

VServer

Cet tutoriel est basé sur [le travail](#) de Michael HARLAUT et présente la mise en place de VServers sous la distribution Debian Sarge. Une compilation du noyau Linux sera nécessaire.

VServer est un patch pour le noyau Linux 2.4/2.6 et une série d'outils qui permettent de mettre en place facilement des environnements d'exécution dans des contextes séparés, en isolant les processus, les systèmes de fichiers, les ressources réseau, et tout ce qu'offre le noyau.

On peut ainsi lancer plusieurs systèmes Linux sur une même machine, pour obtenir autant d'espaces de travail souhaités en partageant le même noyau Linux entre tous les VServers.

Sur les processus: Chaque numéro de processus est préfixé par un numéro de contexte.

Sur le système de fichier: Chaque environnement tourne sur un système qu'il peut utiliser seul.

Sur le réseau: Chaque environnement ne peut utiliser que l'adresse réseau qui lui est attribuée.

Chaque VServer est juste constitué d'une arborescence et d'un fichier de configuration. Cela veut dire que l'on peut librement déplacer un VServer simplement en recopiant ses fichiers!

Chaque environnement peut se voir associer des droits particuliers sur le système.

On peut limiter les ressources allouées à chaque VServer (processus, mémoire, quota disque, ...).

On peut relancer ou faire vivre un VServer sans toucher les autres.

VServer ne consomme pas de ressources. Les seules ressources consommées le sont par les processus en cours d'exécution (les daemons ...).

Il existe d'autres projets de virtualisation Open Source. Xen est le plus avancé, il devrait être intégré au kernel [prochainement](#). Il fera sûrement l'objet d'un nouveau tutoriel quand je l'aurai testé...

Installation des packages

 **debian**

kernel-package_8.130_all.deb
util-vserver_0.30.2045_i386.deb
vserver-debiantools_0.1.10_all.deb

Compilation graphique:
libqt3-headers_3.3.4-3_i386.deb
libqt3c102-mt_3.3.4-3_i386.deb

Compilation du noyau Linux

Installez Sarge sur votre machine, que l'on appellera la machine hôte ou physique, avec un kernel 2.4.x si vous voulez installer un VServer de la branche 1.x (déprécié maintenant) ou un kernel 2.6.x pour la branche 2.x (plus de [fonctionnalités](#)).

Vous devez télécharger la dernière version stable du patch VServer sur [le site du projet](#) (lien pour un [kernel 2.4](#) ou [kernel 2.6](#)).

En fonction de la version proposée vous pourrez télécharger sur <http://www.kernel.org/> le kernel correspondant.

Pour plus de simplicité, utilisez un autre poste Sarge avec un environnement graphique pour compiler le kernel. Placez le patch VServer et la source du kernel dans **/usr/src** puis décompressez le tout. Faites une copie du patch dans le répertoire source de votre kernel puis passez ces commandes:

Compilation d'un kernel 2.4:

```
hote:~# cd /usr/src/
hote:/usr/src# tar xjf linux-2.4.30.tar.bz2
hote:/usr/src# bunzip2 patch-2.4.30-vs1.2.10.diff.bz2
hote:/usr/src# ll
drwxr-xr-x 15 573 573 4096      linux-2.4.30
-rw-r--r--  1 root src 31136728 linux-2.4.30.tar.bz2
-rw-r--r--  1 root src 154007   patch-2.4.30-vs1.2.10.diff
-rw-r--r--  1 root src 31484    patch-2.4.30-vs1.2.10.diff.bz2

hote:/usr/src# cp patch-2.4.30-vs1.2.10.diff linux-2.4.30/
hote:/usr/src# cd linux-2.4.30/
```

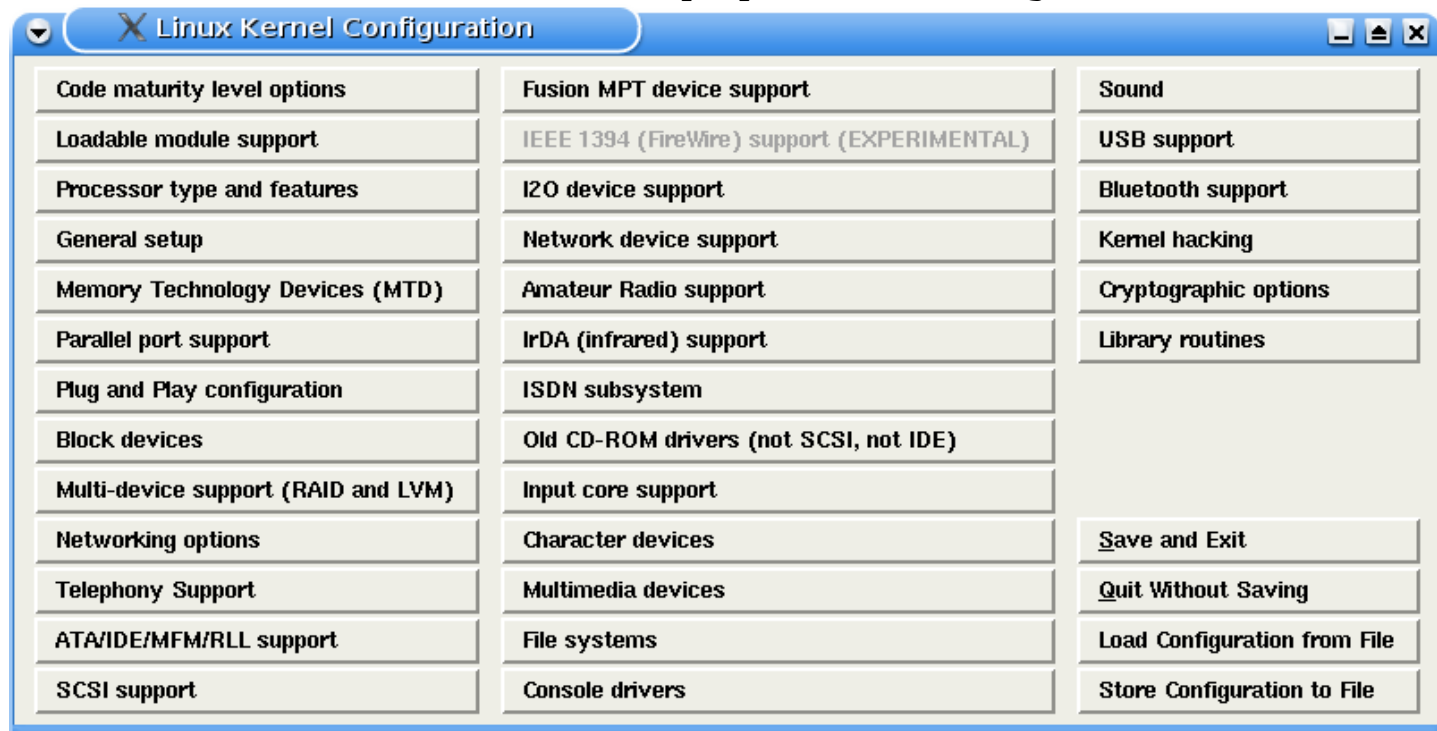
On patche le kernel.

```
hote:/usr/src/linux-2.4.30# patch -p1 < patch-2.4.30-vs1.2.10.diff
patching file Documentation/Configure.help
patching file Makefile
...
patching file net/unix/af_unix.c
```

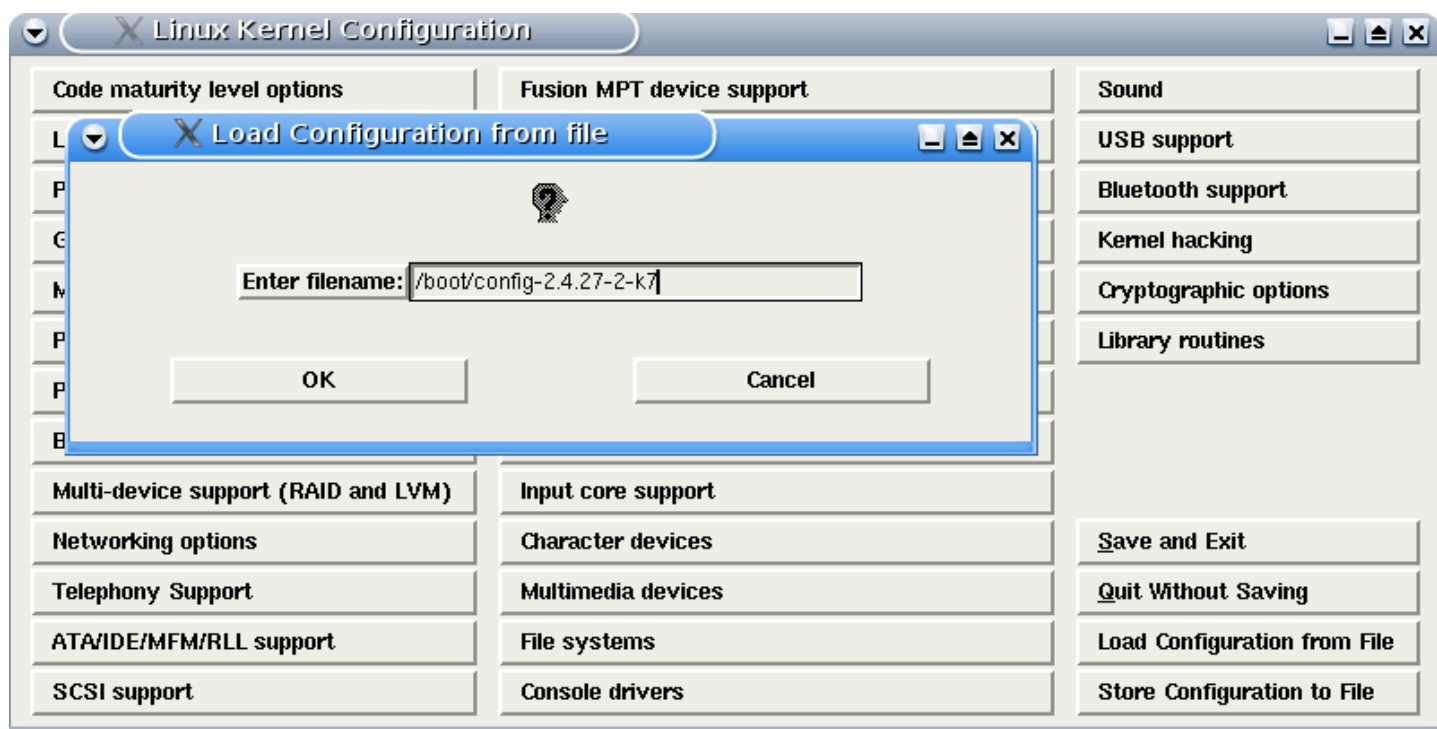
Sous le compte utilisateur avec lequel vous avez ouvert une session X, autorisez root à faire de même.

```
user@hote:~$ xhost +
access control disabled, clients can connect from any host
```

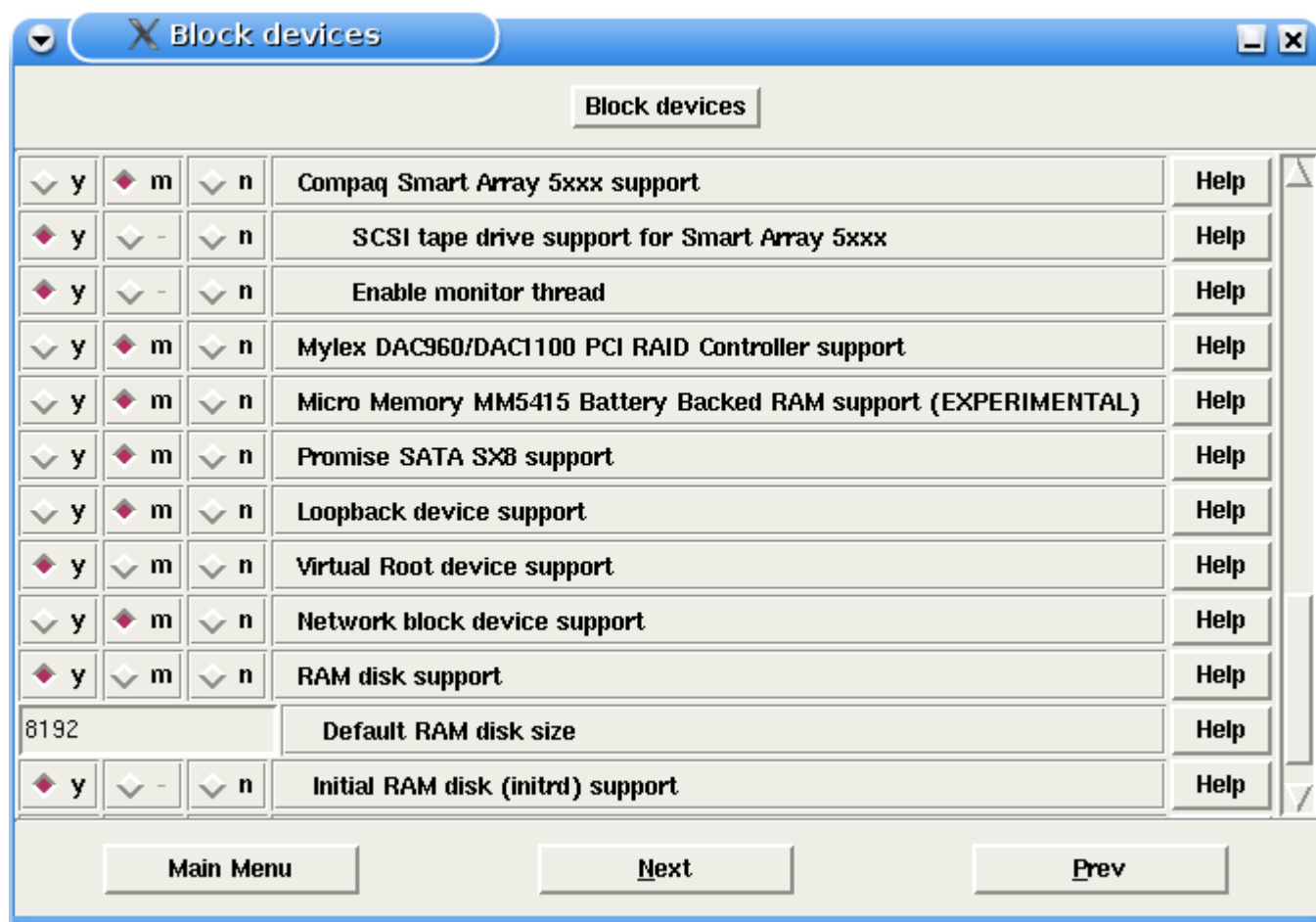
```
hote:/usr/src/linux-2.4.30# make mrproper;make xconfig
```



Chargez le fichier config de votre kernel Sarge (le plus adapté à votre architecture) "Load Configuration from File"



Dans la section "Block Devices" activez l'option "Virtual root device support"



Sauvegardez "Save and Exit" puis lancez la compilation.

note: /usr/src/linux-2.4.30# make dep; make bzImage

Vérifiez la présence du fichier arch/i386/boot/bzImage avant de continuer.

S'il est absent, effacez le répertoire source de votre kernel et recommencez les étapes précédentes en vérifiant vos

paramètres dans le menu "xconfig".

Compilation des modules et installation.

```
hote:/usr/src/linux-2.4.30# make modules;make modules_install
```

Copiez le kernel et le reste.

```
hote:/usr/src/linux-2.4.30# cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.30-vs1.2.10
```

```
hote:/usr/src/linux-2.4.30# cp System.map /boot/System.map-2.4.30-vs1.2.10
```

```
hote:/usr/src/linux-2.4.30# cp .config /boot/config-2.4.30-vs1.2.10
```

Créez l'image minimal chargée en RAM au boot, **elle dépend de votre quantité de RAM**

```
hote:/usr/src/linux-2.4.30# cd /boot/
```

```
hote:/boot# mkinitrd -o /boot/initrd.img-2.4.30-vs1.2.10 2.4.30-vs1.2.10
```

Modifiez votre chargeur grub et rebootez.

```
hote:/boot# vi grub/menu.lst
```

```
...
title          Debian GNU/Linux, kernel Vserver 2.4.30-vs1.2.10
root           (hd0,0)
kernel         /boot/vmlinuz-2.4.30-vs1.2.10 root=/dev/hda1 ro
initrd         /boot/initrd.img-2.4.30-vs1.2.10
savedefault
boot
...
```

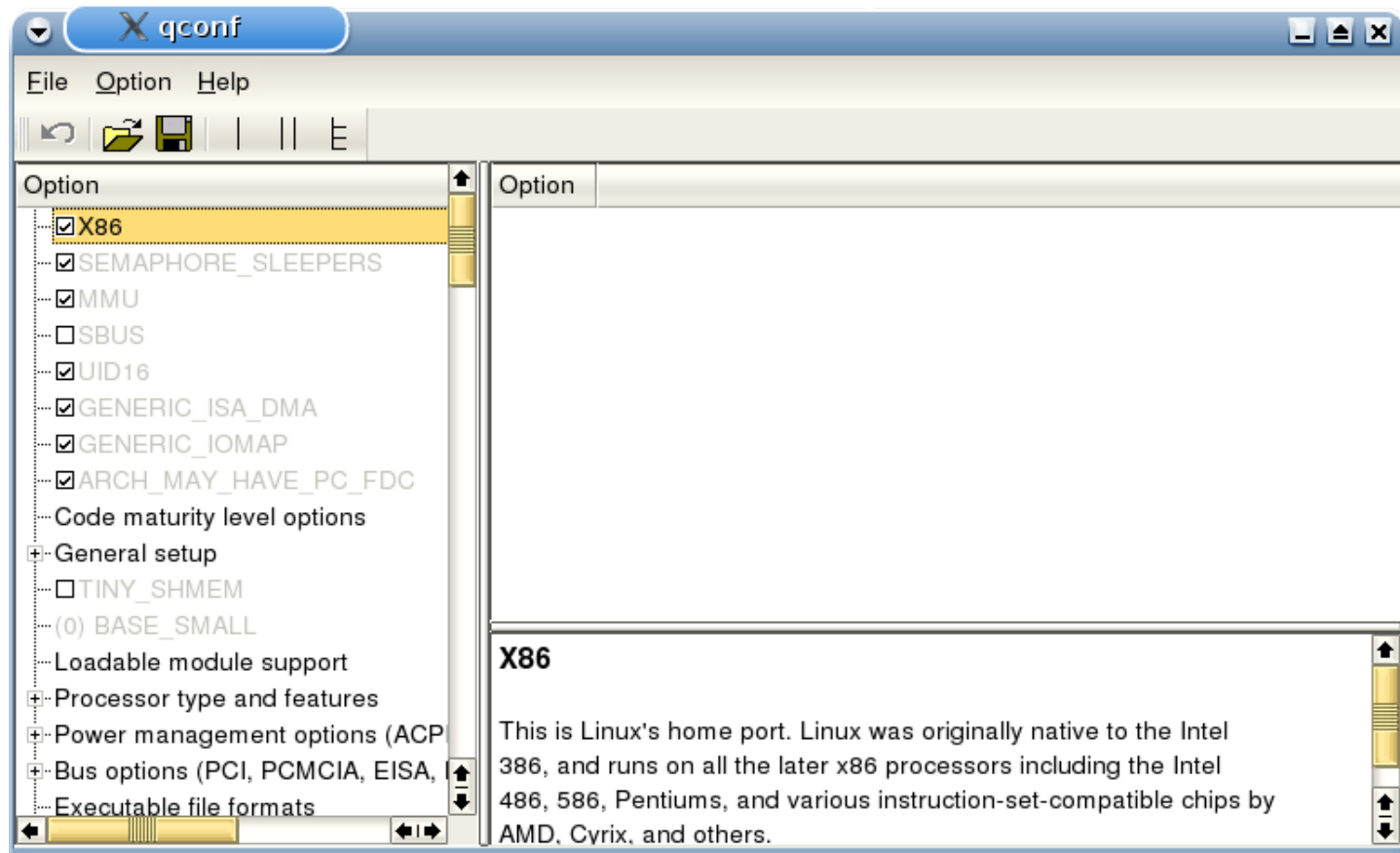
Compilation d'un kernel 2.6:

```
hote:~# cd /usr/src/  
hote:/usr/src# tar xjf linux-2.6.14.3.tar.bz2  
hote:/usr/src# bunzip2 patch-2.6.14.3-vs2.01.diff.bz2  
hote:/usr/src# cp patch-2.6.14.3-vs2.01.diff linux-2.6.14.3/  
hote:/usr/src# cd linux-2.6.14.3/
```

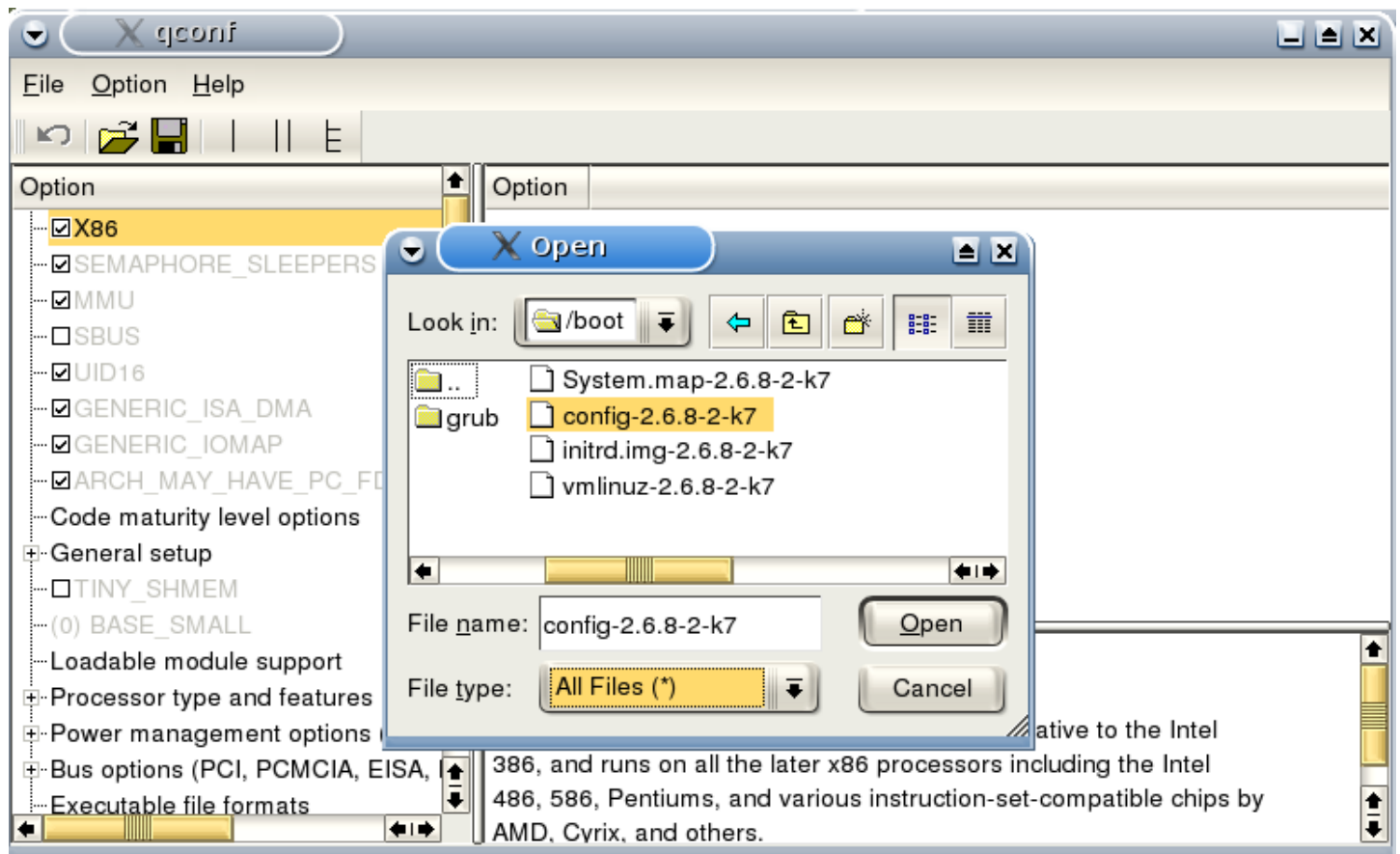
On patche le kernel.

```
hote:/usr/src/linux-2.6.14.3# patch -p1 < patch-2.6.14.3-vs2.01.diff
```

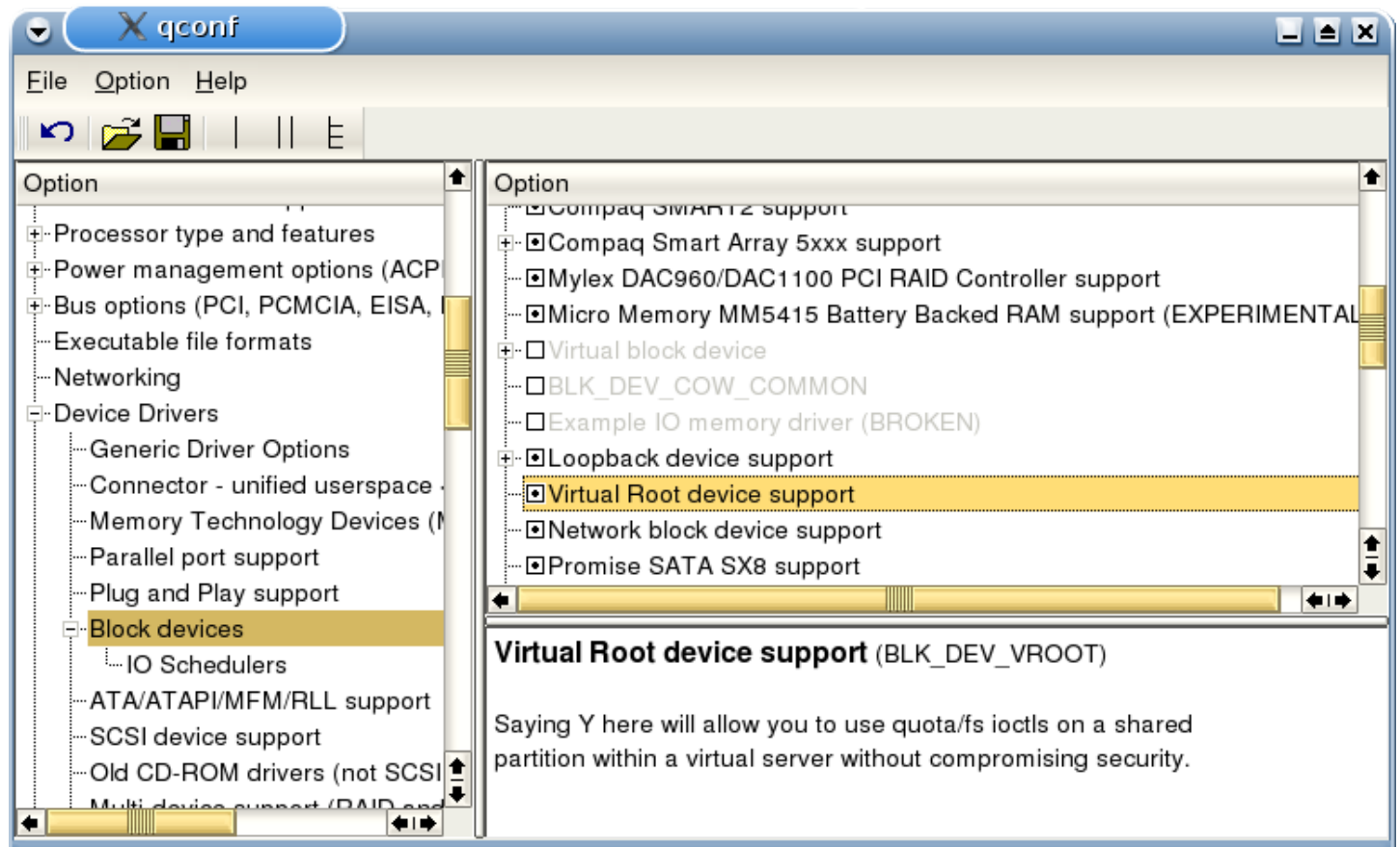
```
hote:/usr/src/linux-2.6.14.3# make mrproper;make xconfig
```



Chargez le fichier config de votre kernel Sarge (le plus adapté à votre architecture)



Dans la section "Block devices" activez l'option "Virtual root device support"



Sauvegardez "Save and Quit" puis lancez la compilation.

note: /usr/src/linux-2.6.14.3# make

Vérifiez la présence du fichier `arch/i386/boot/bzImage` avant de continuer.

```
hote:/usr/src/linux-2.6.14.3# make modules_install
```

```
...
INSTALL sound/synth/snd-util-mem.ko
INSTALL sound/usb/snd-usb-audio.ko
INSTALL sound/usb/snd-usb-lib.ko
if [ -r System.map -a -x /sbin/depmod ]; then /sbin/depmod -ae -F
System.map 2.6.14.3-vs2.0.1; fi
```

Copiez le kernal et le reste.

```
hote:/usr/src/linux-2.6.14.3# cp arch/i386/boot/bzImage /boot/vmlinuz-
2.6.14.3-k7-vs2.0.1
hote:/usr/src/linux-2.6.14.3# cp System.map /boot/System.map-2.6.14.3-k7-
vs2.0.1
hote:/usr/src/linux-2.6.14.3# cp .config /boot/config-2.6.14.3-k7-vs2.0.1
```

Créez l'image minimal chargée en RAM au boot, **elle dépend de votre quantité de RAM**

```
hote:/usr/src/linux-2.6.14.3# cd /boot/
hote:/boot# mkinitrd -o /boot/initrd.img-2.6.14.3-k7-vs2.0.1 2.6.14.3-
vs2.0.1
```

Modifiez votre chargeur grub et rebootez.

Installation et configuration de VServer

Préparation de l'hôte

Dans un premier temps, configurer les démonstrations sur l'hôte pour qu'ils écoutent uniquement leur adresse IP.

Remplacer `inetd` par `xinetd` (`apt-get install xinetd`) ajouter cette ligne dans **/etc/xinetd.conf** (avec l'IP de votre serveur hôte):

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{ bind = 192.168.0.1 }

includedir /etc/xinetd.d
```

Si vous avez `sshd`: Dans **/etc/ssh/sshd_config**

```
...
ListenAddress 192.168.0.1
...
```

Création du 1^{er} VServer

Vous pouvez choisir plusieurs paramètres dont la version Debian (woody, sarge, sid...)

La syntaxe est la suivante:

```
hote:~# newvserver --vsroot /var/lib/vservers --hostname nom_du_vserver --
domain mon_de_votre_domaine --ip ip_du_vserver --interface eth0 --arch
i386 --dist version_Debian --mirror ftp://un_mirrored/debian
```

Le nom d'une interface réseau est limité à 15 caractères, y compris le "ethX:"

Pour l'installation d'un VServer Woody alors que mon hôte est en Sarge:

```
hote:~# newvserver --vsroot /var/lib/vservers --hostname vserver1 --domain
alex.fr --ip 192.168.0.11 --interface eth0 --arch i386 --dist woody --
mirror ftp://ftp2.fr.debian.org/debian
s_context: 0
I: Retrieving debootstrap.invalid_dists_woody_Release
I: Validating debootstrap.invalid_dists_woody_Release
I: Retrieving debootstrap.invalid_dists_woody_main_binary-i386_Packages
I: Validating debootstrap.invalid_dists_woody_main_binary-i386_Packages
I: Checking adduser...
I: Checking apt...
...
WARNING: can not find configuration, assuming legacy method
ipv4root is now 192.168.0.11
New security context is 49152
Stopping periodic command scheduler: cron.
Stopping internet superserver: inetd.
Stopping OpenBSD Secure Shell server: sshd.
Stopping deferred execution scheduler: atd.
Stopping system log daemon: syslogd.
```



```
Sending all processes the TERM signal... done.
Sending all processes the KILL signal... done.
Unmounting remote filesystems... done.
WARNING: can not find configuration, assuming legacy method
Stopping the virtual server vserver1
Server vserver1 is not running
Restarting rebootmgr.
Stopping the reboot manager
Starting the reboot manager
```

You should now adjust `/etc/vservers/vserver1.conf` to suit your needs, or else just go ahead and type ``vserver vserver1 start'` to start your new virtual server. `debian/rules!`

hote:~#

Laissez se dérouler l'installation de votre VServer (création d'une arborescence et choix des paquets à installer dans votre VServer)

Ajustez le fichier de configuration de votre VServer `/etc/vservers/vserver1.conf`

```
# Le nom du vserver.
S_HOSTNAME="vserver1"
# Sur quelle interface l'allias IP sera créé.
IPROOTDEV="eth0"
# IP du vserver (IPROOT écrase l'option IPROOTDEV). Pour une passerelle :
# IPROOT="eth0:192.168.0.11/255.255.255.0
eth1:172.16.141.12/255.255.255.128"
#
# Pour avoir aussi une boucle locale. IPROOT="192.168.0.11 127.0.0.1"
IPROOT="192.168.0.11"
# Le masque et broadcast, si on ne la pas définit dans IPROOT.
IPROOTMASK=255.255.255.0
IPROOTBCAST="192.168.0.255"
# Démarrage ou non de ce vserver au lancement du daemon vserver.
ONBOOT="no"
# Modifie le niveau de priorité pour tous les processus du vserver.
# de -20 (très haut) à 19 (très bas)
S_NICE=""
# Régle les états du contexte de sécurité.
# lock: Empêche le vserver d'être placé dans un nouveau contexte.
# nproc: Limite le nombre de processus dans le vserver selon l'ulimit
# (au lieu d'une limite par-utilisateur, ce devient une limite par-
vserver)
S_FLAGS="lock nproc"
# Limitation à 1024 processus pour l'ensemble du vserver.
ULIMIT="-H -n 1024"
# Capacités à donner au vserver (le moins possible).
# On donnera par exemple CAP_SYS_RESOURCE à un vserver hébergeant
# un serveur DNS Bind9. (voir Vue d'ensemble des capacités et ce site)
# S_CAPS="CAP_NET_RAW CAP_SYS_RESOURCE"
S_CAPS="CAP_NET_RAW"

# *NOT* DNS domain name, for NIS only
S_DOMAINNAME=""
```

Vue d'ensemble des capacités

La liste des capacités POSIX utilisées avec Linux est longue. Bien que la liste détaillée de ces capacités puisse être trouvée dans `/usr/include/linux/capability.h` sur la plupart des systèmes Linux, une vue d'ensemble des capacités importantes est [donnée sur ce site](#).

CAP_CHOWN	Change le propriétaire et le groupe du fichier
CAP_KILL	Envoie un signal au processus avec un ID utilisateur différent, réel ou effectif
CAP_SETGID	Permet le passage de <code>setgid</code> , <code>setgroups</code> , et de gids forgés sur les sockets
CAP_SETUID	Permet le passage de <code>set*uid</code> et d'uids forgés sur les sockets
CAP_SETPCAP	Transfert/déplace toute capacité de l'ensemble permis vers/depuis n'importe quel pid
CAP_LINUX_IMMUTABLE	Permet la modification des attributs de fichier <code>S_IMMUTABLE</code> et <code>S_APPEND</code>
CAP_NET_BROADCAST	Permet la diffusion et l'écoute en multicast
CAP_NET_ADMIN	Permet la configuration de l'interface, le pare-feu IP, le masquering, l'accounting, le débogage de sockets, les tables de routage, la liaison sur n'importe quelle adresse, l'entrée dans le mode de promiscuité, le multicasting...
CAP_NET_RAW	Permet l'utilisation des sockets RAW et PACKET
CAP_SYS_MODULE	Ajoute et retire des modules noyau
CAP_SYS_CHROOT	Permet <code>chroot</code>
CAP_SYS_PTRACE	Permet <code>ptrace()</code> sur n'importe quel processus
CAP_SYS_ADMIN	La liste serait trop longue, ceci permet en gros de faire tout ce qui n'est pas mentionné dans les autres capacités
CAP_SYS_BOOT	Permet <code>reboot</code>
CAP_SYS_NICE	Permet l'élévation des privilèges et la modification de la priorité des autres processus, en modifiant l'ordonnanceur
CAP_SYS_RESOURCE	Passe outre les limites de ressource, de quota, l'espace réservé sur un système de fichier...
CAP_MKNOD	Permet les aspects privilégiés de <code>mknod</code>

Gestion des VServers

Démarrer votre VServer:

```
hote:~# vserver vserver1 start
WARNING: can not find configuration, assuming legacy method
Starting the virtual server vserver1
Server vserver1 is not running
ipv4root is now 192.168.0.11
Host name is now vserver1
New security context is 49153
Starting system log daemon: syslogd.
Starting internet superserver: inetd.
Starting OpenBSD Secure Shell server: sshd.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: cron.
hote:~#
```

Entrez dans votre VServer:

```
hote:~# vserver vserver1 enter
WARNING: can not find configuration, assuming legacy method
ipv4root is now 192.168.0.11
New security context is 49153
vserver1:/# exit
logout
hote:~#
```

N'oubliez pas de configurer les services de votre VServer pour qu'ils écoutent sur leur propre interface (sshd,...) voir [Configuration spécifique des VServers](#)

Visualiser le statut d'un VServer depuis l'hôte:

```
hote:~# vserver vserver1 status
WARNING: can not find configuration, assuming legacy method
Server vserver1 is running
5 processes running
Vserver uptime: 00:01
hote:~#
```

Visualiser le statut des VServers depuis l'hôte:

```
hote:~# vserver-stat
CTX   PROC  VSZ   RSS   userTIME  sysTIME  UPTIME   NAME
0      97    1.8G 208K  2m01s52   0m21s47  34m49s49 root server
49153 5      8.2M 910   0m00s20   0m00s00  1m45s58  [vserver1]
hote:~#
```

D'autres informations depuis l'hôte:

```
hote:~# vserver-info
hote:~# vtop
hote:~# vps
hote:~# vps tree
hote:~# vkill
```

Dupliquer proprement un VServer (arrêtez-le avant):

```
hote:~# dupvserver --vsroot /var/lib/vservers --from vserver1 --to
vserver10 --ip 192.168.0.10
Changed the following files:
/var/lib/vservers/vserver10/etc/exim/exim.conf
```

```
/var/lib/vservers/vserver10/etc/hostname
/var/lib/vservers/vserver10/etc/hosts
/var/lib/vservers/vserver10/etc/mailname
/var/lib/vservers/vserver10/etc/motd
/var/lib/vservers/vserver10/etc/ssh/sshd_config
```

Le nom du VServer et l'adresse IP seront modifiés dans tout les fichiers nécessaire.

Partager un répertoire entre plusieurs VServers:

```
hote:~# mount --bind /home /var/lib/vservers/vserver1/home
hote:~# mount --bind /home /var/lib/vservers/vserver2/home
```

Exécuter des tâches sur un VServer:

Créer un fichier `/etc/vservers/hom_du_vserver.sh` pour chaque VServer.

Ce fichier sera appelé avant et après le démarrage, avant et après l'arrêt d'un VServer, avec les arguments respectifs pre-start, post-start, pre-stop et post-stop.

```
#!/bin/sh
#/etc/vservers/vserver1.sh
case $1 in
pre-start)
    mount --bind /home /var/lib/vservers/$2/home
    ;;
post-start)
    ;;
pre-stop)
    ;;
post-stop)
    umount /var/lib/vservers/$2/home
    ;;
esac
```

Configuration spécifiques des VServers

X forwarding et ssh

Editer le fichier `/etc/ssh/sshd_conf`:

```
...
ListenAddress 192.168.0.11
...
X11Forwarding yes
X11UseLocalhost no
...
```

Session X

Il faut ajouter au VServer la capacité `CAP_SYS_RAWIO` (dans la variable `S_CAPS` du fichier de configuration de VServer)

Sous Debian, installer `x-window-system-core` et `xterm`

Mysql

Dans `/etc/mysql/my.cnf`:

```
...
bind-address = 192.168.0.11
...
```

vsFtpd

Options à modifier dans /etc/vsftpd.conf:

```
...  
listen_address=192.168.0.11  
...
```

PostgreSQL

Options à modifier dans /etc/postgresql/postgresql.conf:

```
...  
virtual_host = '192.168.0.11'  
...
```

Voir les autres problèmes sur [cette page](#).

Sources: http://intranet.harlaut.net/doku/mise_en_place_de_vserver
<http://www.solucorp.qc.ca/howto.hc?projet=vserver>
<http://linux-vserver.org/WorkingPrograms>

Document mis à jour: 22/01/06