Trucs, astuces et rappels livrés en vrac...
Si vous ne comprenez pas ce que vous faites, ne le faites pas !

Setting Semaphore Parameters

Fix Me!

To determine the values of the four described semaphore parameters, run:

```
# cat /proc/sys/kernel/sem
250     32000   32      128
```

These values represent SEMMSL, SEMMNS, SEMOPM, and SEMMNI.

Alternatively, you can run:

```
# ipcs -ls
```

All four described semaphore parameters can be changed in the proc file system without reboot:

```
# echo 250 32000 100 128 > /proc/sys/kernel/sem
```

Alternatively, you can use sysctl(8) to change it:

```
# sysctl -w kernel.sem="250 32000 100 128"
```

To make the change permanent, add or change the following line in the file **/etc/sysctl.conf**. This file is used during the boot process.

```
# echo "kernel.sem=250 32000 100 128" >> /etc/sysctl.conf
```

NFS

Les exports sont réalisés grâce a l'extension NFS de Service Guard
Le fsid de chaque export est précisé lors de l'exportfs afin d'éviter les erreur “NFS Stale” après changement de serveur

Les imports sont réalisés au travers du service autofs
Le fichier /etc/auto.master a été configuré avec ajout de la ligne :

```
/import /etc/auto.import nfsvers=3 --timeout 5
```

Un fichier /etc/auto.import definis les possibilités d'import :

```
EPP -fstype=nfs,rw,udp,nosymlink PROnfs-vip:/export/sapmnt/EPP
ECP -fstype=nfs,rw,udp,nosymlink PROnfs-vip:/export/sapmnt/ECP
INTERFACES -fstype=nfs,rw,udp,nosymlink PROnfsvip:/export/sapmnt/INTERFACES
```

L’option nosymlink est indispensable pour permettre un accès au imports nfs sur le serveur nfs lui même sans utilisation directe de la ressource exportées. Si cette option n'est pas utilisée, cela se traduit par l'ouverture des fichiers présents sous /export/sapmnt lors de l'accès a un fichier par /import sur le serveur NFS. Ce comportement ne permet donc pas de basculer le package nfs d'un serveur a l'autre car les fichier sont ouverts et qu'il n'est donc pas possible de démonter le FS pour le remonter sur un autre serveur du cluster.

How to mount ISO images on HP-UX, Solaris and Linux

On HP-UX

For version 11.11:

• Start the pfs mount daemon and the pfs daemon:

```
shell> nohup pfs_mountd &
shell> nohup pfsd &
```

• Then create the directory to mount to and run the pfs_mount command:

```
shell> mkdir /cdrom
shell> pfs_mount -o xlat=UNIX /path/to/isofile /cdrom
```

Version 11.31:

• Create a new logical volume say ‘iso’ of size slightly greater than the size of your iso image in MB. You can do this from the admin tool-sam, or by using the following command.

```
shell>lvcreate -L size_of_your_iso -n iso /dev/vg00
```

• Write your iso to the new volume. Needless to say, you must have the required free space.

```
shell>dd if=path/to/iso of=/dev/vg00/riso bs=64
```

• Now mount it on the directory of your choice, say /cdrom:

```
shell>mount /dev/vg00/iso /cdrom
```

On Solaris

• Make the iso image accessible through the block device. (See the lofiadm man page for more details)

```
shell> lofiadm -a /path/to/iso/your_isao.iso
```

• Then mount it wherever you want to, say on /cdrom

```
shell> mount -F hsfs -o ro /dev/lofi/1 /cdrom
```

On Linux

• Use the following command to mount on the directory of your choice, say /mnt/myisodir

```
mount -o loop myiso.iso /mnt/myisodir
```

Machine virtuelle ou serveur physique ?

```
# dmidecode -s system-product-name
VMware Virtual Platform
```

```
# dmidecode -s system-product-name
OptiPlex 760
```

Vérifier les programmes utilisant xinetd

```
egrep libwrap /usr/bin/* /usr/sbin/* | sort
```

Création d'un CD-Rom de boot

copier le répertoire isolinux du CD-Rom:

```
# cp -r /mnt/cdrom/isolinux cible
# cd cible
# chmod u+w *
# cd ..
# mkisofs -o file.iso -b isolinux.bin -c boot.cat \
          -no-emul-boot -boot-load-size 4 \
          -boot-info-table -R -J -v -T cible/
```

Ajouter une imprimante réseau

```
/usr/sbin/lpadmin -p net_printer -E -m printer.ppd -v socket://192.168.0.1
```

Mutualiser les connexions SSH

Ajouter dans le ~/.ssh/config :

```
Host *
 ControlMaster auto
 ControlPath ~/.ssh/master-%r@%h:%p
```

Montage cifs Cofiroute

```
mount -t cifs -o username=xxxx "\\\\sevres\\DSO\\DS3I" /mount_point
```

Montage DFS CentOS

install the package “keyutils”:

```
yum install keyutils
```

add these lines to the end of /etc/request-key.conf:

```
create cifs.spnego * * /usr/sbin/cifs.upcall -c %k
create dns_resolver * * /usr/sbin/cifs.upcall %k
```

After that you can mount remote DFS shares as a normal samba share.

Here is some background . . .

In Red Hat Enterprise Linux 4 the support for CIFS was the job of the “smbfs” file system which lived in userland.

When “smbfs” was dropped and the “cifs” driver was added Kerberos and DFS were not supported. This is because the “cifs” driver is in the kernel where no DNS or Kerberos libraries exist! So, to get DFS and Kerberos working now the cifs driver calls userspace via the program /usr/sbin/cifs.upcall. This userspace helper program then performs Kerberos and DFS related functionality for the kernel.

drop_caches

Writing to this will cause the kernel to drop clean caches, dentries and inodes from memory, causing that memory to become free.

```
To free pagecache:
echo 1 > /proc/sys/vm/drop_caches
To free dentries and inodes:
echo 2 > /proc/sys/vm/drop_caches
To free pagecache, dentries and inodes:
echo 3 > /proc/sys/vm/drop_caches
```

As this is a non-destructive operation and dirty objects are not freeable, the user should run `sync' first.

Mesurer l'entropie disponible

```
cat /proc/sys/kernel/random/entropy_avail
```

/dev/urandom est pseudo aléatoire, très pratique pour la plupart des usages, il permet de générer autant de nombres qu'on veut à un rythme soutenu.

/dev/random est complètement aléatoire, on ne l'utilise que lorsqu'on en a vraiment besoin, il utilise directement le pool d'entropie et ne peut donc fournir qu'un nombre limité de bits, il se remplit progressivement mais lentement.

Proxy SSH

Vous souhaitez accéder à un serveur2 qui n'est accessible qu'à partir de serveur1 en une seule commande SSH depuis un de vos serveur.

Solution : modifier la configuration ssh de micha.

Dans le fichier .ssh/config présent dans votre $HOME, ajouter :

```
host nom_serveur2
   Hostname IP serveur2
   ProxyCommand ssh serveur1 nc %h %p 2> /dev/null
```

Vous pouvez ensuite utiliser ssh user@nom_serveur2 pour vous connecter à serveur2. On vous demandera le mot de passe du compte sur serveur1 puis le mot de passe sur serveur2. Si vous souhaitez utiliser un compte particulier pour vous connecter à serveur1 vous pouvez utiliser la commande suivante :

```
   ProxyCommand ssh user@serveur1 nc %h %p 2> /dev/null
```

Le nom_serveur2 est un nom arbitraire. Vous pouvez mettre toto, titi, ... c'est comme vous voulez.

Le chemin de la connexion est le suivant : ma_machine > serveur1 > serveur2. Si serveur1 et serveur2 possède un réseau privé, vous pouvez utiliser l'adresse privé.

Checking read-only filesystems

```
awk '$4~/(^|,)ro($|,)/' /proc/mounts
```