

**M6 : Administration Système – TP2**  
Olivier Dalle – Katell Morin-Allory – Eric Vecchie

---

## 1 Un peu de préparation

**Utilisation de l'espace disque du serveur.** Les machines ont été réinstallées pour le TP de la même façon que l'installation que vous aviez faite au TP précédent. De plus, les répertoires suivants de notre serveur ns.minfo.bug sont montés en /serv304/ :

- share/ : partition en lecture seule où vous pouvez récupérer les paquetages dont vous avez besoin. Cette partition est en lecture seule
- archive/ : une partition où vous pouvez sauvegarder des fichiers d'un TP à l'autre (sachant que les machines sont réinstallées à la fin de chaque séance). Vous vous créez un répertoire propre dans le sous-répertoire *Sauve/* de cette partition (vous lui donnez votre nom). Attention cette partition est accessible en écriture avec les mêmes droits pour tout le monde : faites attention à ne pas détruire le travail de vos petits camarades !

## 2 Terminez le TP précédent !

Ceux qui ont passé trop de temps sur la partie installation du TP1 doivent en priorité essayer de reprendre et terminer l'exercice sur la partie LVM. Pour cela, les machines ont été automatiquement réinstallées dans une configuration équivalente à celle que vous auriez pu obtenir à l'issue de la phase d'installation. Il suffit donc de se connecter root (mdp = root,,) et d'aller aussi loin que possible dans l'exercice. Essayez toutefois de ne pas y passer plus d'une heure, afin de garder un minimum de temps pour les exercices d'aujourd'hui...

## 3 Systèmes de fichiers

- A l'aide de la commande `dumpe2fs`, observez les différences de configuration qui existent entre les différentes partitions de votre système. Faites par exemple un comparatif entre les partitions / (racine) et /usr (taille des blocs, nombre d'i-node, etc). Comment expliquez-vous ces différences ?
- A l'aide de la commande `tune2fs`, modifiez la valeur du nombre maximal de montage sans vérification de la partition /home pour 2 et mettez à zéro son compteur de montage.
- Vérifiez que votre modification a été prise en compte (à l'aide de la commande `dumpe2fs`)
- Démontez puis remontez 3 fois cette partition. Vérifiez que ces montages ont bien été pris en compte. Que devrait-il se passer lors du redémarrage de la machine ? Redémarrez la machine pour vérifier.
- A l'aide de l'utilitaire `fdisk`, créez une nouvelle partition logique, de type linux, de taille 1 Go.
- Formatez cette nouvelle partition en ext2 (donc non journalisé) de type "news" (man `mke2fs`). Notez le nombre d'i-nodes, de blocs, de blocs réservés et de blocs libres de cette partition.
- Reformatez cette partition en ext2 mais cette fois de type "largefile", et comparez les nouvelles valeurs des nombres d'i-nodes, et de blocs avec celles du formatage précédent.
- Utilisez `tune2fs` pour transformer cette partition en ext3.
- **Attention** : avant d'attaquer la question suivante : faites une copie de sauvegarde du contenu de /sbin...
- Positionnez l'attribut "immutable" sur tous les fichiers du répertoire /sbin.
- Vérifiez sa présence avec la commande `lsattr`
- Vérifiez son efficacité avec quelques commandes violentes : `rm -f /sbin/* ...`

## 4 SF et loop device

Le SF est extrêmement flexible et permet de gérer des disks virtuels à l'aide du loop device.

- Cherchez et installez la doc php dans les images ISO disponibles dans le répertoire *share* du serveur (donc sans utiliser les CDs).
- Créez un SF virtuel dans un fichier
  - Commencez par créer un fichier de 10000 blocs de 4Ko contenant un système ext2.
  - Ajoutez ce fichier dans le SF et faites quelques essais de lecture/écriture.

## 5 Réparer après un désastre

Il peut arriver qu'à la suite d'une fausse manœuvre ou à cause d'un virus la table des partitions soit abîmée. Nous allons voir comment il est possible de la réparer.

- Utilisez la commande `fdisk` pour lister vos partitions et notez **scrupuleusement** les informations concernant les cylindres (début et fin) et le type de chaque partition.
- Effacez le MBR et la table des partitions en écrivant un bloc de 512 octets nuls (utilisez `/dev/zero` sur le premier secteur du disque dur (donc directement sur `/dev/hda`). Vous utiliserez pour cela la commande `dd`. Vérifiez le résultat avec la commande `fdisk`.
- Bootez avec le cdrom RedHat et choisissez `linux rescue`
- Dans le shell obtenu, recréez une table des partitions (avec `fdisk`)
- Réinstallez grub dans le MBR à l'aide des commandes `root` et `setup` du shell grub.

Voilà, il ne reste plus qu'à croiser les doigts et à rebooter...

## 6 Disquette/CD de boot ou poupée Russe ?

Voici un exercice un peu plus difficile pour occuper ceux qui ont un peu d'avance. L'idée générale est de comprendre de quelle façon sont construits une disquette ou un CD de boot, afin par exemple d'en produire un jour (peut-être) une version sur mesure, en y intégrant les modules dont le noyau a besoin (ce qui évite d'avoir à recourir à une deuxième disquette de pilotes).

Mais comme ce dernier objectif est trop difficile compte tenu du temps imparti, on se contentera de la première étape : comprendre comment sont organisées les données sur ces disquettes. Vous réfléchirez ensuite par vous-même à la façon de parvenir au bout de la démarche.

Donc notre objectif pour aujourd'hui sera plus simple (et sans intérêt autre que de comprendre) : il s'agit simplement de trouver les modules qui se cachent dans une image et d'en donner le nombre.

L'image que nous allons étudier est celle d'un système de fichiers ISO, destinée à la fabrication d'un CD de boot. Elle se trouve sur le serveur en `share/rh9/image/boot.iso`.

- Récupérez l'image sur le serveur et faites en une copie sur le disque local pour pouvoir la "disséquer" plus librement.
- Puisqu'il s'agit de l'image d'un système de fichiers, vous pouvez la monter avec l'option `loop` de `mount`. Faites-le de façon à ce que le contenu de l'image apparaisse dans le répertoire `/mnt/boot` de l'arborescence.
- Analysez le contenu de ce système de fichiers, et tentez d'en identifier chacun des éléments.
- Vous ne trouvez pas les modules ? C'est normal, ils sont eux-mêmes "imprisonnés" dans l'image d'un système de fichiers : il y a donc une image dans l'image !
- L'image en question est le fichier `initrd.img`. Il s'agit de l'image d'un système de fichiers ext2, destinée à être chargée en mémoire lors du boot, dans un "ramdisk" (`rd = ramdisk`). Ce système de fichier contient l'image d'une racine minimale permettant au noyau de trouver les modules qui lui manquent (certaines fonctions du noyau sont incompatibles entre elles et ne peuvent donc pas être présentes simultanément dans le noyau. Par conséquent, la seule solution consiste à les charger dynamiquement, sous forme de modules pour le noyau).
- Essayez de monter cette image en `/mnt/init` en utilisant à nouveau l'option `loop` de la commande `mount`.
- Vous n'y parvenez pas ? C'est normal, cette image (`initrd.img`) n'est pas directement exploitable... Utilisez la commande `file` pour essayer de comprendre pourquoi, puis contournez le problème (vous serez amené à produire une copie exploitable du fichier `initrd.img` initial).
- Lorsque vous aurez produit la "copie exploitable" évoquée à la question précédente, utilisez-la afin de monter l'image du "ramdisk" en `/mnt/init`.
- Recherchez dans cette image (en `/mnt/init` donc) l'endroit où sont (devraient être) stockés les modules du noyau (ce sont des fichier objets, dont le nom se termine par `.o`).
- Pas de modules ? Là encore, les "packagers" de la RedHat se sont bien amusés, car les modules sont cachés dans un fichier... Utilisez à nouveau la commande `file` pour essayer de trouver lequel, puis essayez d'en extraire le contenu. Attention, cette fois-ci, il ne s'agit pas d'une image de SF...
- Vous y êtes arrivés ? Alors, combien y a-t-il de modules ?