

**Administration système**  
**Module M2102**

*Travaux pratiques*

IUT de Villetaneuse — R&T — DUT R&T 1ere année

Camille Coti  
camille.coti@iutv.univ-paris13.fr



# 1 Gestion des utilisateurs d'un système partagé

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

Compétences attendues à l'issue de cette séance :

- Création et configuration de comptes utilisateurs
- Gestion des droits associés à un utilisateur, groupes d'utilisateurs

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

## Exercice 1.1 : Création d'utilisateurs

1. La commande qui permet de créer un utilisateur est `useradd`. Parcourez l'aide en ligne (`man useradd`) afin de comprendre ses options les plus utiles.
2. Créez un nouvel utilisateur `tp`. Il doit avoir un répertoire utilisateur `/home/tp`.
3. Regardez le contenu des fichiers `/etc/passwd`, `/etc/shadow` et `/etc/group`. Quelles modifications ont été apportées à ces fichiers par la création du nouvel utilisateur ?
4. Créez un utilisateur système `tpsys`. Quelle seront les particularités de cet utilisateur, par rapport aux utilisateurs classiques ?
5. La création de nouveaux comptes utilisateurs est configurée dans le fichier `/etc/adduser.conf`. Parcourez ce fichier. Vous y verrez notamment les valeurs par défaut utilisées lorsque la commande `adduser` est appelée. Y sont définis notamment les intervalles des identifiants utilisateurs et groupes (`iud` et `guid`) pour les utilisateurs normaux et les utilisateurs systèmes. Comment sont appelées les variables définissant les bornes de ces intervalles ?
6. Modifiez ce fichier afin de définir l'`iud` minimal pour les nouveaux utilisateurs à 500.
7. Il est possible de créer des comptes utilisateurs à durée limitée. Une fois le compte expiré, que se passe-t-il ? Comment crée-t-on un compte temporaire ? Créez un compte temporaire `tmpuser`.
8. Quel est l'`uid` de l'utilisateur que vous venez de créer ?
9. La commande `id` permet d'obtenir des informations sur un utilisateur donné. Quelles informations obtenez-vous sur l'utilisateur `tp` que vous venez de créer ? Quelle est la différence avec les informations obtenues par `finger` ?
10. Lorsque l'on crée un nouvel utilisateur, le système utilise des squelettes de fichiers de configuration de l'environnement utilisateur situés dans le répertoire `/etc/skel`. Ces fichiers sont copiés dans le répertoire de l'utilisateur nouvellement créé. Quels fichiers sont présents dans le répertoire `/etc/skel` de votre système ? Comparez-les avec les fichiers présents dans votre répertoire utilisateur.
11. Ajoutez dans ce répertoire `/etc/skel` un fichier `bienvenue.txt` dans lequel vous écrirez quelques lignes. Créez un nouvel utilisateur : quel est le contenu de son répertoire utilisateur juste après sa création ?
12. Lorsqu'un utilisateur se connecte sur un terminal, un message s'affiche dans ce terminal : c'est le `motd` (message of the day). Le contenu de ce message est situé dans le fichier `/etc/motd`. Modifiez le contenu de ce fichier et ouvrez un nouveau terminal. Que constatez-vous ?

## Exercice 1.2 : Groupes d'utilisateurs

1. Créez un groupe `lecteurs` et un groupe `ecrivains`.
2. Créez quatre utilisateurs : `user1`, `user2`, `user3`, `user4`. Mettez `user1` et `user2` dans le groupe `ecrivains` et `user3` et `user4` dans le groupe `lecteurs`.
3. On veut que les utilisateurs `user1` et `user2`, qui font déjà partie du groupe `ecrivains`, fassent aussi partie du groupe `lecteurs`. Pour cela, on leur attribue le groupe `ecrivains` comme groupe primaire, et le groupe `lecteurs` comme groupe secondaire. Mettez `user1` dans les bons groupes en éditant directement les fichiers `/etc/passwd` et `/etc/group`, et `user2` avec la commande `usermod`.
4. Un fichier appartient à un utilisateur et à un groupe. À la création du fichier, l'utilisateur propriétaire est par défaut l'utilisateur qui l'a créé. Quel est le groupe propriétaire ?
5. Créez un répertoire `/opt/donnees` appartenant à l'utilisateur `root`.
6. On souhaite que les utilisateurs `user1` et `user2` aient le droit d'écrire dans le répertoire `/opt/donnees`, tandis que `user3` et `user4` (et le reste du monde) aient le droit de lire uniquement le contenu de ce répertoire. Comment sait-on quelles permissions sont associées à un fichier donné ?
7. Quelles permissions doit-on assigner au répertoire `/opt/donnees` ?
8. Connectez-vous en tant que `user1` et créez un fichier dans le répertoire `/opt/donnees`. Que se passe-t-il ?
9. Connectez-vous en tant que `user3` et essayez de lire le fichier que `user1` vient de créer dans le répertoire `/opt/donnees`. Que se passe-t-il ?
10. Connectez-vous en tant que `user3` et créez un fichier dans le répertoire `/opt/donnees`. Que se passe-t-il ?
11. La commande `umask` définit les droits par défauts des fichiers créés par un utilisateur. Lisez l'aide en ligne de cette commande pour en comprendre le fonctionnement. Attention, les droits définis par `umask` fonctionnent par soustraction (OU logique) et de façon différente selon si le fichier est un fichier normal ou un répertoire : un `umask` valant `0022` définira que les fichiers seront créés avec les droits `644`, soit `rw-r--r--` et les répertoires avec les droits `755` soit `rw-r-xr-x`. Quel est votre `umask` ?
12. Modifiez le `umask` de votre utilisateur actuel pour créer des fichiers ayant par défaut des permissions en lecture et écriture pour leur utilisateur propriétaire et en lecture pour tout le monde et des répertoires ayant par défaut toutes les permissions pour leur utilisateur propriétaire pouvant être traversés et lus par leur groupe propriétaire et uniquement lus par le reste du monde. Créez un fichier et un répertoire et constatez les permissions qui lui sont associées.
13. Supprimez les utilisateurs `user1`, `user2`, `user3` et `user4` ainsi que les groupes `lecteurs` et `ecrivains`. La commande que vous utilisez doit supprimer en même temps le répertoire des utilisateurs.

## Exercice 1.3 : Droits d'administration

1. Les tâches d'administration ne peuvent pas être effectuées par tous les utilisateurs. Elles doivent être réservées à un administrateur : le super-utilisateur, dont le nom de login est `root`. Alors que vous êtes connecté en tant qu'utilisateur normal, changez d'identité pour devenir le super-utilisateur.
2. Regardez la liste des chemins disponibles pour le super-utilisateur dans la variable `$PATH`. Comparez-la avec celle d'un utilisateur normal. Que constatez-vous ?

3. Modifiez le fichier `/etc/sudoers` pour donner à votre utilisateur normal la possibilité d'utiliser `sudo`.
4. Déconnectez-vous de votre identité de super-utilisateur.
5. Exécutez une commande sous l'identité de votre utilisateur en vous donnant temporairement les privilèges du super-utilisateur en utilisant `sudo`.
6. Regardez les dernières lignes du fichier `/var/log/auth.log`. Cherchez ce que vous venez d'exécuter avec `sudo`.
7. On peut aussi donner les droits d'administration (avec `sudo`) à tout un groupe. Créez un groupe `sudoers` dans lequel vous mettez votre utilisateur. Modifiez le fichier `/etc/sudoers` afin de retirer la ligne que vous venez d'ajouter pour donner les droits d'administration à votre utilisateur, et ajoutez le groupe `sudoers`.

## 2 Fixer des limites aux utilisateurs

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

**Compétences attendues à l'issue de cette séance :**

- Configuration de limites systèmes pour les utilisateurs
- Configuration de quotas liés aux utilisateurs sur les partitions du disque
- Notion de système de fichiers, manipulations de l'arborescence de fichiers

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

Dans ce TP, vous aurez besoin de télécharger des fichiers de code source que vous trouverez à l'adresse <http://lipn.fr/~coti/cours/code>.

### Exercice 2.1 : Limites d'utilisation

1. Avec la commande `ulimit`, on peut fixer des limites sur ce que peut faire un utilisateur sur un système, notamment en terme de consommation de ressources. Avec `ulimit -a`, regardez quelles sont les limitations associées à votre compte.
2. D'après la sortie de `ulimit -a`, déduisez comment on peut limiter la taille des fichiers créés par un utilisateur. Fixez cette limite à 131072 Ko.
3. Compilez le code `grosfichier.c`. Que fait ce programme ?
4. Utilisez le programme `grosfichier` que vous venez de compiler pour créer un fichier de 512 Mo. Que se passe-t-il ?
5. Il y a deux types de limites fixées par `ulimit` : les limites soft (avec l'option `-S`) et les limites hard (avec l'option `-H`). Quelle est la différence entre ces deux limites ? Peuvent-elles toutes les deux être modifiées par l'utilisateur ?
6. Quelle est la limite associée à votre compte sur le nombre de fichiers ouverts ?
7. Compilez le code `beaucoupfichiers.c`. Que fait ce programme ?
8. Utilisez le programme `beaucoupfichiers` que vous venez de compiler pour ouvrir plus de fichiers que ne vous le permettent les limitations du système. Que se passe-t-il ?
9. Les limites peuvent être définies par l'administrateur dans le fichier `/etc/security/limits.conf`. Créez deux nouveaux utilisateurs `user1` et `user2` qui sont situés dans le groupe `users`. Modifiez le fichier `/etc/security/limits.conf` afin de limiter le groupe `users` à des fichiers mesurant au plus 50 Mo, et de permettre à l'utilisateur `user1` d'ouvrir au maximum 50 fichiers (limites hard). Vous pourrez utiliser les programmes `grosfichier` et `beaucoupfichiers` pour tester si les modifications sont bien effectives.

### Exercice 2.2 : Systèmes de fichiers

1. Les partitions utilisées dans le système sont déclarées dans le fichier `/etc/fstab`. Ouvrez ce fichier : combien de partition de votre disque dur sont montées dans votre arborescence de fichiers ?
2. Quel est le type du système de fichiers utilisé sur les partitions du disque ?

3. Combien d'espace reste-t-il sur les partitions de votre système ?
4. Installez l'utilitaire `gparted` et lancez-le pour créer une nouvelle partition dans l'espace restant. Vous préciserez dans la configuration de la partition qu'il faut la formater en `ext4`.
5. Nous allons mettre en place un système de gestion de quotas sur la partition contenant les répertoires utilisateurs. Montez la nouvelle partition que vous avez créée au point de montage `/home`. Vous travaillerez sur celle-ci. Éditez le fichier `/etc/fstab` et ajoutez la ligne pour monter automatiquement votre nouvelle partition sur `/home` avec les options `usrquota,grpquota` dans le champ d'options à la ligne idoine.
6. Redémarrez votre machine pour prendre en compte l'utilisation des quotas, ou remontez le système de fichier concerné avec :

```
coti@maximum:~$ sudo mount -o remount /home
```

7. Créez deux nouveaux utilisateurs `user1` et `user2` qui sont situés dans le groupe `users`. Vous pouvez réutiliser les utilisateurs et le groupe de l'exercice précédent.
8. À la racine du système de fichiers concerné (donc dans `/` ou dans `/home` suivant le partitionnement de votre machine), vous devez créer des fichiers qui vont être utilisés par le système pour mettre en place ces quotas. Activez ensuite les quotas.

```
coti@maximum:~$ sudo touch /home/quota.user
coti@maximum:~$ sudo touch /home/quota.group
coti@maximum:~$ sudo chmod 600 /home/quota.user
coti@maximum:~$ sudo chmod 600 /home/quota.group
coti@maximum:~$ sudo quotaon -a
```

9. Avec la commande `quotacheck`, activez les quotas sur le système de fichiers `/home`. Cela peut prendre du temps.

```
coti@maximum:~$ sudo quotacheck -vguma -F vfstv0
```

10. Vous pouvez voir les quotas actifs sur un système de fichiers donné avec la commande `repquota` ou avec la commande `quota`. Essayez ces deux commandes. Quels sont les quotas actifs pour le moment ?
11. Avec la commande `edquota`, fixez une limite de 2048 fichiers pour le groupe `users` et de 150 Mo pour l'utilisateur `user1`.
12. Regardez quel est maintenant l'état

```
coti@maximum:~$ sudo repquota /home
*** Rapport pour les quotas user sur le périphérique /dev/sdb5
Période de sursis bloc : 7days ; période de sursis inode : 7days
          Block limits                File limits
Utilisateur   utilisé souple stricte sursis utilisé souple stricte sursis
-----
root          --      20      0      0              2      0      0
user1         --      24     150      0             13      0      0
user2         --      16      0      0              4      0      0
coti@maximum:~$ sudo quota -u user1
Disk quotas for user user1 (uid 1005):
Système fichiers  blocs  quota limite sursisfichiers  quota limite sursis
      /dev/sdb5    24    150    0              13      0      0
```

13. Vous allez maintenant vérifier que les quotas fonctionnent bien et ce qui se passe lorsqu'ils sont dépassés. Devenez l'utilisateur `user1`. Créez un fichier de plus en plus gros avec le programme `grosfichier` et le programme `beaucoupfichiers` que vous avez utilisés dans l'exercice précédent. Pouvez-vous faire grossir ces fichiers indéfiniment ? Que se passe-t-il ?

14. Les quotas peuvent être activés et désactivés avec `quotaon` et `quotaoff`. Désactivez les quotas, placez un fichier plus gros que la limite dans le répertoire de `user1`, et réactivez les quotas. Que se passe-t-il ?
15. En quelques phrases, comparez les possibilités offertes par `ulimit` et par les quotas.

## 3 Disques durs et volumes de stockage

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

Compétences attendues à l'issue de cette séance :

- Représentation des disques sous Linux
- Notion de partition et de système de fichiers
- Copie d'un système de fichiers
- Sauvegarde et sauvegarde incrémentale d'un répertoire

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

### Exercice 3.1 : Disques durs et arborescence de fichiers

1. Avec la commande `mount`, repérez quels systèmes de fichiers sont montés sur votre système. De quel types sont-ils ? Si certains sont virtuels, précisez-le. Dans le cas de périphériques matériels, précisez ce que la sortie de `mount` nous apprend sur le type de matériel utilisé.
2. Affichez le contenu du fichier `/etc/fstab` et expliquez pourquoi on y retrouve certaines lignes de la sortie de `mount` tandis que d'autres ne correspondent à aucune entrée de `/etc/fstab`, ou inversement, certaines lignes de `/etc/fstab` ne correspondant à rien dans la sortie de `mount`.
3. Combien de disques sont branchés sur votre système ? Combien de partitions figurent sur ce(s) disque(s) ? Vous pourrez par exemple regarder le contenu de `/proc/partitions` pour obtenir cette information.
4. Quelle est la capacité et l'espace utilisé sur les partitions qui sont actuellement montées sur votre système ?
5. Vous allez créer une nouvelle partition avec `gparted` dans l'espace libre en utilisant tout l'espace disponible, puis la redimensionner pour libérer de l'espace. La taille minimale nécessaire à une partition peut s'obtenir grâce à la commande `resize2fs`. Quelle option lui passe-t-on pour avoir cette information ? Quelle est la taille minimale nécessaire pour la partition de votre choix ?  
Attention : ne faites pas ces manipulations sur votre partition `/`.
6. Démontez la partition et redimensionnez-la. Attention à ne pas être trop serrés : gardez un peu de marge. Vérifiez la taille après le démontage, des mécanismes de cache pouvant écrire sur cette partition au moment du démontage.
7. Avec l'outil `fdisk`, Créez deux partitions de tailles égales dans l'espace libre.
8. Mettez en place un système de fichiers `ext3` sur une de ces partitions.
9. Déposez quelques fichiers sur la partition : par exemple, vous pouvez télécharger une archive quelconque et la décompresser dans la partition.
10. Avec l'outil `dd`, faites une copie de la première nouvelle partition sur la deuxième.

### Exercice 3.2 : Sauvegarde incrémentale

Dans cet exercice, vous allez utiliser l'outil `dump` pour effectuer des sauvegardes incrémentales et son pendant `restore` pour rétablir l'état d'un système de fichiers.



1. Reprenez le système tel qu'il est partitionné à la fin de l'exercice précédent. Vous devez avoir au moins trois partitions. Quelles sont-elles ?
2. Montez toutes les partitions dans l'arborescence de fichiers.
3. Copiez le contenu de la partition / sur une des nouvelles partitions. Vous pourrez pour cela utiliser l'outil `dd`. Effacez le contenu de la deuxième petite partition.
4. Effectuez une sauvegarde de niveau zéro de la partition sur laquelle vous venez de copier des fichiers avec l'outil `dump`.
5. Quelle est la taille des données situées sur la partition que vous venez de sauvegarder ?
6. Quelle est la taille de la sauvegarde ?
7. Ajoutez un fichier contenant le texte de votre choix sur cette partition. Vous avez donc effectué une modification sur le système de fichiers depuis la dernière sauvegarde.
8. Effectuez une sauvegarde incrémentale de niveau 1 de la partition.
9. Quelles sont les données prises en compte dans cette sauvegarde ?
10. Quelle est la taille de la sauvegarde ?
11. Supprimez le fichier texte que vous avez créé sur la partition et effectuez une nouvelle sauvegarde incrémentale de niveau 1.
12. Quelle est la taille de cette nouvelle sauvegarde ? Comparez-la aux tailles des deux sauvegardes précédentes.
13. Vous allez maintenant restaurer le contenu sauvegardé dans la deuxième petite partition. Avec le mode interactif de `restore` (option `-i`, listez les fichiers présents dans la sauvegarde.
14. La récupération des données en passant les sauvegardes les unes après les autres, dans le même ordre que celui dans lequel elles ont été faites. Dans la deuxième petite partition, restaurez le contenu de la première petite partition en utilisant l'outil `restore`.

## 4 Configuration d'un espace isolé sur le disque dur

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

**Compétences attendues à l'issue de cette séance :**

- Isolation dans l'arborescence du système de fichiers avec `chroot`
- Préparation d'un environnement `chrooté` minimal

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

### Exercice 4.1 : Réalisation d'un environnement `chrooté` minimal

1. Créez un répertoire `/var/isolate` qui accueillera votre environnement isolé. Avec `chroot`, tentez de lancer un shell (par exemple `/bin/bash` ayant pour racine de l'arborescence du système de fichier ce répertoire `/var/isolate`. Cela fonctionne-t-il ? Pourquoi ?
2. De quelles bibliothèques dynamiques dépend l'exécutable `/bin/bash` ? Vous pourrez obtenir cette information avec la commande `ldd`.
3. Créez un nouveau répertoire dans lequel vous mettrez tout ce dont a besoin `/bin/bash` pour fonctionner. Attention, vous devez bien respecter les noms des sous-répertoires dans lesquels ils se trouvent, et ne pas oublier de copier `/bin/bash` lui-même.
4. Vous pouvez normalement à présent effectuer le déplacement de racine dans votre nouveau répertoire avec `chroot`.
5. Essayez d'exécuter les commandes usuelles : `ls`, `pwd`, etc. Lesquelles fonctionnent ? Lesquelles ne fonctionnent pas ? Pourquoi ?
6. Vous vous trouvez à la racine de votre environnement `chrooté`. À quel répertoire êtes-vous en réalité dans l'arborescence du système de votre système de fichiers ?
7. Pouvez-vous, depuis votre environnement `chrooté`, vous rendre dans le répertoire `/home` de la machine ? Pourquoi ?

## 5 Installation d'une machine virtuelle avec Virtual-Box

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

**Compétences attendues à l'issue de cette séance :**

- Préparation d'une machine virtuelle avec VirtualBox
- Installation d'une machine Linux dans un système virtualisé

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

Dans cet exercice, vous allez utiliser la solution de virtualisation VirtualBox. Vous allez créer une machine virtuelle et installer une machine Linux sur cette machine virtuelle.

1. Installez VirtualBox avec `apt-get` (paquet `virtualbox`).
2. Lancez VirtualBox. Créez une nouvelle machine virtuelle Linux. Vous devez dans un premier temps préciser les paramètres matériels qui seront données à la machine. Quels paramètres lui donnez-vous ?
3. Votre machine virtuelle doit démarrer sur un disque amorçable. Pour cela, vous allez lui fournir une image de disque (un fichier `.iso`) minimale sur laquelle elle va démarrer, et le reste sera téléchargé par le réseau. Allez sur le site <http://www.debian.org/CD/netinst> et choisissez une image de CD d'installation.
4. Dans la configuration de votre machine, dans l'onglet "stockage", le lecteur CD apparaît comme "contrôleur IDE". Il doit être pour le moment vide. Utilisez l'image ISO que vous venez de télécharger comme image de CD dans le lecteur.
5. Pour que votre machine virtuelle puisse accéder au réseau en passant par la machine hôte, vous devez configurer la façon dont elle accède au réseau. Dans l'onglet "réseau" de la configuration de votre machine virtuelle, assurez-vous qu'une carte est activée et en mode NAT.
6. Démarrez votre machine virtuelle et effectuez une installation du système d'exploitation. Vous expliquerez dans votre compte-rendu ce qui est fait à chaque étape de l'installation.
7. À la fin de l'installation, retirez l'image du CD d'installation du lecteur virtuel (dans la configuration de votre machine virtuelle). Redémarrez votre machine et assurez-vous qu'elle peut bien accéder au réseau.

## 6 Exécution d'un noyau en espace utilisateur avec User-Mode Linux

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

Compétences attendues à l'issue de cette séance :

- Exécution d'un noyau en espace utilisateur avec UML
- Isolation dans l'arborescence du système de fichiers avec chroot

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

Dans cet exercice, vous allez mettre en place et observer un noyau s'exécutant un espace utilisateur (noyau UML).

### 1 Préparation de la partition de stockage

Pour s'exécuter, le noyau UML a besoin d'une partition de stockage. Cette partition est *virtuelle* : il s'agit en réalité d'un (gros) fichier qui va être utilisé comme une partition par le noyau UML. Il contiendra en réalité une *image* du système de fichiers utilisé par le noyau UML.

1. Vous avez déjà utilisé l'utilitaire `dd`. On peut s'en servir pour créer un fichier rempli d'une valeur donnée, par exemple 0. Le pseudo-périphérique `/dev/zero` renvoie 0 : il peut être utilisé comme fichier d'entrée de `dd`. En fichier de sortie, on lui donne le chemin vers le fichier à créer. Il faut ensuite lui donner la taille de blocs et le nombre de blocs à mettre dans ce fichier. En utilisant des blocs de 1024 octets, combien de blocs faut-il pour créer un fichier de 2 Go ?
2. D'après la question précédente, créez un fichier rempli de zéros de 2 Go par blocs de 1024.
3. Sur ce fichier, créez un système de fichiers `ext3`.
4. De la même façon, créez un fichier swap de 256 Mo. Il contiendra un bloc de 256Mo.
5. Montez le système de fichiers dans l'arborescence de fichiers, par exemple au point de montage `/mnt`.
6. Que contient le système de fichiers que vous venez de monter ?
7. Debian vous fournit un utilitaire appelée `debootstrap` qui prépare un environnement de base sur un système de fichiers monté dans l'arborescence. Il n'est pas présent sur vos machines. Installez-le avec `apt-get` et utilisez-le pour installer un environnement de base dans votre image de système de fichiers.

```
root@maximum:/tempo# debootstrap --verbose --arch=i386 --include=aptitude,nano,vim \
wheezy /mnt
```

8. Que contient maintenant ce système de fichiers ?
9. Vous devez monter à la main les pseudo-systèmes de fichiers `/dev` et `/proc` :

```
root@maximum:/tempo# mount -t devpts none /mnt/dev/pts
root@maximum:/tempo# mount -t proc none /mnt/proc/
```

10. Avec `df`, regardez l'espace utilisé par chacun des systèmes de fichiers montés. Quel espace est utilisé sur votre image de système de fichiers ?
11. Vous allez maintenant configurer `apt-get` dans votre image de système de fichiers. Copiez le fichier `/etc/apt/sources.list` au même endroit *dans votre image de système de fichier*.

12. Avec `chroot`, déplacez la racine dans l'environnement que vous venez de créer sur l'image du système de fichiers. Quels fichiers voyez-vous maintenant à la racine ?
13. Il faut maintenant créer manuellement quelques périphériques dans ce système de fichiers. Dans l'environnement `chrooté`, exécutez les commandes suivantes :

```
root@maximum:/# mkdir /dev/ubd
root@maximum:/# cd /dev/ubd
root@maximum:/dev/ubd# for i in 0 1 2 3 4 5 6 7; do mknod $i b 98 $[ $i * 16 ]; done
```

14. Éditez le fichier `/etc/fstab` et remplacez son contenu par ce qui suit. À quoi cette configuration correspond-elle ?

```
root@maximum:/dev/ubd# cat /etc/fstab
/dev/ubda      /          ext3      defaults 0 0
/dev/ubdb     none      swap     defaults 0 0
none          /proc     proc     defaults 0 0
sys           /sys      sysfs    defaults 0 0
none          /dev/pts  devpts   defaults 0 0
```

## 2 Préparation du noyau UML

1. Assurez-vous que vous êtes dans l'environnement `chrooté`.
2. Mettez à jour `apt-get` pour prendre en compte la nouvelle définition des sources, et utilisez-le pour installer les paquets suivants : `kernel-package`, `libncurses5-dev`, `bc` et `uml-utilities`.
3. Placez-vous dans le répertoire `/tmp`. Avec `wget`, récupérez le fichier situé à l'URL `http://www.kernel.org/pub/linux/kernel/v3.x/linux-3.11.tar.gz`. Décompressez l'image obtenue. Quels fichiers contient-elle ?
4. Ces fichiers sont les sources du noyau Linux 3.11. Placez-vous dans le répertoire obtenu et utilisez l'utilitaire `make-kpkg` qui va vous préparer un noyau et en faire un paquet Debian. Cet outil est une spécificité de Debian : vous le trouverez sur les distributions qui en sont dérivées, mais pas sur les autres distributions.

```
root@maximum:/tmp# cd linux-3.11
root@maximum:/tmp/linux-3.11# make-kpkg --arch um --revision 1 --config \
menuconfig --rootcmd fakeroot --append-to-version=-uml kernel-image
[...]
```

Vous aurez peut-être des avertissements à propos des locales : ignorez-les, ce ne sont que des avertissements mineurs. Le menu de configuration du noyau doit s'afficher. Choisissez `Exit`.

Le terminal affiche beaucoup de lignes de compilation, puis vous devez obtenir à la fin :

```
dpkg-deb: building package 'linux-uml-3.11.0-uml' in './linux-uml-3.11.0-uml_1_i386.deb'.
make[2]: Leaving directory '/tmp/linux-3.11'
make[1]: Leaving directory '/tmp/linux-3.11'
```

L'utilitaire `make-kpkg` a créé un paquet `.deb` dans le répertoire parent. Installez-le avec `dpkg`.

5. À la fin de l'installation, vous devez voir l'affichage suivant :

```
update-alternatives: using /usr/bin/linux-3.11.0-uml to provide /usr/bin/linux
(linux) in auto mode
```

C'est donc `/usr/bin/linux-3.11.0-uml` qui est l'exécutable qui servira à lancer le noyau en espace utilisateur.

6. Pour que le noyau UML trouve bien `uml-net`, créez un lien symbolique dans `/usr/bin` :

```
root@maximum:/tmp# ln -s /usr/lib/uml/uml_net /usr/bin/
```

7. Préparez la configuration des terminaux pour n'en avoir qu'un seul, sur le terminal d'où vous lancerez votre noyau en espace utilisateur. Pour cela, éditez le fichier `/etc/inittab`. Cherchez les lignes qui terminent par `ttyX`, X étant compris entre 1 et 6. Commentez les lignes 2 à 6 et sur la première, remplacez `tty1` par `tty0`.

```
1:2345:respawn:/sbin/getty 38400 tty0
#2:2345:respawn:/sbin/getty 38400 tty2
#3:2345:respawn:/sbin/getty 38400 tty3
#4:2345:respawn:/sbin/getty 38400 tty4
#5:2345:respawn:/sbin/getty 38400 tty5
#6:2345:respawn:/sbin/getty 38400 tty6
```

8. Créez un utilisateur de votre choix.
9. Sortez de l'environnement chrooté et démontez les pseudo-systèmes de fichiers qui en font partie et l'image du système de fichiers elle-même.

En cas de message

démontage: /mnt: périphérique occupé.

(Dans certains cas, des infos sur les processus l'utilisant sont récupérables par `lsf(8)` ou `fuser(1)`)

Regarder avec `lsf` quel(s) processus ont des descripteurs de fichiers ouverts vers des fichiers de `/mnt`

```
root@maximum:/tempo# lsf /mnt
COMMAND  PID      USER      FD  TYPE DEVICE SIZE/OFF  NODE NAME
uml_switc 3651 messagebus cwd  DIR   7,1    4096      2 /mnt
uml_switc 3651 messagebus rtd  DIR   7,1    4096      2 /mnt
uml_switc 3651 messagebus txt  REG   7,1    18196 488540 /mnt/usr/bin/uml_switch
uml_switc 3651 messagebus mem  REG   7,1   1351816 98337 /mnt/lib/i386-linux-gnu/libc-2
uml_switc 3651 messagebus mem  REG   7,1   117960 98340 /mnt/lib/i386-linux-gnu/ld-2.1
uml_switc 3651 messagebus 0u  CHR   1,3      0t0   8196 /mnt/dev/null
uml_switc 3651 messagebus 1u  CHR   1,3      0t0   8196 /mnt/dev/null
uml_switc 3651 messagebus 2u  CHR   1,3      0t0   8196 /mnt/dev/null
```

Le tuer et démonter

```
root@maximum:/tempo# kill -9 3651
```

### 3 Démarrage du noyau en espace utilisateur

1. Vérifiez que votre machine peut exécuter le noyau virtuel. Pour cela, il faut que `/dev/shm` (sur la machine hôte) soit exécutable. Vérifiez dans votre `/etc/fstab` que la ligne suivante est bien présente et que l'option `noexec` n'est pas présente. Si besoin, modifiez-le

```
shm          /dev/shm          tmpfs          nodev,nosuid   0          0
```

et montez `shm`

```
root@maximum:/tempo# mount shm
```

2. Remontez votre image de système de fichiers. Lancez votre noyau en lui spécifiant l'image du système de fichier à utiliser, l'image de la partition swap, sur quelle partition se trouve la racine absolue de l'arborescence de fichiers et la quantité de mémoire dont il dispose.

```
root@maximum:/tempo# mount disk.img /mnt
root@maximum:/tempo# /mnt/usr/bin/linux-3.11.0-uml ubda=disk.img ubdb=swap.img \
    root=/dev/ubda mem=256M
```

En reprenant le fichier `/etc/fstab` que vous avez écrit dans la section précédente sur l'image du système de fichiers, comment le noyau virtuel va-t-il voir le disque virtuel `disk.img` ?

3. Avec la commande `uname -a`, vous pouvez obtenir la version du noyau qui s'exécute. Quelle version est exécutée par la machine hôte ? Exécutez-cette commande dans un terminal du noyau UML : quelle version y trouve-t-on ?
4. Avec la commande `ps tree`, on peut voir quel est le processus parent de chaque processus. Cherchez votre noyau dans l'arborescence de processus. Quel est son processus parent ? Où se trouve normalement le noyau dans l'arborescence de processus ?
5. En l'état actuel de votre configuration, pouvez-vous accéder aux fichiers de la machine hôte depuis la machine virtuelle UML ?
6. La commande `sleep` ne fait rien, mais elle le fait pendant un nombre de secondes passé en paramètre. Par exemple, `sleep 20` s'endort pendant 20 secondes. Dans la machine virtuelle UML, lancez un `sleep` suffisamment long pour pouvoir aller lister tous les processus qui tournent sur la machine hôte avec la commande `ps aux`. Voyez-vous votre `sleep` ?
7. Avec la commande `ps aux`, listez les processus vus par le noyau UML. Que constatez-vous entre les processus vus depuis la machine virtuelle et ceux vus par la machine hôte ?

# 7 Mise en place d'un hyperviseur avec Xen et d'un réseau entre les machines virtuelles

Ce TP est à réaliser sur l'image Linux Debian de Marcel Bosc.

**Compétences attendues à l'issue de cette séance :**

- Configuration d'un hyperviseur
- Mise en place de plusieurs systèmes invités
- Réseau entre les machines virtuelles

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat, commandes ou outils utilisés, référence au code source produit qui sera fourni par ailleurs).

Xen est un hyperviseur de type I, ou *baremetal*. Dans cet exercice, vous allez mettre en place et observer des machines s'exécutant au-dessus d'un hyperviseur Xen.

## 1 Préparation de l'hyperviseur

1. Xen est un hyperviseur de type I. Quelle est la différence fondamentale entre ce type d'hyperviseur et les hyperviseurs de type II?
2. Avec la commande `uname -r`, affichez les informations sur le noyau qui est actuellement exécuté.
3. Installez les outils Xen :

```
coti@arthas:~$ sudo apt-get install xen-system-amd64 xen-linux-system-amd64 \
    xen-utils-4.1 xenstore-utils xenwatch xen-tools
```

4. Xen n'est que l'hyperviseur. Installez le gestionnaire de machines virtuelles :  

```
coti@arthas:~$ sudo apt-get install virt-manager
```
5. Regardez dans le répertoire `/boot`. Quels sont les divers fichiers que vous y trouvez ? Que constatez-vous depuis l'installation de Xen ?
6. Mettez à jour le gestionnaire de démarrage Grub :  

```
coti@arthas:~$ sudo update-grub
```

Redémarrez votre machine. Au moment de choisir le noyau sur lequel démarrer (le menu qui apparaît après le menu de Clonezilla), vous devriez avoir une nouvelle entrée permettant de démarrer sur Xen. Choisissez-la.

## 2 Préparation du serveur Xen

Xen distingue deux domaines de systèmes :

- Le domaine 0, ou `dom0`, ou encore domaine privilégié, contient un seul système : c'est le système qui est utilisé au démarrage, celui qui a accès à toutes les ressources. Il est donc particulièrement sensible.
- Le domaine non-privilégié, ou `domU` (pour *unprivileged*), contient les autres systèmes. Les nouveaux systèmes que vous allez installer dans ce qui suit sont donc dans le domaine `domU`.

- (a) Éditez le fichier `/etc/xen-tools/xen-tools.conf` et adaptez ce qui suit en utilisant le répertoire Xen de votre choix (en ayant bien sûr créé ce répertoire au préalable). Attention, utilisez des adresses qui sont dans un réseau différent de celui dans lequel est votre machine sur le réseau de l'IUT.



```

dir = /media/usb/xen
install-method = debootstrap
size = 4Gb
memory = 256Mb
swap = 256Mb
fs = ext3
dist = wheezy
image = sparse
gateway = 10.0.0.1
netmask = 255.255.255.0
passwd = 1
kernel = /boot/vmlinuz-`uname -r`
initrd = /boot/initrd.img-`uname -r`
mirror = http://ftp.lip6.fr/pub/linux/distributions/debian/
ext3_options = noatime,nodiratime,errors=remount-ro
ext2_options = noatime,nodiratime,errors=remount-ro
xfs_options = defaults
reiser_options = defaults
serial_device = hvc0
disk_device = xvda

```

- (b) Configurez la façon dont va être géré le réseau entre les machines virtuelles. Pour cela, éditez le fichier `/etc/xen/xend-config.sxp` et décommentez la ligne (`network-script 'network-bridge bridge=br0'`).
- (c) Redémarrez le démon Xen (vous trouverez le script dans `/etc/init.d` ou avec `systemctl`)

### 3 Préparation d'une machine virtuelle

- (a) Créez une machine virtuelle avec la commande `xen-create-image` que vous adapterez à votre situation.

```

coti@arthas:~$ sudo xen-create-image --hostname=serveurXen.domainelocal.com \
--size=10Gb --swap=512Mb --ip=10.0.0.2 --memory=256Mb --dist=wheezy --arch=i386 \
--role=udev

```

D'après les affichages, quels sont les fichiers créés pour cette machine virtuelle et où sont-ils situés ?

- (b) Le fichier de configuration de cette machine est situé dans le répertoire `/etc/xen`. Lisez-le et décrivez rapidement ce qui y est défini.
- (c) Démarrez la machine avec la commande `xm create` :

```

coti@arthas:~$ sudo xm create /etc/xen/serveurXen.domainelocal.com.cfg

```
- (d) Avec la commande `xm list`, vous pouvez lister les machines en cours d'exécution. À quoi correspond ce qui s'affiche ?
- (e) Avec `ifconfig`, observez les interfaces réseaux disponibles. Que remarquez-vous ?
- (f) La commande `xm shutdown` permet d'éteindre une machine virtuelle en lui passant comme paramètre le nom d'hôte de la machine ou son numéro de machine virtuelle (son *id*). Éteignez la machine virtuelle.
- (g) Créez une deuxième image Xen, que vous appellerez par exemple `clientXen.domainelocal.com`. N'oubliez pas de lui donner une adresse IP différente.
- (h) Démarrez-la et redémarrez l'autre machine virtuelle créée précédemment.
- (i) Avec `ifconfig`, observez les interfaces virtuelles

- (j) La commande `xen-list-images` permet de lister les images de machines virtuelles disponibles sur le système. Qu'obtenez-vous dans votre cas précis ?
- (k) Avec la commande `ps aux`, vous pouvez voir tous les processus en cours d'exécution sur le système. Cherchez notamment ceux qui contiennent dans leur nom `blkback`. Que constatez-vous sur le reste de leur nom ?

## 4 Configuration du réseau

Vous devez commencer cette partie en ayant vos deux machines virtuelles allumées.

- (a) Éteignez les deux machines. En éditant leur fichier de configuration, ajoutez le fait que la machine est dans le bridge `br0` avec `bridge=br0` (note : les paramètres sont séparés par des virgules). Redémarrez les machines. Avec `xen-list-images`, assurez-vous que la modification a bien été prise en compte.
7. Affichez le contenu du fichier `/etc/hosts`. Que constatez-vous ?
  8. Vous devez maintenant configurer le pont au niveau de la machine du domaine 0. Éditez le fichier `/etc/network/interfaces` et ajoutez, en l'adaptant à votre situation (notamment au niveau de l'adresse IP), ce qui suit :

```
auto br0
iface br0 inet static
    bridge_ports bridge_ports regex (eth|vif).*
    address 10.0.0.1
    netmask 255.255.255.0
    network 10.0.0.0
    broadcast 10.0.0.255
    gateway 192.168.1.48
```

Redémarrez la machine et, avec `ifconfig`, observez ce qui a été ajouté.

9. Vous avez configuré dans les parties précédentes le réseau en *bridge*. Cela signifie que le système du domaine 0, qui a accès à la carte réseau, sert de pont entre le monde extérieur et les machines du domaine non-privilegié.
 

La commande `brctl show` permet de visualiser l'état de ce bridge. Qu'observez-vous sur son affichage ?
10. Avec `route`, observez les changements effectués dans la table de routage.
11. Avec SSH, connectez-vous à chacune des machines virtuelles dans des terminaux différents. Avec la commande `w`, vous pouvez voir les utilisateurs connectés et quelques informations les concernant, notamment d'où ils sont connectés. De quelle adresse IP les voit-on venir ?
12. Vérifiez que les machines peuvent communiquer avec le pont avec `ping`.
13. Pour que les paquets puissent sortir de la machine physique et être relayés sur le réseau physique, il faut que la machine en domaine 0 accepte de les router. Activez le routage avec la commande
 

```
coti@arthas:~$ sudo sysctl -w net.ipv4.ip_forward=1
```

 Et vérifiez que chaque machine du domaine non-privilegié peut communiquer avec l'extérieur du réseau (par exemple, `ping 8.8.8.8`).
14. Vérifiez que les machines peuvent bien communiquer l'une avec l'autre avec `ping`.
15. Sur une des deux machines, mettez jour les déclarations de paquets et installez un serveur HTTP Apache (paquet `apache2`).
16. Sur la machine en domaine 0, ouvrez un navigateur et entrez le nom d'hôte de la barre d'adresse (précédé de `http://`). Vous devriez voir s'afficher la page d'accueil de Apache.

17. Sur l'autre machine virtuelle, mettez jour les déclarations de paquets et installez un serveur HTTP `lighttpd` (paquet `lighttpd`). De même, ouvrez un navigateur et entrez le nom d'hôte de la barre d'adresse (précédé de `http://`). Vous devriez voir s'afficher la page d'accueil de `lighttpd`.

## 5 Manipulations de machines virtuelles

Au début de cette partie, vous devez avoir deux machines virtuelles lancées dans laquelle tourne pour l'une un serveur Apache et pour l'autre un serveur `lighttpd`.

1. Avec la commande `sudo xm list`, affichez l'état de vos machines Xen.
2. Observez la colonne "State". En lisant la documentation de `xm` (`man xm`), dans quel état sont vos machines ?
3. D'après la documentation, quelle commande sert à mettre une machine virtuelle en pause ?
4. Mettez en pause une de vos machines du domaine non-privilegié. Quel est maintenant l'affichage donné par `xm list` ?
5. Essayez de mettre en pause la machine de domaine 0. Que se passe-t-il ?
6. D'après la documentation, quelle commande sert à sauvegarder l'état d'une machine virtuelle ?
7. Pouvez-vous sauvegarder l'état de la machine qui est en pause ?
8. D'après la documentation, quelle commande sert à réveiller une machine virtuelle qui est en pause ?
9. Réveillez la machine en pause et sauvegardez son état dans un fichier. Quelle est la taille du fichier obtenu ?
10. Affichez la liste des machines. Que constatez-vous ?
11. Quelle est la différence entre sauver l'état d'une machine et effectuer un checkpoint ? Effectuez un checkpoint de l'autre machine virtuelle et affichez la liste des machines.
12. Éteignez toutes les machines (sauf celle du domaine 0 bien sûr) avec la commande `xm shutdown`. Essayez d'afficher les pages HTML des serveurs HTTP et de contacter les machines avec `ping`.
13. D'après la documentation, quelle commande sert à relancer une machine virtuelle depuis un fichier de sauvegarde ?
14. Restaurez l'état d'une des deux machines et assurez-vous qu'elle tourne. Affichez la page d'accueil de son serveur HTTP dans le navigateur de la machine du domaine 0.