

Sommaire

<i>OpenVPN dans un domU.....</i>	<i>2</i>
<i>Additionner des disques avec mhdfs.....</i>	<i>2</i>
<i>Convertir des chm en pdf.....</i>	<i>4</i>
<i>divfix++ le réparateur de video.....</i>	<i>5</i>
<i>Xen et le message 4gb seg fixup, process.....</i>	<i>6</i>
<i>Installation d'une clé wifi USB SAGEM sur une Debian.....</i>	<i>6</i>
<i>Grub et résolution du tty.....</i>	<i>7</i>
<i>Dsh - Shell ditribué.....</i>	<i>8</i>

OpenVPN dans un domU

publié par macsim le mardi, juillet 15 2008 - 14:04

Je n'ai pas d'interface tunX dans un domU debian, pour lancer openvpn j'ajoute dans /etc/init.d/openvpn

```
mkdir /dev/net  
  
mknod /dev/net/tun c 10 200
```

Si vous connaissez mieux.

Additionner des disques avec mhdfs

publié par macsim le dimanche, juillet 13 2008 - 18:27



Attention j'ai testé ce logiciel sur debian testing dans une machine virtuelle.

[Mhdfs](#) est un logiciel qui permet d'additionner plusieurs disques ensemble afin d'en faire un seul virtuel, on parle bien ici d'addition ce n'est donc pas du [RAID0](#), Lorsque l'on copie un fichier en [RAID0](#) la moitié du fichier est sur un disque et le reste sur un autre (Vision Basique). Avec mhdfs on copie un fichier il est mis complètement dans un des disques qui forme la partition virtuelle.

Pour tester tout ça j'ai donc fait une machine virtuelle avec une [Debian](#) testing, un disque dur de 4G (pour installer la debian) et deux disques durs de 1G pour faire les tests.

Pour l'installation on reste sur le classique:

```
#aptitude install mhdfs
```

Ensuite voici un **df -h** du système de fichier avant la fusion.

```
debtest:~# df -h
Sys. de fich.      Tail. Occ. Disp. %Occ. Monté sur
/dev/sda1          3,8G 672M 2,9G 19% /
tmpfs              126M 0 126M 0% /lib/init/rw
udev              10M 100K 10M 1% /dev
tmpfs              126M 0 126M 0% /dev/shm
/dev/sdb1          1004M 18M 936M 2% /mnt/disk1
/dev/sdc1          1004M 18M 936M 2% /mnt/disk2
debtest:~# _
```

/dev/sdb1 est formaté en ext3

/dev/sdc1 est formaté en reiserfs

On additionne les deux disques /mnt/disk1(sdb1) et /mnt/disk2(sdc1)

avec la commande:

```
#mhddfs /mnt/disk1,/mnt/disk2 /mnt/ftp/
```

```
debtest:/mnt# mhddfs /mnt/disk1,/mnt/disk2 /mnt/ftp/
mhddfs: directory '/mnt/disk1' added to list
mhddfs: directory '/mnt/disk2' added to list
mhddfs: mount to: /mnt/ftp/
mhddfs: move size limit 4294967296 bytes
debtest:/mnt# _
```

Et comme résultat on obtiend une belle partition de 2G.

```
debtest:/mnt# df -h
Sys. de fich.      Tail.  Occ.  Disp.  %Occ.  Monté sur
/dev/sda1          3,8G  672M  2,9G   19%    /
tmpfs              126M   0    126M   0%    /lib/init/rw
udev              10M   100K   10M    1%    /dev
tmpfs              126M   0    126M   0%    /dev/shm
/dev/sdb1          1004M   18M  936M   2%    /mnt/disk1
/dev/sdc1          1004M   18M  936M   2%    /mnt/disk2
/mnt/disk1;/mnt/disk2
                  2,0G   35M  1,9G   2%    /mnt/ftp
debtest:/mnt# _
```

La commande peut être tapée avec un **-o** et des options, les options disponibles sont :

- `allow_other` (pour permettre aux autres utilisateurs d'accéder à la partition virtuelle)
- `logfile = /tmp/log.log`
- `loglevel = N`
- `N` peut être 0 = Debug message, 1 = Messages d'info, 2 = Message standard.
- `mlimit-size = (défaut 4giga, mini = 100Mo)`

Si on veut ajouter un automontage avec `/etc/fstab` pour mon exemple voici la ligne :

```
/mnt/disk1,/mnt/disk2 /mnt/ftp fuse defaults,allow_other 0 0
```

Pour ce qui est des limites, la taille maximal d'un fichier copiable dans la partition virtuelle est égale au plus grand espace disque libre sur un des disques durs.

Exemple :

Si on fusionne un disque de 400G avec 200G de libre est un disque de 200G avec 50G de libre.

La taille maximale du plus gros fichier copiable est de 200Giga et non pas 250 Go, même si l'espace disque restant est 250Go.

`mhddfs` est encore en développement donc à utiliser avec précautions.

Convertir des chm en pdf

Macsim's mind: \$tail -f macsim

publié par macsim le vendredi, juin 27 2008 - 09:43

A peine une semaine après [mon billet](#) sur la conversion [chm](#) vers html, [ubuntugeek](#) en poste un sur la conversion chm vers pdf.

Le logiciel utilisé est [chm2pdf](#) qui est disponible dans les dépôt universe d'ubuntu.

L'installation ce fait au travers de synaptic ou alors avec le bien connu

```
sudo aptitude install chm2pdf
```

Il semble y avoir un petit soucis avec les fichiers contenant un(des) espace(s) dans le nom, il suffit de retirer l(es)'espace(s).

Spécifie que la source HTML est non-structurée.

```
chm2pdf -webpage monfichier.chm
```

Spécifie que la source HTML est structurée.

```
chm2pdf -book monfichier.chm
```

divfix++ le réparateur de video

publié par macsim le mercredi, juin 18 2008 - 16:38

J'ai eu il n'y pas très longtemps une video avec l'index endomagé, j'avais l'habitude de réparer les video avec mencoder et je suis tombé par hasard sur [divfix++](#).

L'installation se fait avec le classic **sudo aptitude install divfix++**

La commande s'appel DivFix++

Capture d'écran de la bête.



Xen et le message 4gb seg fixup, process

publié par macsim le mardi, juin 17 2008 - 15:58

En me connectant à un domU depuis mon dom0 j'ai vu apparaitre une multitude message avec **4gb seg fixup, process ...**

Pour résoudre ce problème il faut taper ça depuis le dom0:

```
#echo hwcap 0 nosepeg > /etc/ld.so.conf.d/nosepeg.conf
```

puis un petit **ldconfig** pour appliquer tout ça.

Installation d'une clé wifi USB SAGEM sur une Debian

publié par macsim le jeudi, mars 13 2008 - 00:49



Méthode pour installer la clé SAGEM XG-760A sous Debian.

Installation des packets nécessaires.

```
#aptitude install module-assistant linux-headers-`uname -r`  
ndiswrapper-source
```

Compilation de ndiswrapper

```
#m-a build ndiswrapper-source
```

```
#m-a install ndiswrapper-source
```

```
#modprobe ndiswrapper
```

Installation du driver

```
#wget  
http://macsim.labolinux.net/public/Debian/dongles-sagem/Drivers  
-Dongles-WIFI-Sagem-XG760A.zip
```

```
#unzip Drivers-Dongles-WIFI-Sagem-XG760A.zip
```

```
#cd cd Drivers-Dongles-WIFI-Sagem-XG760A
```

```
#ndiswrapper -i WlanUZG.inf
```

```
#echo ndiswrapper >> /etc/modules
```

Et voilà.

Grub et résolution du tty

publié par macsim le mercredi, mars 12 2008 - 14:01



Bon j'en ai marre de devoir chercher après à chaque fois alors :

Pour du 1024x768 16M de couleur : **792**.

Pour du 1280x1024 16M de couleur : **795**.

Pour ceux qui lise ça et ne comprennent pas de quoi je parle, on peut configurer la résolution de ses TTY (quand on fait ctrl+alt+F1 par exemple) pour régler ça sous grub on édite le fichier /boot/grub/menu.lst et on cherche la ligne (81 normalement) :

```
# defoptions=
```

Et on la change en :

```
#defoptions=vga=792
```

ensuite un petit coup de

```
update-grub
```

et on redemarre.

Dsh - Shell ditribué

publié par macsim le jeudi, juin 14 2007 - 16:40



Si vous avez plusieurs machines ubuntu sur votre réseau, et que vous devez faire des taches répétitives sur ces machines, dsh va bien vous aider.

Dsh (Distributed SHell ou Dancer's SHell) permet de lancer une commande sur une/des machines ou encore mieux des groupes de machines.

Prenons un exemple concret nous avons dix machines a administrer cinq ubuntu-serveur cinq ubuntu-desktop, nous voulons facilement mettre à jours les dix machines et pour les serveurs nous voulons recevoir un logwatch de temps en temps.

Premièrement nous installons ssh sur les machines.

```
sudo aptitude install ssh  
puis dsh sur notre machine
```

```
sudo aptitude install dsh
```

On génère une clé ssh sur notre machine (vous n'avez pas envie de taper votre mot de passe 10 fois quand même ;))

```
ssh-keygen
```

On obtient quelque chose comme ça.

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/macsim/.ssh/id_rsa):  
(on appuye sur entré)  
Enter passphrase (empty for no passphrase): ( on tape notre  
passphrase, la plus longue possible)  
Enter same passphrase again: (on la retape pour être sur)  
Your identification has been saved in /home/macsim/.ssh/id_rsa.  
Your public key has been saved in /home/macsim/.ssh/id_rsa.pub.  
The key fingerprint is:  
13:b3:4c:ed:5c:bc:c2:e8:1c:a4:e7:XX:XX:XX:XX macsim@leonidas
```

Bon maintenant on va diffuser notre clé publique sur nos machines.

```
ssh-copy-id -i ~/.ssh/id_rsa.pub IP_MACHINE_1  
ssh-copy-id -i ~/.ssh/id_rsa.pub IP_MACHINE_2
```

...

...
ssh-copy-id -i ~/.ssh/id_rsa.pub IP_MACHINE_10

Maintenant si nous essayons de nous connecter sur une des machines ssh nous demande notre passphrase plutôt que notre mot de passe.

Bon vous n'avez pas envie de taper votre passphrase dix fois non plus donc on va la taper une bonne fois pour toute au démarrage de notre session.

On va dans **Systeme-> Préférence -> Sessions**

On fait **Nouveau**
On remplit **Nom:** avec **Ssh Passphrase**
et **commande** avec **ssh-add ~/.ssh/id_rsa**

Au démarrage de gnome il vous sera demandé de taper votre passphrase et elle sera valide pour la durée de votre session.

Pour voir si tout à bien fonctionné on se connecte en ssh normalement aucun mot de passe nous est demandé.

Bon c'est la que dsh intervient ;)

Dans /etc/dsh nous avons 2 fichiers de configurations:

- dsh.conf

```
verbose = 0 (je vous conseil de laisser 0)
remoteshell = rsh (on laisse comme ça)
showmachinenames = 0 (permet d'afficher le nom des machines)
```

- machines.list

Ce fichier contient nos machines
par default il y a localhost que je vous conseil de retirer sauf si vous voulez faire partie du groupe de machine.
a la place on met les ip des machines/nom dns que l'on administre
dans notre cas on aura quelque chose comme:

```
192.168.1.10
192.168.1.11
...
192.168.1.19
```

Maintenant nous avons le dossier group

Dans ce dossier par default il y a all qui est un lien vers /etc/dsh/machines.list

Nous créons un fichier que nous appellerons serveurs et qui contiendra les ip ou nom dns de nos serveurs.
Ensuite nous créons un fichier que nous appellerons desktop qui contiendra les ip ou nom des postes desktop.

Donc pour serveurs nous aurons quelque chose comme ceci:

```
192.168.1.10
192.168.1.11
serveur3
serveur4
192.168.1.14
```

et pour desktop

```
192.168.1.15
desktop2
192.168.1.17
desktop4
192.168.1.19
```

Maintenant si nous voulons lancer une commande sur tous les postes nous tapons:

```
dsh -g all NOTRECOMMANDE
```

si nous voulons lancer une commande sur tous les serveurs:

```
dsh -g serveurs NOTRECOMMANDE
```

si nous voulons lancer une commande sur tous les postes clients

```
dsh -g desktop NOTRECOMMANDE
```

pour compléter l'exemple pris au début nous pourrions lancer tous les matins

La mise à jour du parc

```
dsh -g all aptitude dist-upgrade (a condition d'avoir utiliser le compte root c'est pas tiptop comme méthode)
```

Le logwatch des serveurs

```
dsh -g serveurs logwatch -mailto mon@address.mail
```

Voilà amusez vous bien