
Administration de clients diskless: Un cas d'école

Partie II: Aspects Techniques

Marwan Burelle - Pierre Letouzey

22 mars 2003

Les 3 étapes du Boot du client diskless:

- Phase I: trouver un noyau Linux
- Phase II: trouver des fichiers
- Phase III: lancer l'environnement graphique

Phase I:
Récupérer un noyau Linux

Quel noyau, et comment l'utiliser ?

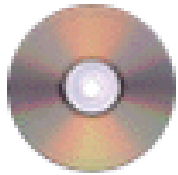
- Le même noyau pour tous les clients.
 - ⇒ besoin de contenir tous les drivers (réseau...)
- Problème 1: acheminer ce noyau au client
- Problème 2: au démarrage, ce noyau doit déterminer:
 - sa propre configuration réseau (n° IP)
 - qui fournit les fichiers (Serveur NFS)
 - qui relaie vers Internet (Passerelle)

Acheminer le noyau jusqu'au client

Possibilités:



⇒ L'ancienne solution. Pénible ...



⇒ Peu de clients en ont ...

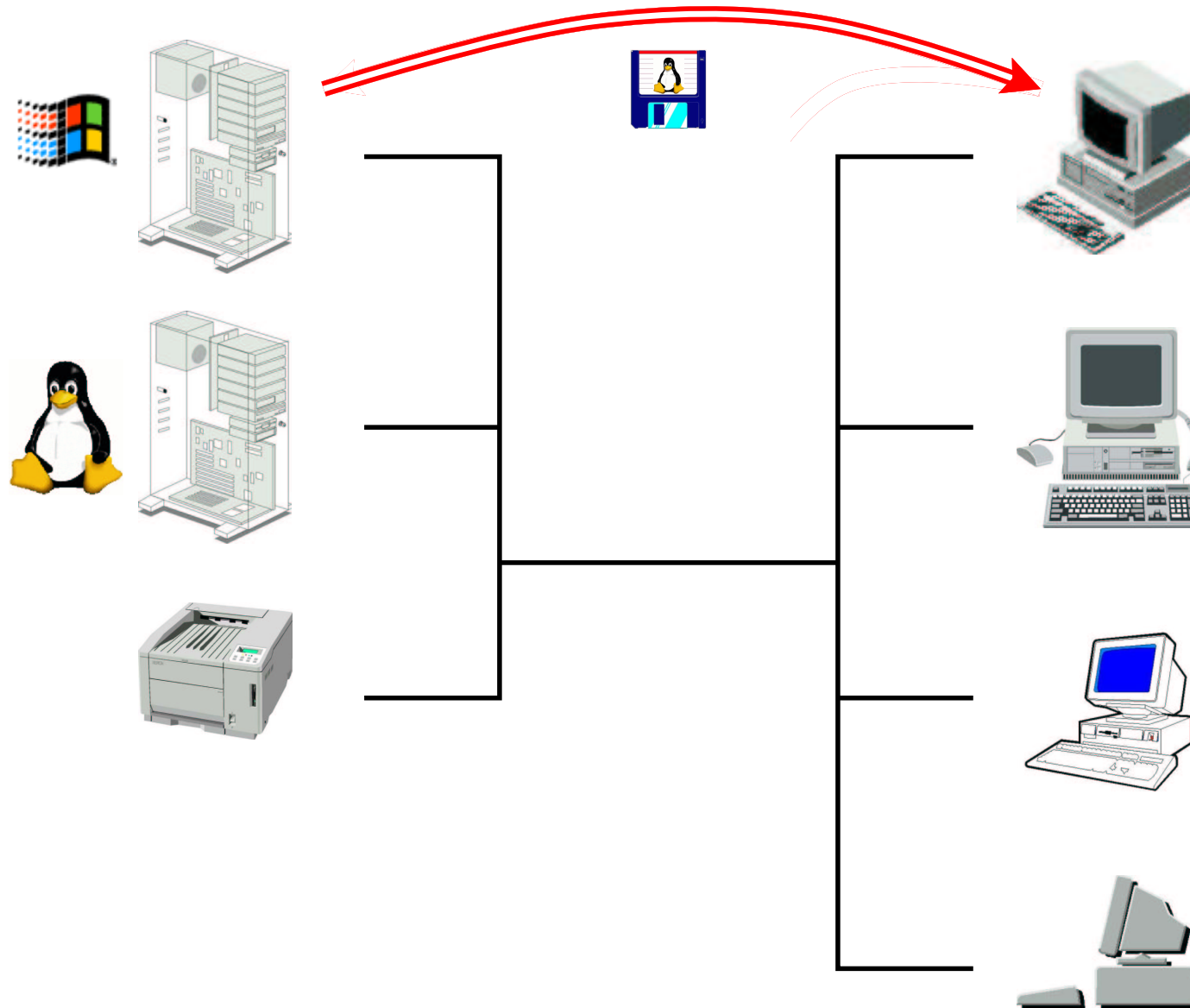


⇒ sauf que normalement le réseau nécessite des drivers, et donc le noyau ...

Solution: **Pre-boot eXecution Environment (PXE)**.

Une carte réseau “intelligente” qui prend la main après le BIOS et qui dialogue avec un serveur PXE.

L'échange PXE



Les paramètres du noyau

On utilise GRUB comme chargeur de noyau.

Il sert à fabriquer la bonne ligne de commande:

```
default 0
```

```
timeout 0
```

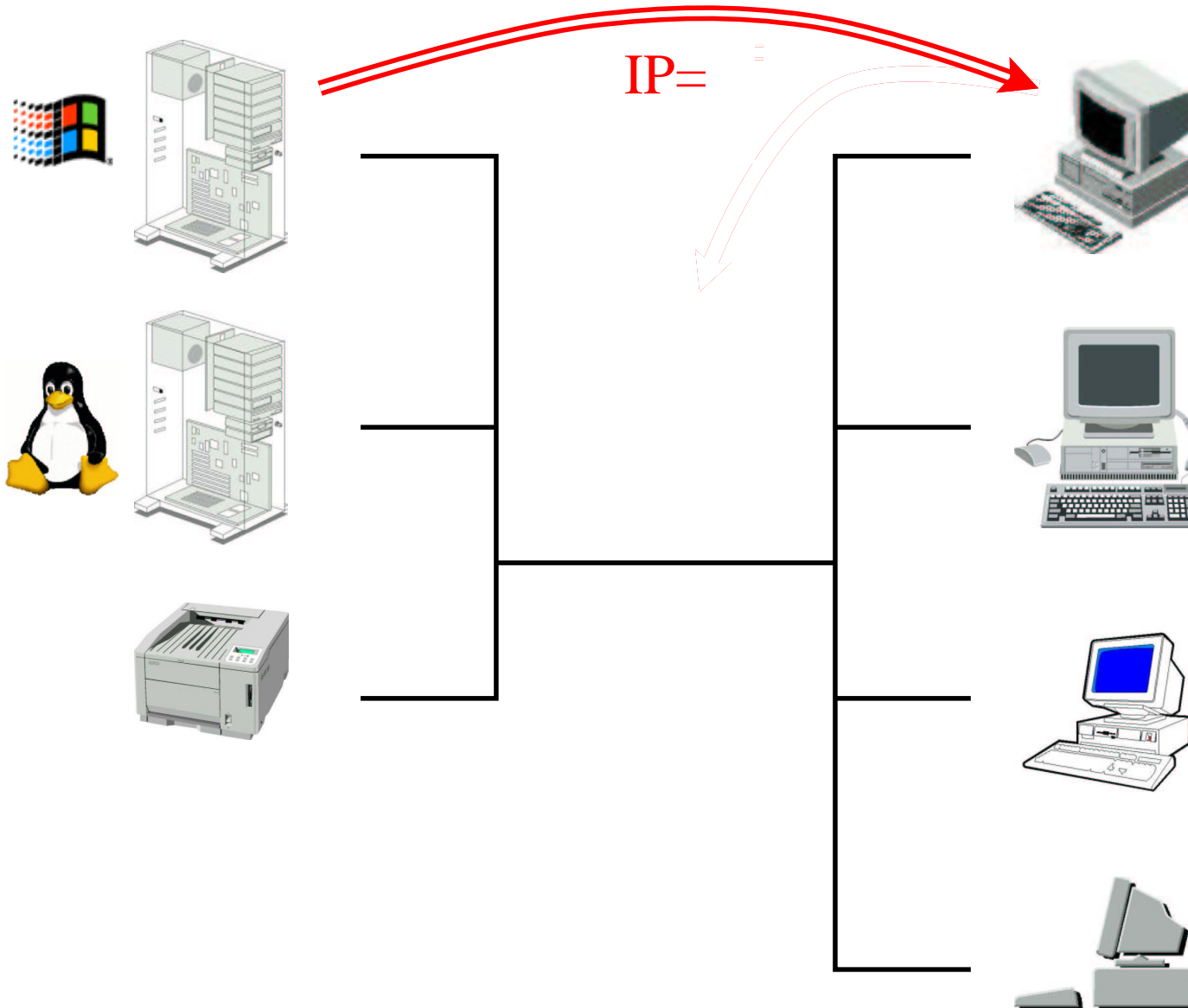
```
title Linux 336
```

```
dhcp
```

```
ifconfig --server=192.168.188.8 --gateway=192.168.188.1
```

```
kernel --keep-ip vmlinuz root=nfs nfsroot=/tftpboot
```

L'échange DHCP



Phase II:
Reconstituer une arborescence de fichiers (FS)

Noyau et FS

Le noyau s'exécute, charge ses drivers, puis passe la main à `init`, 1^{er} accès au FS.

Au départ / du client = / `tftpboot` du serveur.

Famélique: `init`, `mount`, `sh`, `rcS` ... + `libc`.

Protocole: NFS = disque dur à distance via réseau.

Particularité: / `read-only`, car le même pour tous.

`init` lance le 1^{er} script de config, `rcS` qui va maintenant jouer au légo pour construire un FS complet

Buts et contraintes pour le FS

- idéalement, une copie de l'arborescence du serveur Linux
- mais avec des configurations propres aux clients
- on ne peut pas exporter tout / du serveur
- on ne peut pas faire non plus 130 copies du serveur

Solutions pour le FS

- les parties immuables (`/usr, ...`) \Rightarrow NFS `read-only`
- les parties variables mais communes (`/home, ...`) \Rightarrow NFS `read-write`
- les fichiers de configuration `/etc`: \Rightarrow NFS “différentiel”
à base de liens symboliques.
- enfin les parties variables par machine: FS “virtuels”
2 cas à distinguer:
 - les devices de `/dev` \Rightarrow DevFS spécialisé
 - les locks, logs ... de `/tmp` & `/var` \Rightarrow TmpFS
(= ramdisk adaptatif)

Propriétés d'un tel montage

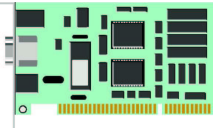
- un client ne peut modifier les fichiers systèmes du serveur
- un client voit ses particularismes (locks,logs,...) perdus à chaque reboot (RAM volatile). Un nettoyage radical!
- la différenciation entre les clients se fait par des scripts d'auto-détection.
- un utilisateur retrouve ses fichiers depuis n'importe quel poste.

Phase III:

Lancement de l'environnement graphique (X)

Une gageure: l'auto-configuration de X

Une hétérogénéité importante, mais non extrême:



4 types de cartes vidéos, bien supportées (Xfree 4.3)



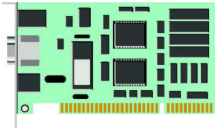
PS/2 ou séries



Plats ou cathodiques

Une gageure: l'auto-configuration de X

Une hétérogénéité importante, mais non extrême:



Détection simple: `lspci` ou `discover`



Chance: PS/2 \pm PnP et un seul autre cas. `mdetect`



Protocole DDC

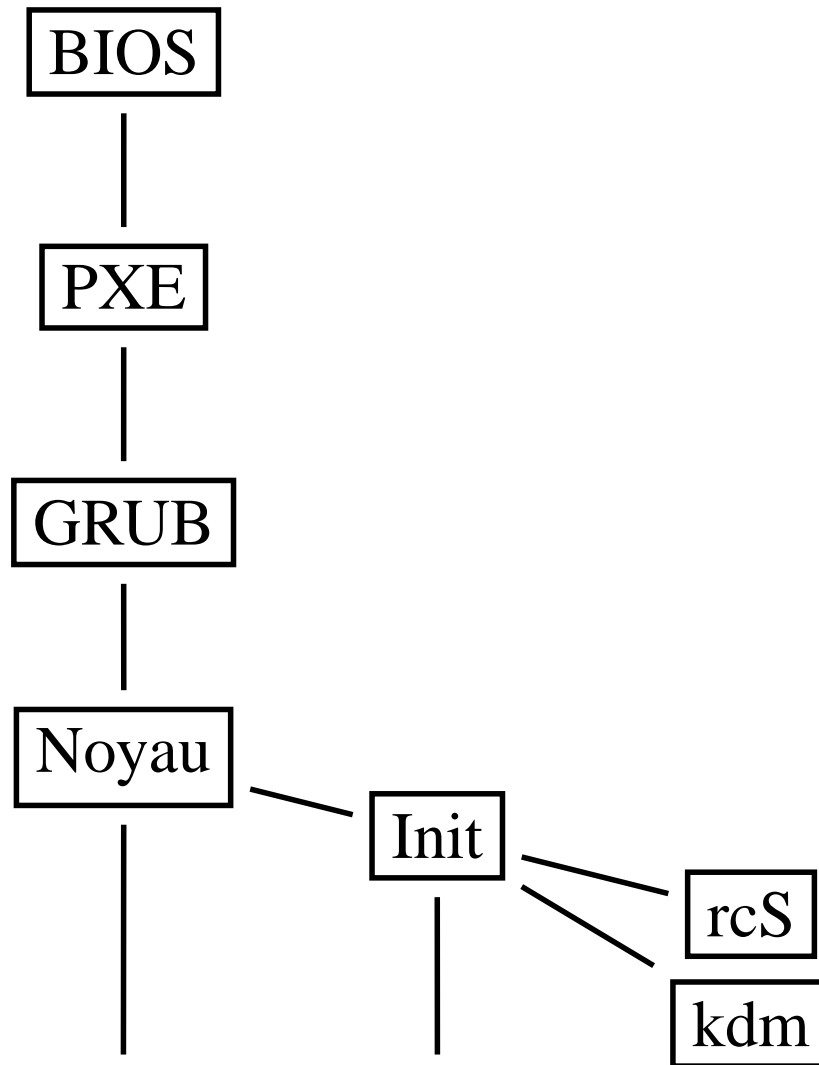
Au final, un script maison `xdetect` dit par exemple `i810.ps2`.

Et **un seul** fichier de configuration `XF86Config-4`, qui contient autant de sections `ServerLayout` que de cas.

Enfin, X est lancé avec l'option `-layout `xdetect``

Conclusion

Récapitulatif du Boot



Bilan matériel

Ca tourne bien!

Points critiques:

1. réseau (switch 100 Mb/s)
2. disque dur du serveur (RAID 5)
3. RAM des clients (256 Mo)

Bilan Linux

Très instructif à gérer.

Savant mélange de techniques anciennes:

- NFS = 1987
- principe du diskless: plus vieux encore...

Et de techniques jeunes/expérimentales/non-standards:

- PXE
- Grub
- DevFS
- TmpFS
- auto-config X ...

Logiciels Libres en Vrac

linux+gnu grub lilo debian dhcpd
rarpd tftpd nfsd apache ssh iptables
exim mars-nwe cups xfree86
netscape fvwm2 kde emacs ocaml
perl gimp blender gpc dosemu latex
hevea advi cvs rsync samba nessus
nmap ethereal