sommaire de Medintux et net boot Mageia2

---

1 Le net-boot (sommaire)

1-1 Merci Mr Colin Guthrie

Cette aventure n'aurait pas été possible sans le superbe article de Mr Colin Guthrie : http://colin.guthr.ie/2012/09/nfs-root-media-centre-v2-0/
Chapeau bas monsieur et encore merci. (sommaire)

1-2 Extreme progamming et open space (sommaire)

Je ne suis pas informaticien, mais juste quelqu'un qui aime un peu mettre les mains sous le capot du moteur. Je suis aussi l'auteur principal et à l'origine du projet MedinTux http://medintux.org .

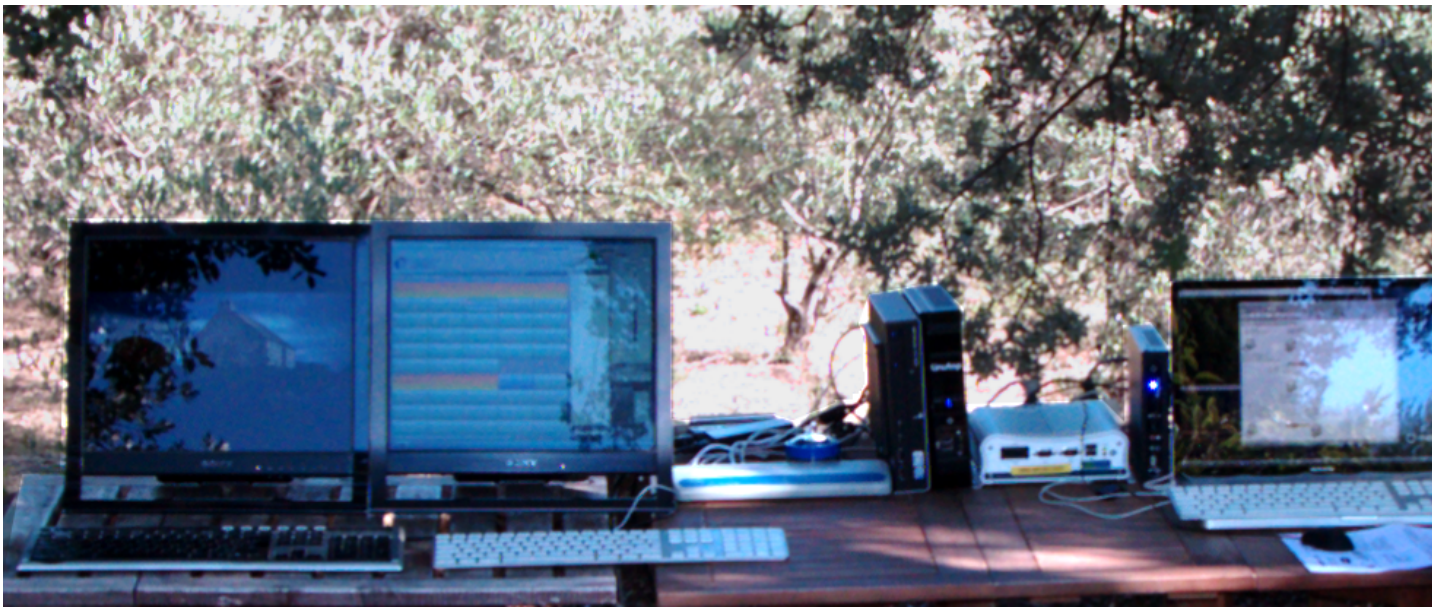
Dans le cadre d'un déploiement de MedinTux sur plusieurs postes, j'ai imaginé une solution client serveur, dont le serveur mettrait à disposition des clients :

- l'OS une Mageia2 qui est pour moi la meilleure distribution linux pour ce cas;
- les applicatifs de la suite MedinTux;
- la base de donnée SQL.

Les postes clients légers seraient, sans ventilateur, sans disque dur (des terminaux). Les avantages sont :

- maintenance d'une seule archive MedinTux;
- maintenance d'un seul OS;
- faible coût des postes de travail qui sont silencieux, sans partie mécanique, pas de poussières brassée;
- faible consomation énergetique.

Ci-dessous la salle d'informatique.



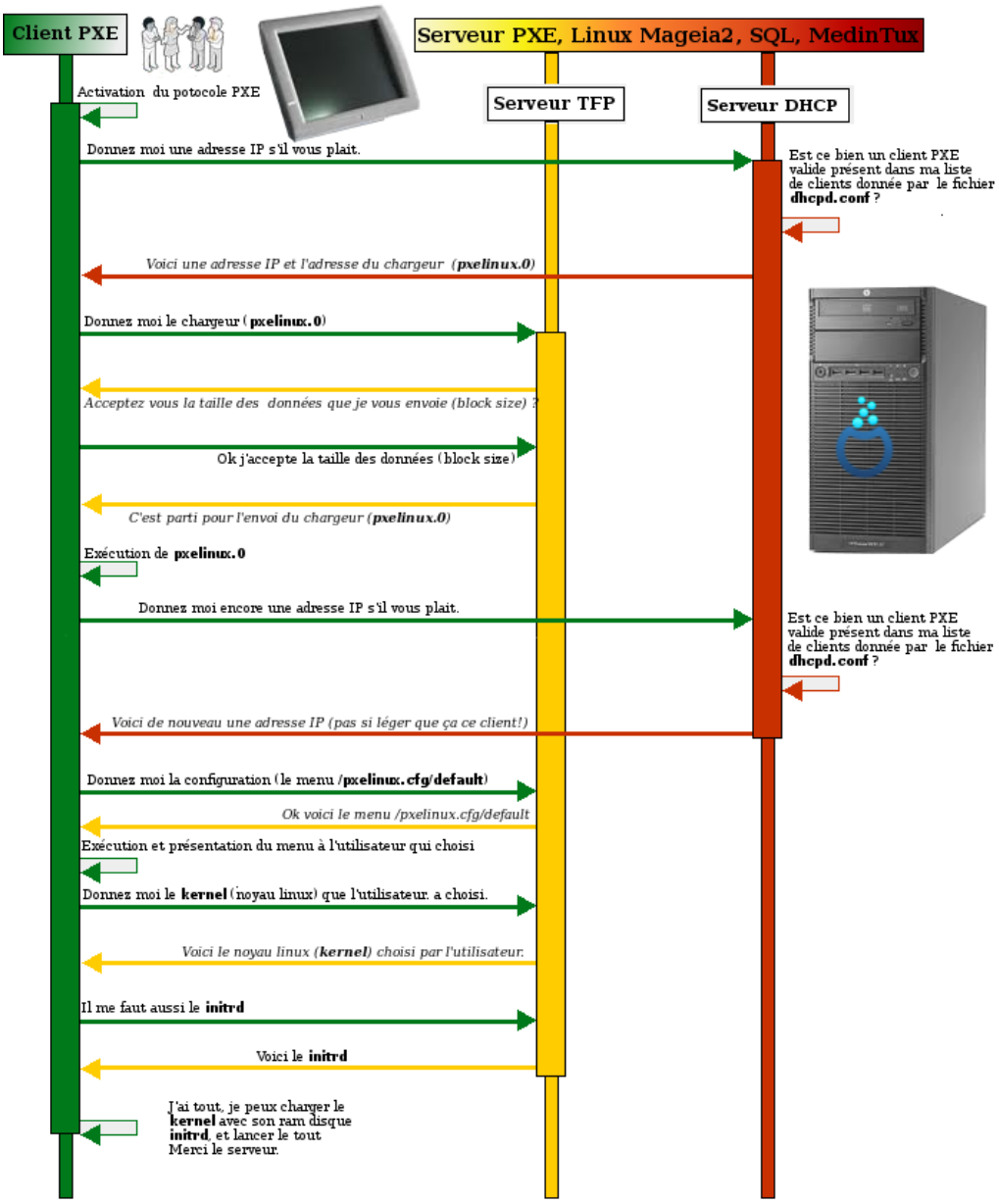
Oh il y a un mac ?? mais avec une Mageia hein !!. C'est le serveur de test. Les quatres clients légers sont au milieu. Ils ont des configurations de cartes graphiques, de processeurs, et cartes ethernet un peu disparates. De même pour les écrans.

Mais vous pouvez constater que les clients légers ont tous démarré une **Mageia2** et qu'ils lancent automatiquement **MedinTux** prêt à travailler, pas de disque dûr ils sont allés chercher la superbe Mageia2 et le pitoyable MedinTux sur l'infâme mac OSX .

Vous pouvez voir des conditions de travail open space, dans un environnement d'extreme programming : quand il pleut il faut bâcher, quand il fait froid, on se couvre et on met des moufles, quand c'est la saison des glands qui tombent on met des casques, bref ce n'est pas drôle tous les jours.

1-3 Rappel (sommaire)

Le principe : un ordinateur (appelé **client net-boot** ou **client léger**) est configuré via son **bios** pour pouvoir démarrer via sa carte ethernet, en allant chercher sur le réseau tous les éléments nécessaires à son démarrage.



Shéma adapté et traduit à partir d'une image trainant sur mon disque dur, l'auteur m'est inconnu, si il se reconnaît, merci.

Le protocole le plus utilisé est **PXE**. Si il n'est pas activé (c'est le plus souvent le cas) vous devrez rentrer dans le bios de ce client léger, et activer le démarrage PXE. Lors du démarrage de cette machine, PXE tente le dialogue suivant :

Les clients légers de test.

-- **Le client léger** : Holà les potes, il me faut une adresse IP pour booter (il envoie une demande d'IP en indiquant au passage son adresse MAC

exemple `38:60:77:bd:50:9e` sur l'adresse 0.0.0.0. L'adresse IP 0.0.0.0 est particulière car elle est écoutée par toute machine du réseau disposant d'un dhcp) il me faut aussi un **noyau linux** appelé aussi **Kernel** et le **initrd** (disque ram) qui va bien pour démarrer ce noyau.

-- **Une machine serveur** disposant d'un service ou serveur **dhcp** (couplé à un **tftp**) en écoute sur ce réseau (espérons que ce soit notre serveur) reçoit cette requête sur l'adresse 0.0.0.0 voit dans son fichier de configuration **dhcpd.conf** que cette adresse MAC 00:10:f3:18:88:e7 la concerne et répond avec les indications notées dans la section correspondant à cette adresse MAC comme indiqué dans son fichier **/etc/dhcpd.conf** :

```
hardware ethernet 38:60:77:bd:50:9e;   # Holà je suis concerné par cette demande d'adresse IP liée à la
                                       # machine ayant pour adresse physique MAC 38:60:77:bd:50:9e
fixed-address 192.168.0.24;            # mon fichier dhcpd.conf me dit de te donner l'adresse 192.168.0.24
next- server 192.168.0.10;             # moi je suis en 192.168.0.10 où tu trouveras tout pour booter (J'active le tftp pour ça)
filename "pxelinux.0";                 # notamment le petit programme pxelinux.0 que tu téléchageras de chez moi via tftp
                                       # et chez toi tu le lances, il activera le menu de choix default qu'il trouve
                                       # chez moi dans le répertoire pxelinux.cfg ce menu te renseignera après le choix
                                       # de l'utilisateur sur l'emplacement du kernel à booter (ligne KERNEL du menu default )
                                       # et les options d'appel liées à ce kernel (ligne APPEND) après le choix de l'utilisateur
                                       # tu n'as plus qu'à aller chercher tout ça chez moi et booter, je laisse le tftp actif pour toi.
```

2 Le net-boot Mageia2 pour MedinTux (sommaire)

La distribution NetBoot-MedinTux se fonde sur la magnifique **Mageia2**. Un script que j'ai écrit (indulgence) : **installm2.sh** automatise beaucoup d'opérations permettant de créer le **filesystem** dans un répertoire de votre choix avec tout ce qui va bien pour booter une **mageia2** par le net avec tous les paquets nécessaires au fonctionnement de **MedinTux**. Dans notre exemple il est en : **/home/urg/nb/fm**

2-1 L'environnement, plantons le décor avant de planter les machines (sommaire)

Il nous faut :

2-1-1 un dhcp (le guetteur) sur le serveur : (sommaire) (installer les paquets dhcp-server)

Le service dhcd est en permanence à l'écoute du réseau et délivre des adresses IP aux machines qui en font la demande.

2-1-1-1 Le fichier /etc/dhcpd.conf détermine les adresses IP des postes clients en fonction de leur adresse MAC. Après chaque modification du fichier **dhcpd.conf**, il convient de relancer le serveur dhcp en faisant dans une console root (enfin c'est ce que je fais):

```
killall dhcpd;
dhcpd;
```

Mon fichier **/etc/dhcpd.conf**

```
# cat /etc/dhcp/dhcpd.conf
# sur mac OSX dhcp est bloque si partage internet actif car netbootpd est lance

ddns-update-style interim;
authoritative;
allow booting;
allow bootp;
option domain-name "rolanddhcp0.org";
default-lease-time 7200;
max-lease-time 86400;
log-facility local0;


#------------ le sous reseau pour BRAMADOU -----------------------------
subnet 192.168.100.0 netmask 255.255.255.0 {
      pool {
            range 192.168.0.20 192.168.0.99; # la plage d'adresses a distribuer
            }
      option routers 192.168.0.254;                    # la passerelle
      option domain-name "roland.bramadou.org";        # un nom de domaine
      option domain-name-servers 212.27.40.240,212.27.40.241; # les DNS
}

#......... Group the PXE bootable hosts together .....................
group {

host machine-1 {
      #-------- bramadou ---------
      next-server 192.168.0.10;            # le serveur dhcp/tftp
      hardware ethernet 00:14:0B:80:33:06; # Adresse matérielle du client TFTP
      fixed-address 192.168.0.25;
      filename " pxelinux.0";
      option routers 192.168.0.254;        # la passerelle
      option domain-name "roland.bramadou.org"; # un nom de domaine
      option domain-name-servers 212.27.40.240,212.27.40.241; # les DNS
      }
host machine-4 {
      #-------- bramadou ---------
      next-server 192.168.0.10;            # le serveur dhcp/tftp
      hardware ethernet 38:60:77:bd:50:9e; # Adresse matérielle du client TFTP
      fixed-address 192.168.0.24;
      filename "pxelinux.0";
      option routers 192.168.0.254;        # la passerelle
      option domain-name "roland.bramadou.org"; # un nom de domaine
      option domain-name-servers 212.27.40.240,212.27.40.241; # les DNS
```

```
    }
}
```

2-1-1-2 Conflits entre DHCP (sommaire) : vous avez une box internet qui elle aussi attribue des adresses IP via son dhcp, et prend la main. Celui de votre serveur n'est plus concerné et n'active donc pas le net boot PXE indiqué dans chaque définition de poste client comme vous pouvez le voir ci-dessous.
Solution un peu bancale : sur la box internet fixer une plage d'adresses limitant les adresses IP à strictement celles liées au matériel concerné : Exemple : le serveur 192.168.0.10 les blocs PCL 192.168.0.11 et PCL 192.168.0.12 et les baux dhcp de la box internet sont donnés en fonction des MAC adresses surtout de façon à ce que le serveur ait toujours la même adresse (chez moi 192.168.0.10).
Si d'autres demandes d'IP (notamment des clients net-boot) arrivent sur la box internet, alors ne pouvant répondre car saturée, c'est le dhcp du serveur qui prend la main et joue son rôle pour les clients net-boot.

2-1-2 un tftp (répertoire lanceur) sur le serveur (sommaire) : (installer les paquets `tftp-server`)

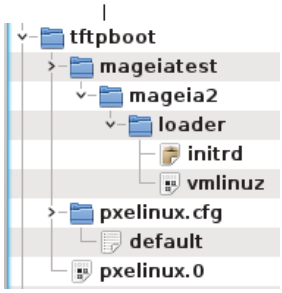
Le service **tftp** est appelé automatiquement par **dhcp** pour que le **client net-boot** puisse télécharger sur le serveur les éléments dont il a besoin.

Ce service est contitué :

2-1-2-1 de l'exécutable tftpd (automatiquement appellé par le dhcp lorsqu'un net-boot est nécessaire)

2-1-2-2 de son fichier de configuration lancement : **/etc/xinetd.d/tftp** comme d'habitude avec linux.

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
# protocol. The tftp protocol is often used to boot diskless \
# workstations, download configuration files to network-aware printers, \
# and to start the installation process for some operating systems.
service tftp
    { disable = no
      socket_type = dgram
      protoco = udp
      wait = yes
      user = root
      server = /usr/sbin/in.tftpd
      server_args = -s /var/lib/tftpboot -r blksize
      per_source = 11      ^
      cps = 100 2          |
      flags = IPv4         |
}                          |
```



dans ce fichier :

la ligne : **server = /usr/sbin/in.tftpd** indique l'emplacement de l'exécutable (on est content)
la ligne : **server_args = -s /var/lib/tftpboot -r blksize** indique le répertoire de données dans lequel doivent se trouver les menus et éléments de boot (kernel ram disque de démarrage lié à ce kernel etc.).

2-1-3 un serveur nfs (le serveur livreur) sur le serveur : (sommaire)

2-1-3-1 On nous a encore caché un truc (sommaire)

Quoi encore un truc à paramétrer ? On ne vous a pas encore tout dit.

Une fois que le **client net-boot** a réussi à booter son kernel celui-ci (le kernel et sa suite) aura besoin d'un espace disque dans lequel il va trouver le reste du système d'exploitation, et les outils dont il va avoir besoin le long de sa vie de forcené. On apelle cet espace disque le **filesystem**. C'est le truc avec les répertoires **/boot /usr /lib /var /etc** etc. Ca en jette au moins un nom pareil le filesystem.

Bon alors il existe plusieurs solutions pour lui en offrir un ... les squash.fs et autres filesystem. sous formes compressées et monolithiques.
Moi j'ai choisi pour l'instant de tout créer dans un répertoire linux classique de mon choix, de tout mettre dedans et d'indiquer au **kernel & cie** que c'est là que ça se passe. Dans notre exemple il est en : **`/home/urg/nb/fm`** sur le serveur.

Avantage : lors de la mise au point on y accède à partir du serveur avec des fichiers en clair, il est possible d'éditer les fichiers de configuration, faire du copier coller, faire du chroot. Le confort.
Inconvénient : ce n'est pas compressé donc un peu plus long à booter.

2-1-3-2 Intérêt du partage NFS (sommaire)

Comment mettre à disposition ce répertoire pour le kernel. Et bien tout simplement (sic.) avec le système de fichiers partagés via le net, apellé NFS, abréviation de **N**et **F**ile **S**ystem.

Une machine qui a activé un **serveur NFS** propose à d'autres machines distantes dites **clientes nfs** et autorisées (via encore un fichier de configuration) un accès à certains de ses répertoires qu'elle met à disposition pour ces machines distantes. On dit que le **serveur nfs exporte** ses répertoires. Les machines distantes peuvent (selon les autorisations paramétrées en écriture et ou lecture) y accéder comme si ces répertoires étaient sur leur propre disque dur.

Cela implique deux choses :

- Installer les services d'un serveur nfs sur le serveur principal qui exporte ses répertoires (dans notre cas nous exporterons le filesystem linux mageia2)

- Installer chez nos clients net-boot les fonctionnalité d'accès client nfs, le kernel et surtout son ram disque initrd élaboré pour notre client net-boot est donc prévu pour démarrer avec.

2-1-3-3 Installation et paramétrage du serveur NFS: (sommaire) (installer les paquets nfs-utils)

2-1-3-3-1 Démarrage arrêt : (sommaire) le serveur nfs est démarré et arrêté avec les commandes suivantes (sous console root).

```
/etc/init.d/nfs-server restart    # redémarre le serveur nfs (à faire à chaque modification de sa configuration)
/etc/init.d/nfs-server stop       # arrëte le serveur nfs
/etc/init.d/nfs-server start      # démarre le serveur nfs (il le reste même aprés reboot)
```

2-1-3-3-2 Configuration paramètrage coté serveur : (sommaire) le fichier /etc/exports

Dans cet exemple le **filesystem linux mageia2** à exporter se trouve sur notre serveur en :
/home/urg/nb/fm

Mon fichier **/etc/exports** comporte la ligne suivante :

```
/home/urg/nb/fm 192.168.0.11/30(fsid=root,ro,no_root_squash,no_subtree_check,async,insecure)
```

qui autorise tous les clients de l'adresse **192.168.0.11** à **192.168.0.30** à lire comme si ils étaients chez eux le répertoire **/home/urg/nb/fm**
Cette ligne montre que le répertoire du **filesystem** est exporté en lecture seulement. Les autres paramètres un peu obscurs sont nécessaires au bon fonctionnement du démarage du kernel.

Il y a aussi la ligne suivante :

```
/home/urg/Documents 192.168.0.11/30(rw,sync)
```

qui autorise tous les clients de l'adresse **192.168.0.11** à **192.168.0.30** à lire et écrire comme si ils étaients chez eux dans le répertoire **/home/urg/Documents** du serveur.
Cela permettra aux clients de partager le même répertoire pour les documents communs, et surtout il s'y trouvera le répertoire complet de **medintux**, permettant à chaque client de démarrer et disposer de la suite medintux.

2-1-3-3-3 Configuration paramètrage coté client : (sommaire) (installer les paquets nfs-utils-clients)

Les paquets **nfs-utils-clients** étant installés il suffit dans une console root chez le client, de taper :

```
mount -t nfs 192.168.0.10:/home/urg/Documents /home/urg-01/Documents
```

Le répertoire **/home/urg/Documents** du serveur à l'adresse **192.168.0.10** par montage chez le client en **/home/urg-01/Documents,** devient partagé. Ce qui est modifié dans ces répertoires d'un côté ou de l'autre sera visible des deux cotés. Il faut bien sûr avant que la commande **mount** soit exécutée, et que le répertoire **/home/urg-01/Documents** existe déjà chez le client. Tout ce qu'il contenait auparavant sera non accessible. Cela le redeviendra lorsque le répertoire distant sera démonté (commande **umount**).

Bonne nouvelle, vous n'aurez même pas à taper la commande qui permet cette magie, car nous verrons plus loin qu'elle est directement exécutée par un script lors du démarrage du client.

2-2 La séquence de boot d'un client : (sommaire)

client net-boot
pas de disque dur. Null quoi
MAC 00:10:f3:18:88:e7

holà je n'ai pas de disque dur
et je dois booter (démarrer)
allez lançons une petite requête
sur le 0.0.0.0 dès fois que l'on m'écoute

serveur à l'écoute
mageia2-MedinTux
192.168.0.10

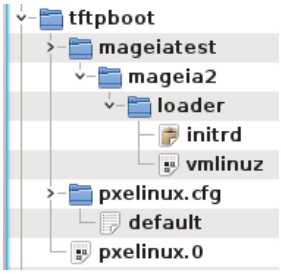
---> ? ? 0.0.0.0 ? ? --->
et moi être
MAC 38:60:77:bd:50:9e

ouaip mon dhcp qui lui écoute
me dit avec dhcpd.conf que
MAC `38:60:77:bd:50:9e`
fait partie de mes administrés (sommaire)

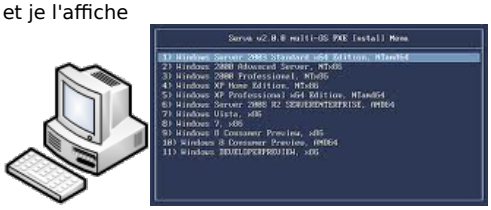
```
host machine-4 {
       #-------- bramadou ---------
       next-server 192.168.0.10;             # moi être le serveur dhcp ici en 192.168.0.10
       hardware ethernet 38:60:77:bd:50:9e; # Adresse matérielle MAC du client TFTP
       filename "pxelinux.0";                 # tu dois te servir de pxelinux.0 et pour
       fixed-address 192.168.0.24;           # commencer je te donnes l'IP 192.168.0.24
       }                                                                  /
                                                                         /
                                          <----- 192.168.0.24 <--------/
```

merci pour l'IP 192.168.0.24

en plus `tu veux booter, bon`
j'ouvre mon tftp pour toi en
192.168.0.10 tu y trouveras tout



Le client : Ok je prends le menu **tftpboot/pxelinux.cfg/default** (sommaire)
et je l'affiche



```
DEFAULT menu.c32
MENU MARGIN 0
MENU ROWS -9
MENU TABMSG
MENU TABMSGROW -3
MENU CMDLINEROW -3
MENU HELPMSGROW -4
MENU HELPMSGENDROW -1
MENU COLOR SCREEN 30;47
MENU COLOR BORDER 30;47
MENU COLOR TITLE 30;47
MENU COLOR SCROLLBAR 30;47
MENU COLOR SEL 37;40
MENU COLOR UNSEL 30;47
MENU COLOR CMDMARK 30;47
MENU COLOR CMDLINE 30;47
MENU COLOR TABMSG 37;40MEN
U COLOR DISABLED 37;40
MENU COLOR HELP 37;40
timeout 20

LABEL Mageia-Medintux
KERNEL mageiatest/mageia2/loader/vmlinuz
APPEND ip=dhcp initrd=mageiatest/mageia2/loader/initrd root=nfs:192.168.0.10:/home/urg/nb/fm:ro,rsize=32768,wsize=32768,exec,hard,proto=tcp LANG=fr
rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 medintuxRootFs@192.168.0.10:/home/urg/Documents@

LABEL harddisk
localboot 0x80
```

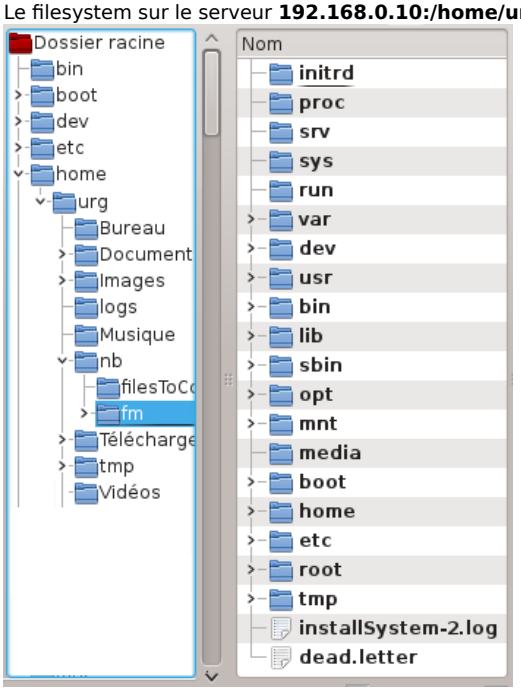
L'utilsateur a choisi : **LABEL Mageia-Medintux** (sommaire)

Donc le kernel à booter est indiqué à la ligne **KERNEL** et c'est : **mageiatest/mageia2/loader/vmlinuz** du répertoire **tftpboot** que le serveur m'a mis à disposition.

De même le ram disque de démarrage y est indiqué à la ligne **APPENDspan> initrd=mageiatest/mageia2/loader/initrd** du répertoire tftpboot.

Tiens le filesystem est indiqué comme étant un répertoire partagé en lecture seulement (ro) du serveur **root=nfs:192.168.0.10:/home /urg/nb/fm:ro**

Le filesystem sur le serveur **192.168.0.10:/home/urg/nb/fm**



de plus il y a une drôle d'option : medintuxRootFs@192.168.0.10:/home/urg/Documents@ je crois qu'un script s'en sert pour localiser le répertoire de **MedinTux** qui s'y trouve (on voit là qu'il peut être sur un autre serveur).

Le client : Allez on lance tout ça avec le chargeur **tftpboot/pxelinux.0** et on voit .....
super j'ai chargé la **Mageia** et **Medintux** s'est lancé automatiquement magique.

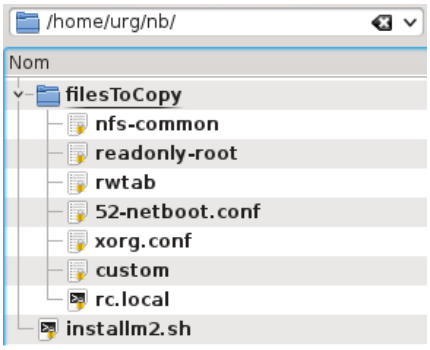
3 La fabrique à filesystem (sommaire)

Le filesystem on en parle on en parle mais comment le fabriquer ? et bien que non spécialiste j'ai fait un petit script bash qui permet d'automatiser un peu toutes les tâches de génération du filesystem.

3-1 Les ingrédients (sommaire)

Ma fabrique est assez rudimentaire et se compose d'un script : **installm2.sh**
dont la mission est de remplir le file system avec tous les paquets et fichiers nécessaires.

Ce script fait aussi appel à certains fichiers qu'il place ou replace dans les bons endroits du file system. Ces fichiers (scripts et fichiers de configuration modifiés pour la cause) se trouvent dans le répertoire **filesToCopy** de mon usine à filesystem. Nous les verrons plus loin.



3-2 Le script : installm2.sh (sommaire)
Indulgence S.V.P. je ne suis pas un spécialiste du bash les améliorations et corrections sont les bienvenues.

3-2-1 Paramètres d'appel du script : (sommaire)
Ce script s'appelle en mode **root** avec **trois paramètres** en ligne de commande.

3-2-1-1 Premier paramètre emplacement du filesystem à créer (sommaire)
Le premier paramètre est l'emplacement de destination où doit être fabriqué le filesystem exemple sur le serveur en : **/home/urg/nb/fm**

3-2-1-2 Deuxième paramètre emplacement du répertoire filesToCopy (sommaire)
Le deuxième paramètre est l'emplacement des divers fichiers dont se sert le script, pour les copier dans le filesystem à créer. Exemple sur le serveur en **/home/urg/nb/filesToCopy**.

3-2-1-3 Troisième paramètre le dns (sommaire)
Si vous voulez que les machines qui boutent sur ce filesystem bénéficient d'internet il vous faut définir dans ce paramètre **l'adresse IP** du résoluteur de nom de domaine ou **DNS** exemple : **212.27.40.240**

Si par la suite vous voulez le changer il vous faudra modifier dans le filesystem le fichier : **/etc/resolvconf/resolv.conf.d/base** avec la nouvelle adresse comme ci-après : **nameserver 212.27.40.241**

3-2-1-4 Exemple d'appel (sommaire)

```
./installm2.sh '/home/urg/nb/fm' '/home/urg/nb/filesToCopy' '212.27.40.240'
```

et attendre car il faut plusieurs heures.

3-2-2 Le fichier de log : installSystem-2.log (sommaire)
Ce fichier se situe à la racine du filesystem et contiendra et tracera tous les messages résultant des diverses opérations liées à l'exécution du script **intallm2.sh**

3-3 Les fichiers du répertoire : filesToCopy (sommaire)

3-3-1 Le fichier rc.local (sommaire)
Ce fichier sera copié dans le filesystem en **/etc/rc.d** . Ce fichier est exécuté par le système lors du boot et me sert à régler plusieurs choses liées à **MediTux** .

3-3-1-1 avoir un clavier français (sommaire)
L'instruction `loadkeys fr` y pourvoie

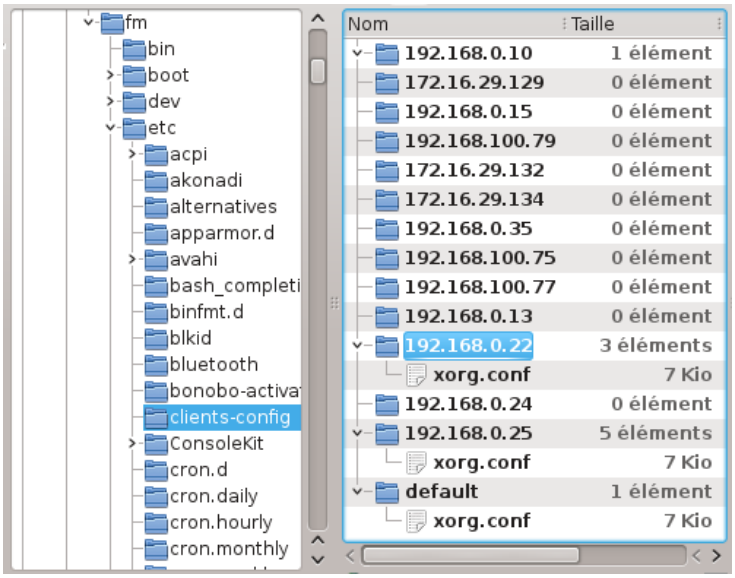
3-3-1-2 le montage du répertoire nfs pour accéder à MediTux (sommaire)
Nous avions vu en 2-2 lors de la séquence de boot qu'un curieux paramètre d'appel **medintuxRootFs@192.168.0.10:/home/urg/Documents@** était présent.
Ce paramètre est récupéré ici dans ce script, et permet de connaître et monter le répertoire nfs distant du serveur, dans lequel se trouve la suite MediTux, sur le poste client, afin de rendre accessibles les éléments de la suite **MediTux** aux utilisateurs. Ce répertoire peut donc être changé de place en modifiant le menu d'appel dans **tftpboot/pxelinux.cfg/default** . Cela permet de tester plusieurs versions.

```
LABEL Mageia-Medintux
KERNEL mageiatest/mageia2/loader/vmlinuz
APPEND ip=dhcp initrd=mageiatest/mageia2/loader/initrd-new root=nfs:192.168.0.10:/home/urg/nb/fm:ro,rsize=32768,wsize=32768,exec,hard,proto=tcp LANG=fr
rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 medintuxRootFs@192.168.0.10:/home/urg/Documents@
```

3-3-1-3 Copier la bonne définition d'écran et de carte graphique (sommaire)

Les postes clients n'ont pas tous la même carte graphique. Pour que l'affichage soit correct il convient sur certains postes de modifier les paramètres d'affichage. Pour l'instant en attendant que je trouve une méthode d'auto configuration, lors du démarrage du système, le fichier **xorg.conf** lié à ce poste est copié à la volée dans `/etc/X11` et `/etc/X11/xorg.conf.d/`

Pour repérer le bon fichier **xorg.conf** lié à ce poste, le système récupère l'adresse IP du poste client et à partir de cette adresse, cherche le fichier **xorg.conf** dans un répertoire `/etc/clients-config/xx.xx.xx.xx` où `xx.xx.xx.xx` est l'adresse IP du client. Pour le générer sur le poste client la première fois j'utilise une **mageia2 live**, et je le copie. Bricolage infâme.... Ce qui est certain c'est qu'une distribution live automatise tout cela. Honte à moi, je n'ai pas encore trouvé le moyen de le faire.




```
#!/bin/sh
#
### BEGIN INIT INFO
# Provides: rc.local
# X-Mandriva-Compat-Mode
# Default-Start: 2 3 4 5
# Short-Description: Local initialization script
# Description: This script will be executed at the end of the boot process.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
### END INIT INFO

touch /var/lock/subsys/local

loadkeys fr

# ------ reperer le serveur nfs Documents où se trouve medintux (varie selon ligne de commande) ...................
servnfs=`grep 'medintuxRootFs@' /var/log/syslog|cut -d@ -f2|sed 's/[ \t]*$//'`
echo "MEDINTUX CUSTOM ----- mount -t nfs $servnfs /mnt/nfs -----"
mount -t nfs $servnfs /mnt/nfs

# ----------------- copy default xorg.conf .....................
echo "MEDINTUX CUSTOM ----- copy /etc/clients-config/default/xorg.conf to /etc/X11/xorg.conf.d/"
cp -Rf /etc/clients-config/default/xorg.conf /etc/X11/xorg.conf.d/

# ----------------- copy specific xorg.conf from client IP .....................
ipadr=`/sbin/ifconfig eth0|grep 'inet adr:'|cut -d: -f2|cut -dB -f1|sed 's/[ \t]*$//'`;
clientfolder=/etc/clients-config/$ipadr;
if [ -f $clientfolder/xorg.conf ]
then
echo "MEDINTUX CUSTOM ----- copy $clientfolder/xorg.conf to /etc/X11/xorg.conf. d/"
cp -Rf $clientfolder/xorg.conf /etc/X11/xorg.conf.d/
else
echo "MEDINTUX CUSTOM --------- can't copy specific xorg.conf ----------------"
echo "MEDINTUX CUSTOM because unknow $clientfolder/xorg.conf"
fi
exit 0
```

3-3-2 Le fichier nfs-common (sommaire)

Il sera copié dans **/etc/sysconfig/** car celui d'origine comporte la ligne **NEED_IDMAPD=no** au lieu de **NEED_IDMAPD=yes**

3-3-3 Le fichier readonly-root (sommaire)

Il sera aussi copié dans **/etc/sysconfig/** car celui d'origine comporte la ligne **READONLY=no** au lieu de **READONLY=yes**
Ce paramètre est important à retenir car permet de mettre le systeme de fichier en lecture/écriture si positionné sur READONLY=yes nous verrons par la suite que cela peut être utile.

3-3-4 Le fichier rwtab (sommaire)

Il sera copié dans **/etc/** car celui d'origine comportait la ligne **/var/lib/dbus** qu'il convient de ne pas mentionner.

3-3-5 Le fichier 52-netboot.conf (sommaire)

Il sera copié dans **/etc/dracut.conf.d/** car n'existe pas d'origine, il sert à **dracut** pour la génération du **ramdisque**.

3-3-6 Le fichier xorg.conf (sommaire)

Fichier de configuration par défaut, sera copié dans **/etc/X11/xorg.conf.d/xorg.conf.d/** et **/etc/clients-config/default/** pour l'affichage.

3-3-7 Le fichier custom (sommaire)

Fichier de configuration des répertoires à mettre en ecriture. Il sera copié dans **/etc/rwtab.d/**
Le contenu de ce fichier est ci-dessous

```
empty /root
files /home/urg-01
files /etc/X11/xorg.conf.d
files /etc/clients-config
files /var/run
files /var/lock
files /var/lib/dbus
```

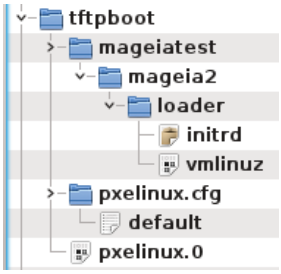
nous voyons notamment que les répertoires **/home/urg-01 /etc/X11/xorg.conf.d** et **/etc/clients-config** sont en écriture.

4 La fabrique du ram disque de démarrage (sommaire)

Nous avons vu que lors du démarrage le kernel se sert d'un mini disque de démarrage dans lequel il place les éléments nécessaires. (drivers matériels initialisations diverses etc ..)
Ce disque doit ëtre créé pour ëtre adapté à notre systëme.

Ce disque est sous forme monolithique et dans notre exemple c'est : **initrd** comme **rd** abréviation de **r**am **d**isque et **init** car sert à l'**init**ialisation du kernel. Dans notre exemple le kernel est **vmlinuz**.

Le **vmlinuz** placé dans le répertoire **tftpboot/mageiatest/maeia2/loader/** et est une recopie de celui **vmlinuz-3.3.8-desktop-2.mga2** qu'a créé notre fabrique de filesystem dans le répertoire **/home/urg/nb/fm/boot/** .



notez que tous les emplacements dans le répertoire **tftpboot/** peuvent ëtre modifiés, on peut par exemple faire :

tftpboot/
vmlinuz
initrd
pxelinux.cfg/
pxelinux.0

4-1 Le cas dracut (sommaire)

Dans l'article de **Mr Guthrie** il est indiqué que la génération du ram disque se fait avec l'outil **dracut** utilisé de la façon suivante :
(chemins adaptés à mon exemple)

```
dracut -c /home/urg/nb/fm/etc/dracut.conf --confdir /home/urg/nb/fm/etc/dracut.conf.d -f /home/urg/nb/fm/initrd 3.3.6-desktop-2.mga2
```

Chez moi cela produit bien un fichier **initrd 3.3.6-desktop-2.mga2** à la racine du répertoire /home/urg/nb/fm/ je le copie en **tftpboot/mageiatest/maeia2/loader/** le renomme **initrd** et bien cela ne marche pas !!!. J'ai passé des jours la dessus pour me rendre compte que la version de dracut livrée avec ma mageia2 est buguée j'ai donc récupéré sur le site de dracut, une archive que j'ai installée.

4-2 récupérer un dracut fonctionnel (sommaire)

4-2-1 récupérer les fichiers de dracut (sommaire)

Installation préalable de **git** avec la commande suivante **urpmi git**
se mettre dans une console **root**
se placer dans le répertoire où l'on veut télécharger dracut exemple
cd /home/urg/Documents
Récupérer dracut avec la commande suivante :
git clone git://dracut.git.sourceforge.net/gitroot/dracut/dracut
cela téléchargera dans **/home/urg/ Documents** un répertoire **dracut** tout frais venant du site de developpement.

4-2-2 installer les fichiers de dracut (sommaire)

Mon installation n'est pas classique car je n'ai pas voulu installer la suite gcc sur mon serveur pour juste compiler l'utilitaire d'installation.
Comme dracut est essentiellement composé de fichiers bash, leur recopie dans les endroits adéquats a fonctionné. On y va.

```
mv -f /usr/lib/dracut/modules.d /usr/lib/dracut/modules.d.old;

cp -Rf /home/urg/Documents/dracut/modules.d /usr/lib/dracut/;

mv -f /usr/lib/dracut/dracut-functions /usr/lib/dracut/dracut-functions-old;
mv -f /usr/lib/dracut/dracut-functions.sh /usr/lib/dracut/dracut-functions.sh-old;
mv -f /usr/lib/dracut/dracut-initramfs-restore /usr/lib/dracut/dracut-initramfs-restore-old;
# mv -f /usr/lib/dracut/dracut-install /usr/lib/dracut/dracut-install-old;
mv -f /usr/lib/dracut/dracut-logger.sh /usr/lib/dracut/dracut-logger.sh-old;
```

```
mv -f /usr/lib/dracut/dracut-version.sh /usr/lib/dracut/dracut-version.sh-old;

cp -f /home/urg/Documents/dracut/dracut-functions.sh /usr/lib/dracut/;
mv -f /usr/lib/dracut/dracut-functions.sh /usr/lib/dracut/dracut-functions;
cp -f /home/urg/Documents/dracut/dracut-functions.sh /usr/lib/dracut/;

cp -f /home/urg/Documents/dracut/dracut-initramfs-restore.sh /usr/lib/dracut/;
mv -f /usr/lib/dracut/dracut-initramfs-restore.sh /usr/lib/dracut/dracut-initramfs-restore;
cp -f /home/urg/ Documents/dracut/dracut-initramfs-restore.sh /usr/lib/dracut/;

cp -f /home/urg/Documents/dracut/dracut-logger.sh /usr/lib/dracut/;

mv -f /usr/bin/mkinitrd-dracut /usr/bin/mkinitrd-dracut-old;
cp -f /home/urg/Documents/dracut/mkinitrd-dracut.sh /usr/bin/;
mv -f /usr/bin/mkinitrd-dracut.sh /usr/bin/mkinitrd-dracut;
cp -f /home/urg/Documents/dracut/mkinitrd-dracut.sh /usr/bin/;

mv -f /usr/bin/mkinitrd /usr/bin/mkinitrd-old;
mv -f /usr/bin/mkinitrd-dracut.sh /usr/bin/mkinitrd;
cp -f /home/urg/Documents/dracut/mkinitrd-dracut.sh /usr/bin/;

mv -f /usr/bin/lsinitrd /usr/bin/lsinitrd-old;
mv -f /usr/bin/lsinitrd-dracut /usr/bin/lsinitrd-dracut-old;
cp -f /home/urg/Documents/dracut/lsinitrd.sh /usr/bin/;
mv -f /usr/bin/lsinitrd.sh /usr/bin/lsinitrd;
cp -f /home/urg/Documents/dracut/lsinitrd.sh /usr/bin/;
mv -f /usr/bin/lsinitrd.sh /usr/bin/lsinitrd-dracut;

mv -f /usr/bin/dracut /usr/bin/dracut-old;
mv -f /usr/bin/dracut-catimages /usr/bin/dracut-catimages-old;

cp -f /home/urg/Documents/dracut/dracut.sh /usr/bin/;
cp -f /home/urg/Documents/dracut/dracut-catimages.sh /usr/bin/;

mv -f /usr/bin/dracut.sh /usr/bin/dracut;
mv -f /usr/bin/dracut-catimages.sh /usr/bin/dracut-catimages;
```

4-3 Générer et installer le ram disque et kernel (sommaire)

4-3-1 Générer le ram disque avec dracut (sommaire)

dracut -o "" --omit-drivers "" --add-drivers "nfs" -c /home/urg/nb/fm/etc/dracut.conf --confdir "/home/urg/nb/fm/etc/dracut.conf.d" -v --force '/home/urg/nb/fm/initrd-new' 2> '/home/urg/nb/fm/dracut-debug.log';

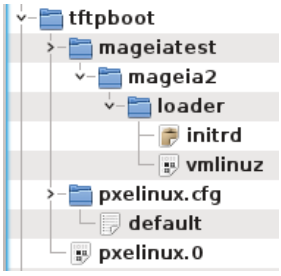
Cette commande génère un magnifique : **/home/urg/nb/fm/initrd-new** qui est votre ramdisque.

4-3-2 Installer le ram disque dans le lanceur du systeme (sommaire)

Vous devrez copier à l'emplacement de votre répertoire tftp : **/tftpboot/mageiatest/loader/** en le renommant en **initrd** ou si vous ne voulez pas le renommer changez la ligne d'appel dans le fichier de configuration : **/tftpboot/pxelinux.cfg/default** comme dans l'exemple ci-dessous.

```
LABEL Mageia-Medintux
KERNEL mageiatest/mageia2/loader/vmlinuz
APPEND ip=dhcp initrd=mageiatest/mageia2/loader/initrd-new root=nfs:192.168.0.10:/home/urg/nb/fm:ro,rsize=32768,wsize=32768,exec,hard,proto=tcp LANG=fr
rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 medintuxRootFs@192.168.0.10:/home/urg/Documents@
```

N'oubliez surtout pas avant de placer le nouveau initrd de renommer l'ancien initrd en initrd-old ou de le ranger quelque part au cas où le nouveau ne fonctionnerait pas.



Un fichier de log est aussi produit : **/home/urg/nb/fm/dracut-debug.log**' si cela ne marche pas lisez le ...

4-3-3 Installer le Kernel dans le lanceur du systeme (sommaire)

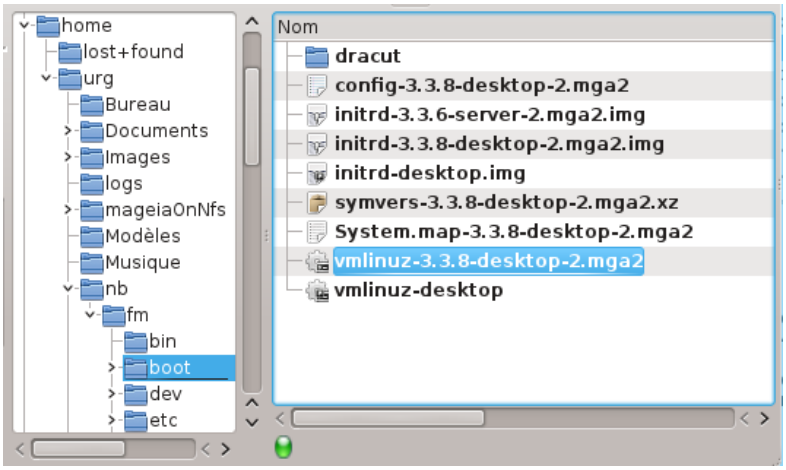
N'oubliez pas non plus de copier le kernel de votre filesystem que vous trouverez dans le répertoire **/boot/** du filesystem nouvellement créé en le renommant **vmlinuz** ou changeant le nom du kernel dans la ligne d'appel du fichier de configuration du meni de démarrage : **/tftpboot/pxelinux.cfg/default**

vmlinuz-3.3.8-desktop-2.mga2 --> **vmlinuz**

```
LABEL Mageia-Medintux
KERNEL mageiatest/mageia2/loader/vmlinuz
```

```
APPEND ip=dhcp initrd=mageiatest/mageia2/loader/initrd-new root=nfs:192.168.0.10:/home/urg/nb/fm:ro,rsize=32768,wsize=32768,exec,hard,proto=tcp LANG=fr
rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 medintuxRootFs@192.168.0.10:/home/urg/Documents@
```

Emplacement du kernel dans notre filesystem créé avec la fabrique



Tout devrait être prêt vous devriez pouvoir démarrer comme vous pouvez le constater ci-dessous.

5 La mise au point du poste client (sommaire)

5-1 Vérification générale avant lancement (sommaire)

Côté serveur vérifier que tout est prët :

5-1-1 NFS (sommaire)

- service NFS : Le file system **home/urg/nb/fm** est exporté par notre **serveur** via **nfs.**
on vérifie ue nfs tourne en par exemple le relançant le service :

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

on vérifie que le répertoire du filesystem est bien exporté, en par exemple le montant le répertoire exporté.

```
mount -t nfs 192.168.0.10:/home/urg/nb/fm /home/urg/tmp/nfs
```

5-1-2 DHCP (sommaire)

- Service dhcp : on le relance pour ëtre sûr qu'il tourne bien en faisant par exemple :

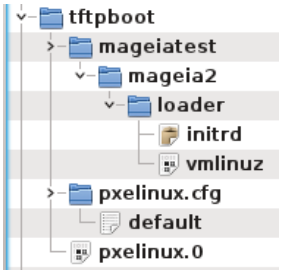
```
killall dhcpd;
dhcpd;
```

vérifions que le fichier de congiration **/etc/dhcp/dhcpd.conf** comporte bien les sections concernant les machines clientes.

```
host machine-1
        { next-server 192.168.0.10;             # le serveur dhcp/tftp
         hardware ethernet 00:14:0B:80:33:06;   # Adresse matérielle du poste client (mac adresse)
         fixed-address 192.168.0.25;            # Adresse IP à lui attribuer
        }
```

5-1-3 TFTP et tftpboot (le répertoire lanceur) (sommaire)

- le répertoire lanceur tftpboot : comporte bien tous les fichiers nécessaires.



on démarre le poste client, et après quelques secondes une mageia est à l'écran. Vous serez probablement en mode console root sans mot de passe, nous allons voir plus loin comment configurer la machine cliente.

5-2 Le poste client en écriture ou lecture seule (sommaire)

Vous avez réussi à booter le client léger mais vous êtes en mode console **root** (pas de mot de passe !! on en mettra un après). et souhaiteriez un affichage graphique : **startx**, bon cela ne démarre pas car votre carte graphique n'a pas le bon fichier **xorg.conf**. Rien n'est perdu, récuperez le bon fichier **xorg.conf**. Comment ? Si vous êtes savant créez le de toutes pièces .... Solution de facilité : démarrez votre client en boutant sur une **Mageia2 live en cd rom**, et copiez le fichier **/etc/X11/xorg.conf** sur une clef USB et placez le dans le filesystem du client léger (donc sur le serveur) au même endroit. CAD dans notre exemple en **/home/urg/nb/fm/etc/X11/xorg.conf** reboutez le client léger, console **root**, et **startx**. Cela devrait démarrer une Mageia sous Kde. Il ya du progrès.

5-2-1 Pourquoi deux modes ? (sommaire)

Le problème que vous allez rencontrer c'est que tous les réglages faits avec les outils de configuration KDE ou le magnifique Mageia Configuration Center seront volatiles et non retrouvés lors du prochain boot, car votre filesystem est en lecture seulement. Trop null.

Ce mode est obligatoire pour que plusieurs clients puissent travailler sans se télescoper dans le filesystem. Pour cela dans ce mode, un mécanisme très spécial appelé tmpfs est mis en place par Mageia. Cela fait notre affaire en fonctionnement normal multi postes, mais pas pour régler les paramètres et configurer le poste de travail.

5-2-2 Mettre le filesystem en mode écriture (sommaire)

Heureusement il est possible de mettre le filesystem en mode écriture, de bouter un client (attention **UN SEUL CLIENT SEULEMENT**) de régler les paramètres, le bureau, créer d'autres comptes utilisateurs, installer d'autres paquets activer la connexion graphique automatique, bref de travailler normalement.

Tout se mettra dans le filesystem, et sera disponible au prochain redémarrage. Une fois les réglages finis, remettez le tout en mode lecture seule, le filesystem sera protégé, plusieurs clients légers peuvent bouter dessus sans problème de téléscopage.

Où cela se passe t-il ? trois modifications à faire.

5-2-2-1 Mettre sur le serveur l'exportation NFS en lecture écriture (sommaire)

Le fichier **/etc/exports** de mon serveur comporte la ligne suivante dont il faut modifier le deuxième paramètre après les parenthèses comme suit :

```
   /home/urg/nb/fm 192.168.0.11/30(fsid=root,ro,no_root_squash,no_subtree_check,async,insecure)
```

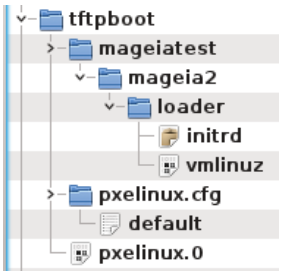
```
   /home/urg/nb/fm 192.168.0.11/30(fsid=root,rw,no_root_squash,no_subtree_check,async,insecure)

relancez le serveur NFS après cette modification.

     /etc/init.d/nfs-server restart    # redémarre le serveur nfs
```

5-2-2-2 Mettre le lanceur en lecture écriture (sommaire)

Pour cela éditez et modifiez le menu du lanceur : **/tftpboot/pxelinux.cfg/default**
changez le paramètre **ro** en **rw** dans la mention indiquant a l'initialisation du kernel l'emplacement du filesystem :
root=nfs:192.168.0.10:/home/urg/nb/fm:ro




```
LABEL Mageia-Medintux
KERNEL mageiatest/mageia2/loader/vmlinuz
APPEND ip=dhcp initrd=mageiatest/mageia2/loader/initrd-new root=nfs:192.168.0.10:/home/urg/nb/fm:ro,rsize=32768,wsize=32768,exec,hard,proto=tcp LANG=fr
rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 medintuxRootFs@192.168.0.10:/home/urg/Documents@
```

```
LABEL Mageia-Medintux
```

```
KERNEL mageiatest/mageia2/loader/vmlinuz
APPEND ip=dhcp initrd=mageiatest/mageia2/loader/initrd-new root=nfs:192.168.0.10:/home/urg/nb/fm:rW,rsize=32768,wsize=32768,exec,hard,proto=tcp LANG=fr
rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 medintuxRootFs@192.168.0.10:/home/urg/Documents@
```

5-2-2-3 Indiquer au filesystem que son accès NFS est en lecture écriture (sommaire)

Pour cela éditez le filesystem le fichier : **/etc/sysconfig/readonly-root** et modifiez le paramètre de la ligne

READONLY=yes

READONLY=no

reboutez le client faites vos réglages, dès qu'il est au point replacez vous en lecture seulement en faisant les trois modifications précédentes inverses.

6 Conclusion (sommaire)

L'économie de temps est fabuleuse en terme de déploiement et maintenance et prix (on trouve des clients légers à 145$) , ma solution fonctionne très bien, il faut noter une certaine lenteur liée aux faibles performances des clients légers mais l'utilisabilité est très correcte et très suffisante pour l'applicatif MedinTux (pas de 3D temps réel).
Il reste à régler le problème de la configuration automatique de X11. Là il me faut des gourous. Mais si ils tardent je trouverais...

Merci encore à Mr Colin Guthrie, et à ceux qui m'ont aiguillés dans ce travail.
Une archive contenant tous les trucs et machins que j'ai exposés plus haut est disponible sur le site de medintux.

Comme vous l'avez remarqué la mise en page est sommaire, pour ceux à qui ont mauvaise vue, je l'ai rappelé très souvant dans ce document.
(sommaire) .