur le diskless

Lancer /commun/fdisk/dkl-fdisk.sh.

Neuvième partie

Annexes

A Glossaire

Adresse IP : Série de 4 nombres identifiant un ordinateur sur un réseau. S'écrit sous la forme : "195.220.97.220".

Bios : Basic Input-Output System ?

Boot : Démarrer. Un ordinateur peut démarrer un système d'exploitation accessible sur le réseau, une disquette, un disque,

DHCP : Un serveur DHCP permet d'attribuer des adresses IP aux clients qui en font la requête.

Diskless : Un PC sans disque ou "PC diskless" est un poste de travail dont le disque s'il en a un, n'est pas dédié au système Linux. Celui-ci sera chargé au "boot" du PC, via le réseau à partir d'un serveur distant.

GNU/Linux : Désigne le système d'exploitation composé du noyau linux et des programmes GNU.

Grille de calcul : Ensemble d'ordinateurs travaillant en collaboration pour faire des calculs intensifs distribués. <http://www.urec.cnrs.fr/publications/JRES01.Datagrid.pdf>

GRUB : Chargeur de démarrage, tout comme LILO. Mais GRUB est plus pratique et possède plus de fonctionnalités.

Inode : Tous les fichiers d'une même partition possèdent un numéro unique appelé indode.

LILO : LInux LOader. Petit logiciel se plaçant au début du disque et permettant de charger Linux au démarrage de l'ordinateur.

Lien dur : Aussi appelé lien matériel. Fonctionnalités sous UNIX permettant qu'un fichier soit présent dans plusieurs répertoire à la fois.

Linux : Nom du noyau de système d'exploitation utilisé au LAOG.

Monter : Monter un disquette, un CDROM.

PXE : Pre-boot Execution Environment. Protocole implémenté dans certaines cartes réseau permettant de démarrer l'ordinateur non pas sur le disque ou une disquette mais sur le réseau en utilisant un serveur DHCP et TFTP.

PXE : Pre-boot Execution Environment implémenté par le logiciel Grub.

NFS : Network File System. Système de fichiers en réseau.

Serveur de diskless : Serveur qui va permettre aux PC diskless de démarrer.

Terminal X : Console permettant d'utiliser des applications graphiques sur un serveur. Les TX n'ont souvent aucun disque.

TFTP : Protocole de transfert de fichiers, similaire à FTP mais simplifié.

XFree : Logiciel permettant d'afficher des applications graphiques utilisé sur les systèmes UNIX.

B Arborescence du serveur de diskless

```
/diskless
| - CDROM
|   | - isos : les images des CD seront placees ici
|   | - root : correspond a la racine du CD
|       | - isolinux : contient le binaire d'isolinux et son fichier de config.
|       | - alt0 : contient le noyau vmlinuz
| - scripts
| - shares
|   | - commun
|   | - bin, sbin, home, lib, usr, opt
| - tftpboot
|   | - pxelinux.cfg
|   | - 195.220.79.220
|   | - 195.220.97.242
```

C Machines de tests

J'ai eu deux machines de tests. Voici leurs adresses IP et MAC.

	<i>Serveur</i>	<i>Diskless</i>
<i>Nom</i>	<i>gag217</i>	<i>gag220</i>
<i>MAC</i>	00 : 07 : 95 : 4E : B9 : D9	00 : 07 : 95 : C6 : 9F : A8
<i>IP</i>	195.220.79.217(<i>IP fixe</i>)	195.220.79.220(<i>DHCP</i>)
<i>IPenhexa</i>	<i>C3DC4FDC</i>	<i>C3DC4FDF</i>

À partir de la troisième semaine de mon stage, j'ai eu un autre PC de test. Celui-ci a servi de deuxième PC diskless :

	<i>SecondDiskless</i>
<i>Nom</i>	<i>gag242</i>
<i>MAC</i>	00 : d0 : b7 : c4 : dc : ba
<i>IP</i>	195.220.79.242(<i>DHCP</i>)

gag217 (le serveur de diskless) et gag220 (le premier PC diskless) ont exactement le même matériel. Cela ne posait donc pas trop de problèmes. Mais gag242 (le deuxième PC diskless) n'a pas le même matériel.

Pour récupérer l'adresse MAC d'un PC diskless, on peut procéder de la manière suivante :

1. S'assurer que le serveur DHCP tourne et surveiller `/etc/log/messages` (avec la commande `tail -f /etc/log/messages`).
2. Démarrer le PC diskless sur le CD pour qu'il fasse une requête DHCP.
3. L'adresse IP apparaît dans `/etc/log/messages`, par exemple : "DHCP-DISCOVER from 00 :d0 :09 :71 :a0 :f5".

D Fichiers XF86Config-4

Voici deux fichiers XF86Config-4.

D.1 Fichier du diskless gag220

Section "Files"

```
# Multiple FontPath entries are allowed (they are concatenated together)
# By default, Mandrake 6.0 and later now use a font server independent of
# the X server to render fonts.
RgbPath "/usr/X11R6/lib/X11/rgb"
#FontPath "tcp/195.220.79.217:7100"
FontPath "/usr/X11R6/lib/X11/fonts/misc:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/drakfont"
FontPath "/usr/X11R6/lib/X11/fonts/drakfont/Type1"
FontPath "/usr/X11R6/lib/X11/fonts/drakfont/ttf"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
```

```

FontPath "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/Type1"
FontPath "/usr/X11R6/lib/X11/fonts/TTF"
FontPath "/usr/X11R6/lib/X11/fonts/Speedo"
FontPath "/usr/X11R6/lib/X11/fonts/pcf\_drakfont:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/mdk:unscaled"
FontPath "/usr/share/fonts/default/Type1"
FontPath "/usr/share/fonts/ttf/western"
FontPath "/usr/share/fonts/ttf/decoratives"
EndSection

Section "ServerFlags"
    AllowMouseOpenFail # allows the server to start up even if the mouse doesn't work
EndSection

Section "Module"
    Load "dbe" # Double-Buffering Extension
    Load "v4l" # Video for Linux
    Load "extmod"
    Load "type1"
    Load "freetype"
    Load "/usr/X11R6/lib/modules/extensions/libglx.so"
EndSection

Section "InputDevice"
    Identifier "Keyboard1"
    Driver "Keyboard"
    Option "XkbModel" "pc105"
    Option "XkbLayout" "fr"
    Option "XkbCompat" ""
    Option "XkbOptions" ""
EndSection

Section "InputDevice"
    Identifier "Mouse1"
    Driver "mouse"
    Option "Protocol" "PS/2"
    Option "Device" "/dev/psaux"
    Option "Emulate3Buttons"
    Option "Emulate3Timeout" "50"
EndSection

Section "Monitor"
    Identifier "monitor1"

```

```

VendorName "Plug'n Play"
HorizSync 30-96
VertRefresh 50-160

# Sony Vaio C1(X,XS,VE,VN)?
# 1024x480 @ 85.6 Hz, 48 kHz hsync
ModeLine "1024x480"      65.00 1024 1032 1176 1344   480  488  494  563 -hsync -vsync

# TV fullscreen mode or DVD fullscreen output.
# 768x576 @ 79 Hz, 50 kHz hsync
ModeLine "768x576"      50.00  768  832  846 1000   576  590  595  630

# 768x576 @ 100 Hz, 61.6 kHz hsync
ModeLine "768x576"      63.07  768  800  960 1024   576  578  590  616
EndSection

Section "Device"
    Identifier "device1"
    VendorName "nVidia Corporation"
    BoardName "RIVA TNT2"
    Driver "nvidia"
    Option "DPMS"
EndSection

Section "Screen"
    Identifier "screen1"
    Device "device1"
    Monitor "monitor1"
    DefaultColorDepth 24

    Subsection "Display"
        Depth 8
        Modes "1024x768" "800x600" "640x480"
    EndSubsection

    Subsection "Display"
        Depth 15
        Modes "1024x768" "800x600" "640x480"
    EndSubsection

    Subsection "Display"
        Depth 16
        Modes "1024x768" "800x600" "640x480"
    EndSubsection

```

```
    Subsection "Display"
        Depth 24
        Modes "1024x768" "800x600" "640x480"
    EndSubsection
EndSection
```

```
Section "ServerLayout"
    Identifier "layout1"
    InputDevice "Keyboard1" "CoreKeyboard"
    InputDevice "Mouse1" "CorePointer"
    Screen "screen1"
EndSection
```

D.2 Fichier du diskless gag220

```
Section "Files"
    # Multiple FontPath entries are allowed (they are concatenated together)
    # By default, Mandrake 6.0 and later now use a font server independent of
    # the X server to render fonts.
    RgbPath "/usr/X11R6/lib/X11/rgb"
    #FontPath "tcp/195.220.79.217:7100"
    FontPath "/usr/X11R6/lib/X11/fonts/misc:unscaled"
    FontPath "/usr/X11R6/lib/X11/fonts/drakfont"
    FontPath "/usr/X11R6/lib/X11/fonts/drakfont/Type1"
    FontPath "/usr/X11R6/lib/X11/fonts/drakfont/ttf"
    FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
    FontPath "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
    FontPath "/usr/X11R6/lib/X11/fonts/Type1"
    FontPath "/usr/X11R6/lib/X11/fonts/TTF"
    FontPath "/usr/X11R6/lib/X11/fonts/Speedo"
    FontPath "/usr/X11R6/lib/X11/fonts/pcf_drakfont:unscaled"
    FontPath "/usr/X11R6/lib/X11/fonts/mdk:unscaled"
    FontPath "/usr/share/fonts/default/Type1"
    FontPath "/usr/share/fonts/ttf/western"
    FontPath "/usr/share/fonts/ttf/decoratives"
EndSection
```

```
Section "ServerFlags"
    AllowMouseOpenFail # allows the server to start up even if the mouse doesn't work
EndSection
```

```
Section "Module"
    Load "dbe" # Double-Buffering Extension
```

```

Load "v4l" # Video for Linux
Load "extmod"
Load "type1"
Load "freetype"
Load "/usr/X11R6/lib/modules/extensions/libglx.so"
EndSection

```

```

Section "InputDevice"
    Identifier "Keyboard1"
    Driver "Keyboard"
    Option "XkbModel" "pc105"
    Option "XkbLayout" "us_intl"
    Option "XkbCompat" ""
    Option "XkbOptions" ""
EndSection

```

```

Section "InputDevice"
    Identifier "Mouse1"
    Driver "mouse"
    Option "Protocol" "PS/2"
    Option "Device" "/dev/psaux"
    Option "Emulate3Buttons"
    Option "Emulate3Timeout" "50"
EndSection

```

```

Section "Monitor"
    Identifier "monitor1"
    VendorName "Plug'n Play"
    HorizSync 30-96
    VertRefresh 50-160

    # Sony Vaio C1(X,XS,VE,VN)?
    # 1024x480 @ 85.6 Hz, 48 kHz hsync
    ModeLine "1024x480"      65.00 1024 1032 1176 1344   480  488  494  563 -hsync -vsync

    # TV fullscreen mode or DVD fullscreen output.
    # 768x576 @ 79 Hz, 50 kHz hsync
    ModeLine "768x576"      50.00  768  832  846 1000   576  590  595  630

    # 768x576 @ 100 Hz, 61.6 kHz hsync
    ModeLine "768x576"      63.07  768  800  960 1024   576  578  590  616
EndSection

```

```

Section "Device"

```

```

Identifier "device1"
VendorName "nVidia Corporation"
BoardName "RIVA TNT2"
Driver "ati"
#Driver "nvidia"
Option "DPMS"
EndSection

Section "Screen"
Identifier "screen1"
Device "device1"
Monitor "monitor1"
DefaultColorDepth 24

Subsection "Display"
Depth 8
Modes "1024x768" "800x600" "640x480"
EndSubsection

Subsection "Display"
Depth 15
Modes "1024x768" "800x600" "640x480"
EndSubsection

Subsection "Display"
Depth 16
Modes "1024x768" "800x600" "640x480"
EndSubsection

Subsection "Display"
Depth 24
Modes "1024x768" "800x600" "640x480"
EndSubsection
EndSection

Section "ServerLayout"
Identifier "layout1"
InputDevice "Keyboard1" "CoreKeyboard"
InputDevice "Mouse1" "CorePointer"
Screen "screen1"
EndSection

```

E Ce qu'il reste à faire

E.1 Vitesse de KDE

Accelerer KDE avec startkde...
PC diskfull : 16 secondes
PC diskless : 28 secondes

F Scripts

Les scripts sont peuvent être téléchargés ici :
<http://alban.apinc.org>
Voici le listing des scripts et des exemples de fichiers de configuration.

F.1 dkl-init-serveur.sh

```
#!/bin/sh -x
#
# Ce script prepare un serveur de diskless

# Avant de l'executer, il faut:
# 1/ Creer /diskless
# 2/ Creer /diskless/scripts
# 3/ Copier tous les fichiers de diskless-kit.tar dans /diskless/scripts

# S'assurer que /diskless existe
mkdir -p /diskless

# S'assurer que /diskless/scripts existe
mkdir -p /diskless/scripts

# Preparer /diskless/tftpboot
mkdir -p /diskless/tftpboot
ln -s /diskless/tftpboot /tftpboot
mkdir -p /diskless/tftpboot/pxelinux.cfg
cp /diskless/scripts/pxelinux.0 /diskless/tftpboot/
tee /diskless/tftpboot/pxelinux.cfg/exemple.txt <<EOF
# Fichier exemple
DEFAULT bzImage-diskless-pxel
APPEND root=/dev/nfs devfs=mount ip=:::::dhcp nfsroot=195.220.79.217:/diskless/tftpboot
PROMPT 1
TIMEOUT 50
EOF
```

```

# Preparer /diskless/CDROM
mkdir -p /diskless/CDROM
mkdir -p /diskless/CDROM/isos/
mkdir -p /diskless/CDROM/root/
mkdir -p /diskless/CDROM/root/isolinux
mkdir -p /diskless/CDROM/root/isolinux/alt0
cp /diskless/scripts/isolinux.bin /diskless/CDROM/root/isolinux/
cp /diskless/scripts/isolinux.cfg /diskless/CDROM/root/isolinux/

# Preparer /diskless/shares
mkdir -p /diskless/shares
cp -a /bin /diskless/shares
mkdir -p /diskless/shares/commun
ln -s /diskless/shares/commun /commun
ln -s /diskless/shares/commun /diskless/tftpboot/commun
cp -a /home /diskless/shares
cp -a /lib /diskless/shares
mkdir -p /diskless/shares/opt
cp -a /root /diskless/shares
cp -a /sbin /diskless/shares
mkdir -p /diskless/shares/tmp
chmod 777 /diskless/shares/tmp
chmod +t /diskless/shares/tmp
cp -a /usr /diskless/shares

# Preparer /commun
rm -rf /commun/*
cp -af /etc/termcap /tftpboot/commun
cp -af /etc/ld.so.conf /tftpboot/commun
cp -af /etc/ld.so.cache /tftpboot/commun
cp -af /etc/passwd /tftpboot/commun
cp -af /etc/shadow /tftpboot/commun
cp -af /etc/group /tftpboot/commun
cp -af /etc/hosts /tftpboot/commun
cp -af /etc/hosts.allow.diskless /tftpboot/commun/hosts.allow
cp -af /etc/hosts.deny.diskless /tftpboot/commun/hosts.deny
cp -af /etc/hosts.equiv /tftpboot/commun
cp -af /etc/sudoers /tftpboot/commun
cp -af /etc/ftpusers /tftpboot/commun
cp -af /etc/ftphosts /tftpboot/commun
cp -af /etc/ntp.conf /tftpboot/commun
cp -af /etc/auto.master /tftpboot/commun
cp -af /etc/*.map /tftpboot/commun
cp -af /etc/printcap /tftpboot/commun

```

F.2 dkl-init-diskless.sh

```
#!/bin/sh
if [ $# != 1 ]
then
echo Usage: $0 client-IP-addr
echo Exemple: $0 195.220.79.220
exit 1
fi

cd /

umask 022

# Creer la racine du diskless
mkdir -p /tftpboot/$1

# Creer les principaux repertoires du diskless
for d in bin sbin home mnt proc tmp usr opt lib commun local
do
mkdir /tftpboot/$1/$d
done

# Changement des droits de tmp
chmod 1777 /tftpboot/$1/tmp

touch /tftpboot/$1/fastboot
#chattr +i /tftpboot/$1/fastboot

# Copier les repertoires
cp -a dev etc root var /tftpboot/$1

# Creer quelques repertoires
mkdir -p /tftpboot/$1/mnt/cdrom
mkdir -p /tftpboot/$1/mnt/cdrom2
mkdir -p /tftpboot/$1/mnt/floppy
mkdir -p /tftpboot/$1/mnt/disk
mkdir -p /tftpboot/$1/mnt/test1
mkdir -p /tftpboot/$1/mnt/test2
```

```

cat <<EOF
Maintenant, aller voir /tftpboot/$1/etc, et editer:

sysconfig/network
sysconfig/network-scripts/ifcfg-eth0
fstab
modules.conf

Puis lancer dkl-placer-liens-durs-sur-un-diskless.sh

EOF

```

F.3 dkl-diskless-menage.sh

```

#!/bin/sh
if [ $# != 1 ]
then
echo Usage: $0 client-IP-addr
echo Exemple: $0 195.220.79.220
exit 1
fi

IP=$1

echo "Ne pas lancer ce script si le diskless tourne!!"
echo "Ce script efface plein de fichiers sur le diskless."
echo "Taper [entree] pour continuer ou CONTROL-C pour quitter"
read PAUSE

[ ! -d /diskless/tftpboot/${IP}/ ] && echo "$IP n'existe pas."
[ ! -d /diskless/tftpboot/${IP}/ ] && exit 1

# fichiers temporaires
rm -rf /diskless/tftpboot/${IP}/root/*
rm -rf /diskless/tftpboot/${IP}/root/.??*
rm -rf /diskless/tftpboot/${IP}/tmp/*
rm -rf /diskless/tftpboot/${IP}/tmp/.??*
rm -rf /diskless/tftpboot/${IP}/var/tmp/*
rm -rf /diskless/tftpboot/${IP}/var/tmp/.??*

# NFS problemes
rm -f /diskless/tftpboot/${IP}/var/lib/nfs/etab

```

```

rm -f /diskless/tftpboot/${IP}/var/lib/nfs/rmtab
rm -f /diskless/tftpboot/${IP}/var/lib/nfs/xtab
touch /diskless/tftpboot/${IP}/var/lib/nfs/etab
touch /diskless/tftpboot/${IP}/var/lib/nfs/rmtab
touch /diskless/tftpboot/${IP}/var/lib/nfs/xtab
rm -f /diskless/tftpboot/${IP}/var/lib/nfs/statd/sm/*
rm -f /diskless/tftpboot/${IP}/var/lib/nfs/statd/sm.bak/*

# Un diskless ne doit pas faire serveur DHCP
[ -f /diskless/tftpboot/${IP}/etc/dhcpd.conf ] && \
mv -f /diskless/tftpboot/${IP}/etc/dhcpd.conf /diskless/tftpboot/${IP}/etc/dhcpd.conf-se

# Un diskless ne doit a priori pas exporter /diskless
[ -f /diskless/tftpboot/${IP}/etc/exports ] && \
if [ "'cat /diskless/tftpboot/${IP}/etc/exports | grep -- /diskless'" != "" ]
then
mv -f /diskless/tftpboot/${IP}/etc/exports /diskless/tftpboot/${IP}/etc/exports-serveur
fi

echo "${IP}: FAIT."

```

F.4 dkl-liens-durs.sh

```

#!/bin/bash

. /diskless/scripts/dkl-liens-durs-fonctions.sh

echo "Ce script va placer TOUS les liens durs sur TOUS les diskless."
echo "Cela prends du temps et aucun diskless ne doit tourner."
echo "Appuyez sur [entree] pour continuer ou ctrl+c pour arreter"
echo ""
read PAUSE

echo "Liste de diskless:"
echo $ListeDiskless
echo

for i in $ListeDiskless
do
copiediskless $i
done

```

F.5 dkl-liens-durs-1diskless.sh

```
#!/bin/bash

. /diskless/scripts/dkl-liens-durs-fonctions.sh

if [ $# != 1 ]
then
echo Usage: $0 client-IP-addr
echo Exemple: $0 195.220.79.220
exit 1
fi

copiediskless $1
```

F.6 dkl-1lien-dur-diskless.sh

```
#!/bin/bash

. /diskless/scripts/dkl-liens-durs-fonctions.sh

if [ $# != 2 ]
then
echo Usage: $0 Source Destination
echo Exemple: $0 /diskless/shares/commun/passwd /etc/passwd
exit 1
fi

SOURCE=$1
DEST=$2

unliendursurtous $IP $SOURCE $DEST
```

F.7 dkl-1lien-dur-1diskless.sh

```
#!/bin/bash

. /diskless/scripts/dkl-liens-durs-fonctions.sh

if [ $# != 3 ]
then
```

```
echo Usage: $0 IP Source Destination
echo Exemple: $0 195.220.79.220 /diskless/shares/commun/passwd /etc/passwd
exit 1
fi
```

```
IP=$1
SOURCE=$2
DEST=$3
```

```
unliendur $IP $SOURCE $DEST
```

F.8 dkl-suppr-1lien-dur-diskless.sh

```
#!/bin/bash

. /diskless/scripts/dkl-liens-durs-fonctions.sh

if [ $# != 1 ]
then
echo Usage: $0 Source
echo Exemple: $0 /etc/passwd
exit 1
fi

SOURCE=$1

for IP in $ListeDiskless
do
rm -Rf /diskless/tftpboot/${IP}${SOURCE}
echo "${IP}: fichier $SOURCE supprimé."
done
echo
```

F.9 dkl-suppr-1lien-dur-1diskless.sh

```
#!/bin/bash

. /diskless/scripts/dkl-liens-durs-fonctions.sh

if [ $# != 2 ]
then
```

```
echo Usage: $0 IP Source
echo Exemple: $0 195.220.79.220 /etc/passwd
exit 1
fi
```

```
IP=$1
SOURCE=$2
```

```
rm -Rf /diskless/tftpboot/${IP}${SOURCE}
echo "${IP}: fichier $SOURCE supprimé."
```

F.10 dkl-tftp-start.sh

```
#!/bin/sh
```

```
/usr/local/tftp-hpa-0.29/tftpd/tftpd -p -u tftpd -s /diskless/tftpboot
```

F.11 dkl-graver-cdrom.sh

```
#!/bin/bash
```

```
IMAGEISO=/diskless/CDROM/isos/iso1.iso
```

```
mkisofs -o $IMAGEISO \  
-b isolinux/isolinux.bin \  
-c isolinux/boot.cat \  
-no-emul-boot -boot-load-size 4 \  
-boot-info-table \  
/diskless/CDROM/root/
```

```
# RQ: Le fichier boot.cat n'existe pas mais c'est normal (cf doc.)
```

```
echo ""  
echo "Maintenant, vous pouvez graver l'image iso suivante:"  
echo /diskless/CDROM/isos/iso1.iso  
echo ""  
echo "Vous pouvez utiliser une des commandes suivantes:"  
echo "cdrecord dev=0,0,0 blank=fast -eject /diskless/CDROM/isos/iso1.iso"  
echo "cdrecord dev=0,0,0 -eject /diskless/CDROM/isos/iso1.iso"  
echo ""
```

F.12 dkl-syslogd.sh

```
#!/bin/bash

# Ce script est necessaire pour retablir syslogd sur un diskless
# Ce script cree les fichiers necessaires a syslogd

if [ $# != 1 ]
then
echo Usage: $0 client-IP-addr
exit 1
fi

IP=$1

# RACINE doit etre / si on execute ce script a partir du diskless
# ou doit etre /diskless/tftboot/195.220.79.xxx/ si on l'execute
# a partir du serveur

# RACINE=/
RACINE=/diskless/tftboot/${IP}/

mkdir -p ${RACINE}var/log/mail
mkdir -p ${RACINE}var/log/cron
mkdir -p ${RACINE}var/log/kernel
mkdir -p ${RACINE}var/log/daemons
mkdir -p ${RACINE}var/log/lpr

touch ${RACINE}var/log/mail/info
touch ${RACINE}var/log/mail/warnings
touch ${RACINE}var/log/mail/errors
touch ${RACINE}var/log/cron/info
touch ${RACINE}var/log/cron/warnings
touch ${RACINE}var/log/cron/errors
touch ${RACINE}var/log/kernel/info
touch ${RACINE}var/log/kernel/warnings
touch ${RACINE}var/log/kernel/errors
touch ${RACINE}var/log/daemons/info
touch ${RACINE}var/log/daemons/warnings
touch ${RACINE}var/log/daemons/errors
touch ${RACINE}var/log/lpr/info
touch ${RACINE}var/log/lpr/warnings
touch ${RACINE}var/log/lpr/errors
```

```
# service syslog start
```

F.13 dkl-paquetage-install.sh

```
#!/bin/sh
if [ $# != 2 ]
then
echo Usage: $0 client-IP-addr Fichier
echo " - 1er argument: adresse IP du diskless ou le paquetage est deja installe"
echo " - 2e argument: nom du fichier liste des fichiers de /etc a repercuter"
echo "Exemple: $0 195.220.79.220 /tftpboot/195.220.79.220/tmp/MAJ-paquetages.txt"
exit 1
fi

IP=$1
ListeFic=$2

[ ! -d /diskless/tftpboot/${IP}/ ] && echo "$IP n'existe pas."
[ ! -d /diskless/tftpboot/${IP}/ ] && exit 1
[ ! -r "$ListeFic" ] && echo "$ListeFic n'existe pas."
[ ! -r "$ListeFic" ] && exit 1

# ListeDiskless contient la liste de toutes les IP des diskless
ListeDiskless=\
`ls -1 /diskless/tftpboot|grep -- '[0123456789]*\.[0123456789]*\.[0123456789]*\.[0123456789]*`

# ListeAutresDiskless contient la liste de toutes les IP des diskless
# SAUF celle du diskless qui est deja installe.
ListeAutresDiskless=
for disk in $ListeDiskless
do
if [ "$disk" == "$IP" ]
then
continue
fi
ListeAutresDiskless="${ListeAutresDiskless} ${disk}"
done

echo "Avant de continuer, verifier que ceci est correct:"
echo "ListeFic=$ListeFic"
echo "IP=$IP"
echo "ListeDiskless='echo $ListeDiskless| xargs echo'"
echo "ListeAutresDiskless='echo $ListeAutresDiskless|xargs echo'"
```

```

echo "Taper [entree] pour continuer ou CONTROL-C pour quitter"
read PAUSE

echo ""

# Ne mettre que les repertoires qui ne sont pas communs, comme /var/ et /etc/
for k in etc var
do
for i in `cat $ListeFic|grep "^/${k}/"`
do
for j in $ListeAutresDiskless
do
echo ""
echo cp -ai /diskless/tftpboot/${IP}${i} /diskless/tftpboot/${j}${i}
cp -ai /diskless/tftpboot/${IP}${i} /diskless/tftpboot/${j}${i}
done
done
done

mv -f $ListeFic ${ListeFic}.old

```

F.14 dhcpd.conf-exemple.txt

```

# allow booting
# allow bootp
ddns-update-style none;
subnet 195.220.79.0 netmask 255.255.255.0 {
    # default gateway
    option routers 195.220.79.250;
    option subnet-mask 255.255.255.0;

    option domain-name "obs.ujf-grenoble.fr";
    option domain-name-servers 195.220.79.11;

deny unknown-clients;
# Adresses dynamiques
#     range dynamic-bootp 195.220.79.124 195.220.79.125;

# Adresses statiques affectees aux PC diskless bootant
# a travers PXE
#une directive "host" pour chaque diskless
# l'option next-server donne l'adresse IP du serveur de boot
# filename donne le nom de fichier de boot (attention, pas

```

```

# le noyau linux)

# Diskless de test
host gag220 {
    allow booting;
        hardware ethernet 00:07:95:C6:9F:A8;
        fixed-address 195.220.79.220;
}

# PC de Jerome Letellier
# host gag211 {
# allow booting;
#     hardware ethernet 00:07:95:c8:52:ea;
#     fixed-address 195.220.79.211;
# }

# Diskless de test numero 2
host gag242 {
    allow booting;
        hardware ethernet 00:d0:b7:c4:dc:ba;
        fixed-address 195.220.79.242;
}

}

```

F.15 exports-exemple.txt

```

/diskless 195.220.79.220(rw,no_root_squash,insecure,async)
/diskless 195.220.79.242(rw,no_root_squash,insecure,async)

```

F.16 fstab-exemple.txt

```

# /etc/fstab du diskless

# 195.220.79.217 est l'adresse IP du serveur
# 195.220.79.220 est l'adresse IP du diskless

195.220.79.217:/diskless/tftpboot/195.220.79.220 / nfs defaults,bg,intr,noac,rsize=8192,
0 0
195.220.79.217:/diskless/shares/bin /bin nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0

```

```

195.220.79.217:/diskless/shares/usr /usr nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0
195.220.79.217:/diskless/shares/sbin /sbin nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0
195.220.79.217:/diskless/shares/home /home nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0
195.220.79.217:/diskless/shares/lib /lib nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0
195.220.79.217:/diskless/shares/commun /commun nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0
195.220.79.217:/diskless/shares/opt /opt nfs defaults,bg,intr,noac,rsize=8192,wsiz=8192
0 0

```

```

none /dev/pts devpts mode=0620 0 0
none /proc proc defaults 0 0

```

```

# DISQUETTES ET CDROM
#none /mnt/cdrom supermount dev=/lib/dev-state/ide/host2/bus1/target1/lun0/cd,fs=auto,ro
#OU: none /mnt/cdrom supermount dev=/dev/cdrom,fs=auto,ro,--,iocharset=iso8859-15,codepage=
#none /mnt/cdrom2 supermount dev=/lib/dev-state/ide/host2/bus1/target0/lun0/cd,fs=auto,ro
#none /mnt/floppy supermount dev=/dev/fd0,fs=auto,--,iocharset=iso8859-15,sync,codepage=

```

```

# DISQUES DURS LOCAUX
#/dev/ide/host2/bus0/target0/lun0/part1 /mnt/disk ext3 defaults 0 0
#/dev/ide/host2/bus0/target0/lun0/part3 /local ext3 defaults 0 0
#/dev/ide/host2/bus0/target0/lun0/part2 swap swap defaults 0 0

```

F.17 isolinux.cfg

```

# Fichier de configuration isolinux.cfg

default linux
# Pour afficher le menu
#prompt 1
# Pour ne pas afficher le menu
prompt 0

#####
# CHOIX 1: "linux"
label linux
# A la racine du CDROM, repertoire isolinux/alt0
KERNEL alt0/vmlinuz
# A ce stade, le noyau est charge du CD en memoire

```

```
# Parametres passes au noyau pour booter sur le reseau
APPEND root=/dev/nfs devfs=mount ip=:::::dhcp nfsroot=/diskless/tftpboot/%s

#####
# CHOIX 2: "linux-2"
label linux-2
# A la racine du CDRom, repertoire isolinux/alt2
KERNEL alt1/vmlinuz
# A ce stade, le noyau est charge du CD en memoire
# Parametres passes au noyau pour booter sur le reseau
APPEND root=/dev/nfs devfs=mount ip=:::::dhcp nfsroot=/diskless/tftpboot/%s
```

G À propos de ce document

Ce document et les scripts qui l'accompagnent sont disponibles sur :
<http://alban.apinc.org/>

Ce document a été écrit par Alban Créquy pendant son stage au LAOG du 1 juillet au 8 août. Son utilisation en dehors d'un cadre privé est interdit sauf accord préalable.