

# TD n° 1

## Installation de l'environnement

---

Le but de ce TD est de télécharger et d'installer sur votre machine l'environnement de travail que nous utiliserons tout au long du semestre.

### 1 Installation d'un environnement de développement

Afin de pouvoir compiler un nouveau noyau, nous allons utiliser une machine sous Linux avec tous les outils de compilation nécessaires. Afin de rester dans nos environnements favoris et pour faciliter la reprise en cas de plantage du noyau que vous aurez modifié, nous allons travailler dans une machine virtuelle.

#### 1.1 Configuration de la machine hôte

Commencez par installer le logiciel qui vous permettra de décompresser les fichiers au format 7zip :

<https://trolen.polytech.unice.fr/cours/sea/td01/install/7z1805.exe>

ou

<https://trolen.polytech.unice.fr/cours/sea/td01/install/7z1805-x64.exe>

#### 1.2 Installation de VirtualBox

Si vous ne l'avez pas déjà (où dans une version antérieure), téléchargez VirtualBox sur votre machine hôte :

<http://trolen.polytech.unice.fr/cours/sea/td01/install/VirtualBox-5.2.18-124319-Win.exe>

ou

<http://trolen.polytech.unice.fr/cours/sea/td01/install/VirtualBox-5.2.18-124319-OSX.dmg>

Pour une installation sur un hôte Unix, voir les différentes possibilités à l'adresse suivante :

[https://www.virtualbox.org/wiki/Linux\\_Downloads](https://www.virtualbox.org/wiki/Linux_Downloads)

### 2 Installation d'un système d'exploitation GNU/Linux

#### 2.1 Installation d'un système à partir d'une image disque de machine virtuelle

Afin d'obtenir un système déjà préconfiguré avec lequel travailler, vous pourrez récupérer les fichiers pour la machine virtuelle que nous allons utiliser durant ces prochains TD à l'adresse suivante :

<https://trolen.polytech.unice.fr/cours/sea/td01/SEA2018.vbox>

<https://trolen.polytech.unice.fr/cours/sea/td01/sda-system.7z>

<https://trolen.polytech.unice.fr/cours/sea/td01/sdb-linux-kernel.7z>

Après avoir démarré VirtualBox, il ne vous reste plus qu'à ouvrir le fichier `.vbox` pour lancer la machine virtuelle.

#### 2.2 Adaptation de la machine virtuelle à votre machine

Démarrer l'application VirtualBox. Dans le menu *Machine*, *Ajouter* la machine virtuelle que vous venez de télécharger en lui donnant le chemin du fichier `SEA2016.vbox`. Pour tirer les meilleures performances possibles de votre machine physique, il va falloir adapter la configuration de base qui vous a été fournie.

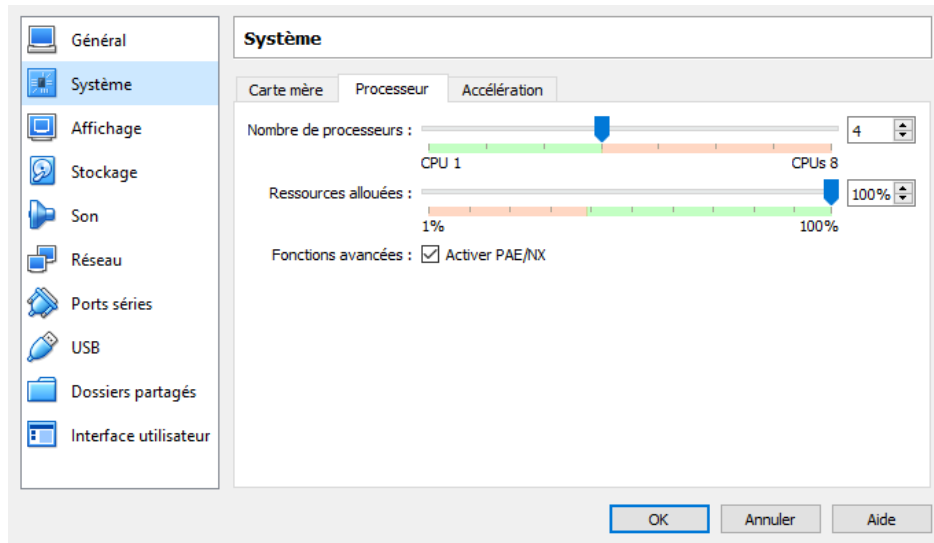
Nous allons devoir compiler de nombreux programmes pendant ce cours. Il est donc nécessaire de tirer au mieux partie des performances de votre machine physique.

##### 2.2.1 Configuration du nombre de processeurs de la VM

Dans le panneau configuration de votre machine virtuelle, aller dans *Système/Processeur* et utiliser le nombre de processeurs spécifiés dans la zone verte (soit la moitié des CPU de votre machine). Cela vous donnera le maximum de puissance de calcul pour la compilation tout en laissant votre système natif fonctionner correctement.

# TD n° 1

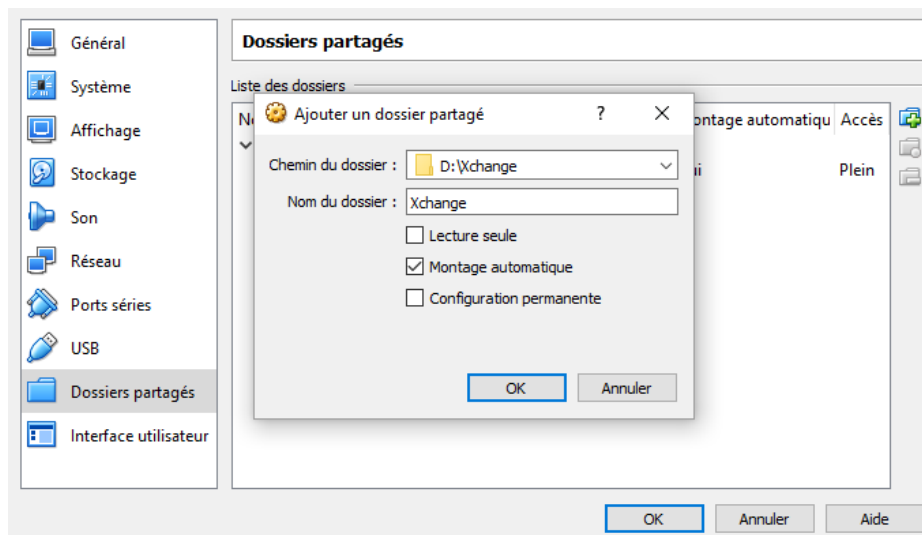
## Installation de l'environnement



### 2.2.2 Echange de données entre la machine physique et virtuelle

Nous allons devoir échanger des fichiers entre la machine physique et virtuelle à certaines occasions. Il est donc nécessaire d'activer le partage de dossier entre machine virtuelle et physique.

Toujours dans le panneau de configuration aller dans Dossiers partagés. Ajoutez-le(s) dossier(s) que vous souhaitez partagés et sélectionnez le montage automatique. Ceux-ci seront alors visible dans le sous dossier `/media/sf_Nom_Dossier` dans votre machine virtuelle.



### 2.2.3 Démarrage de la VM

Votre machine virtuelle étant maintenant configurée au mieux par rapport aux caractéristiques de votre machine, vous pouvez démarrer celle-ci pour débiter le travail. Si le démarrage de votre VM n'est pas possible (vous obtenez une erreur lors du lancement), il faut passer à la section suivante sur l'ajout des « Virtual Guest Addons »

Pour vous connecter sous le système Debian GNU/Linux que vous venez de démarrer, il y a deux comptes qui ont été créés :

- le compte du super utilisateur :            identifiant : `root`            mot de passe : `root`
- un compte utilisateur normal :            identifiant : `user`            mot de passe : `user`

Nous allons travailler la plupart du temps en tant que super-utilisateur.

# TD n° 1

## Installation de l'environnement

### 2.3 Installation des « Virtual Guest Addons »

Pour bénéficier des fonctionnalités avancées depuis la machine virtuelle (copier/coller, accès à un périphérique USB, ...) il est nécessaire d'installer (ou de mettre à jour) les « Virtual Guest Addons » dans la machine virtuelle. Cette installation est indépendante de l'architecture de votre machine. Deux méthodes sont possibles suivant votre configuration : installer ce pack avant de lancer le VM ou bien installer le pack une fois la VM lancée.

#### 2.3.1 Installation des Addons avant le démarrage de la VM

L'installation du pack avant le démarrage de la VM est possible, mais **attention** à bien télécharger la version correspondant à la version du programme VirtualBox que vous avez installé. Pour ceux qui ont installé la version préconisée dans ce sujet, vous pouvez récupérer le fichier d'extension pack à l'adresse suivante, sinon aller sur le site de VirtualBox :

[http://trolen.polytech.unice.fr/cours/sea/td01/install/VirtualBox\\_Extension\\_Pack-5.2.18-124319.vbox-extpack](http://trolen.polytech.unice.fr/cours/sea/td01/install/VirtualBox_Extension_Pack-5.2.18-124319.vbox-extpack)

Sinon, vous pouvez télécharger la version correspondant à VirtualBox à l'adresse suivante :

<https://www.virtualbox.org/wiki/Downloads>

Il ne vous reste alors plus qu'à installer ce logiciel et l'extension pack pour pouvoir exécuter la machine virtuelle que nous allons maintenant télécharger.

#### 2.3.2 Installation des Addons une fois la VM démarrée

Pour installer ou mettre à jour ces addons, il faut :

- Aller dans le menu Périphériques / Insérer l'image CD des Additions Invités (si le cdrom des addons n'est pas déjà installé par la procédure précédente, le système vous proposera de télécharger la version adaptée à votre version de VirtualBox).
- Puis, dans un terminal, en tant que root, faire :

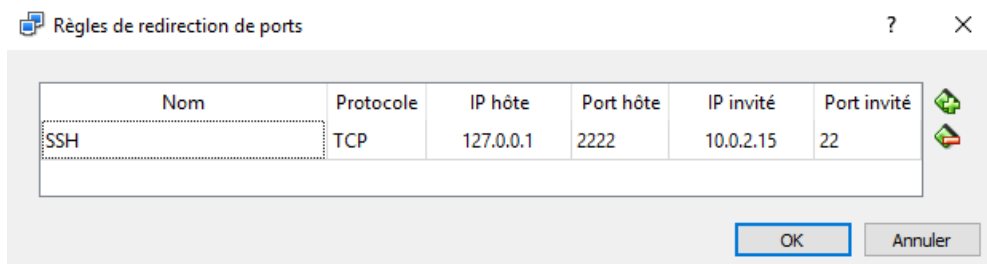
```
mount /dev/cdrom /media/cdrom
cd /media/cdrom
./VBoxLinuxAdditions.run
```

Cette étape installera les ajouts nécessaires à votre système invite pour prendre correctement en compte l'interface graphique

### 2.4 Travailler avec la VM

#### 2.4.1 Communication par SSH avec la VM

Plus nous avons de moyens de communiquer entre la VM et la machine physique, mieux c'est. Un serveur SSH a donc été installé et configuré sur votre VM et une redirection de port a été réalisée entre votre machine virtuelle et votre machine physique via la règle suivante (accessible via Configuration / Réseau / Redirection de ports sur la carte réseau 1).

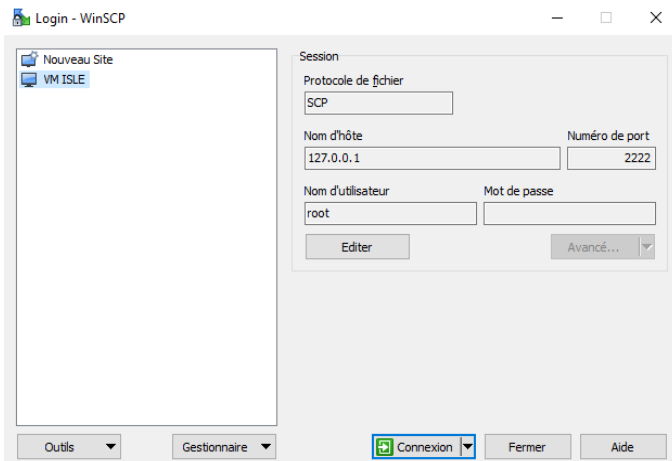


Vous pouvez donc communiquer avec votre machine virtuelle via l'adresse localhost de votre machine physique (127.0.0.1) sur le port 2222 pour adresser le port 22 (SSH) de votre machine virtuelle.

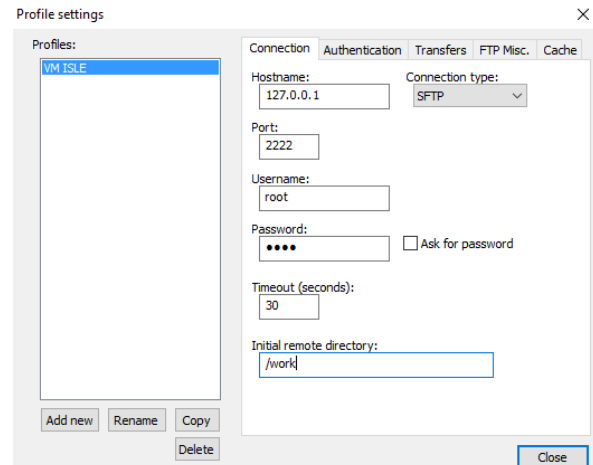
# TD n° 1

## Installation de l'environnement

Ceci vous donne donc accès à votre machine virtuelle via des outils classiques comme WinSCP ou Notepad++ pour copier, déplacer ou éditer des fichiers. Pour NotePad++, il est nécessaire d'installer le plugin NppFTP et de la configurer comme montré ci-dessous



Configuration de WinSCP pour communication avec la VM



Configuration de NppFTP pour NotePad++

### 2.5 Mise à jour de votre système

Vous pourrez mettre à jour votre système Debian GNU/Linux à l'aide des commandes suivantes :

```
apt-get update (mettre à jour la liste des paquetages disponibles)
apt-get upgrade (installer les mises à jour nécessaires)
```

### 2.6 Démarrage de l'environnement graphique

Par défaut, la VM démarre en mode console. Après vous être authentifié, vous pouvez basculer dans le mode graphique grâce à la commande

```
startx
```

Vous vous retrouvez alors dans un environnement graphique minimaliste (Windows Manager FluxBox) qui a l'énorme avantage d'être peut gourmand en espace de stockage contrairement à des environnements comme Gnome ou KDE utilisés dans les distributions classiques.

Vous disposez maintenant de tous les outils nécessaires au TD de ce module ! Nous nous contenterons par la suite d'ajouter des disques durs virtuels supplémentaires à ce système en fonction de nos besoins.

## 3 Configuration et Compilation du noyau

### 3.1 Installation du Noyau Linux

#### 3.1.1 Utilisation d'un noyau préinstallé sur une image

Montez le deuxième disque dur de la machine virtuelle dans l'arborescence de votre système de fichier dans le dossier /work :

```
mount /dev/sdb1 /work
```

Vous avez maintenant accès dans /work aux sources de noyau Linux.

# TD n° 1

## Installation de l'environnement

---

### 3.1.2 Récupération du noyau

Dans le cas où vous auriez besoin d'installer vous-même votre noyau à partir de sources qui se trouvent sur le site <http://www.kernel.org/>, il vous faudra vérifier que l'archive fournie avec la clé de `kernel.org` qui a le numéro `0x6092693E` :

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys 6092693E  
gpg --verify linux-2.6.32.68.tar.sign linux-2.6.32.68.tar
```

et à installer les sources dans `/usr/src`.

### 3.2 Familiarisation avec les sources du noyau

En tant que développeur de modules pour Linux, vous aurez très souvent à trouver où se trouve telle ou telle fonctionnalité dans les sources. Vous devrez être capable de naviguer dans plus de 420Mo de sources et de documentation, soit plus de 12.500.000 de lignes de code C et documentation... Il est donc important de se « familiariser » avec l'organisation des sources du noyau !

Vous pourrez vous aider des vos outils habituels comme `locate`, `find`, `grep`, `ctags`, ...

Vous pouvez vous amuser à chercher :

- le code source du driver pour `nezk` (la carte émulée de notre machine virtuelle)
- la documentation sur les cartes TV DVB (Digital Video Broadcast) supportées par le noyau
- ...

### 3.3 Configuration du noyau

Nous souhaitons construire un noyau Linux pour une configuration minimaliste. Nous devons donc inclure dans cette configuration le minimum de pilotes et de fonctionnalités afin de réduire au minimum le temps de compilation ainsi que les ressources nécessaires pour le faire fonctionner.

Pour éviter de devoir recréer une configuration, ce qui serait trop long pour ce TD, nous allons nous contenter de compiler le noyau avec une configuration existante et de la modifier très légèrement.

Dans l'outil de configuration du noyau, chaque question attend une réponse :

- 'oui' (Y) ou (\*) : cette fonctionnalité sera incluse dans le code du noyau
- 'non' (N) ou ( ) : cette fonctionnalité ne sera pas incluse dans le noyau
- 'module' (M) : cette fonctionnalité sera chargeable dynamiquement dans le noyau (comme un plugin)

Donc pour configurer le noyau Linux avec une configuration existante et pour la modifier, procédez de la manière suivante :

- Copier le fichier de configuration fourni dans les sources du noyau linux que vous souhaitez compiler en le nommant `.config`
- Lancer la commande `make oldconfig` pour vérifier que la configuration est correcte pour cette version de noyau
- Lancer `make menuconfig` pour adapter la configuration selon vos besoins. Nous allons ajouter un pilote pour permettre la prise en charge des supports de stockage USB : rendez-vous dans le menu `Devices Drivers / USB Support / USB Mass Storage Support`. Activez cette fonctionnalité en tant qu'incluse dans le noyau.

En quittant la configuration, le programme vous demande de confirmer l'enregistrement de votre configuration. Cette configuration sera enregistrée dans le fichier `.config`. Si vous souhaitez faire une sauvegarde de ce fichier de configuration, il vous suffit de copier ce fichier sous un autre nom. Mais le `Makefile` du noyau prendra toujours le fichier `.config` comme étant le fichier de référence par rapport auquel il fera la compilation.

# TD n° 1

## Installation de l'environnement

Et n'oubliez pas non plus d'activer la bonne option à `make` pour avoir une compilation le plus rapide possible :

```
make -j 2 x nb cœurs
```

Le résultat de la compilation se trouve à l'endroit suivant :

```
linux-2.6.32.68/arch/x86/boot/bzImage
```

### 3.4 Test et déploiement du noyau

#### 3.4.1 Test du noyau avec une machine virtuelle

Une fois la compilation terminée, il ne vous reste plus qu'à installer ce nouveau noyau. Pour éviter de modifier le système avec lequel vous travaillez, nous allons utiliser un système émulé dans un environnement virtuel. Pour cela, nous allons utiliser `qemu` qui va nous permettre de démarrer un système existant avec ce nouveau noyau compilé.

```
qemu-system-i386 -m 64 -kernel linux-2.6.32.68/arch/x86/boot/bzImage -hda hda-bench.qcow2 -append root="/dev/hda1"
```

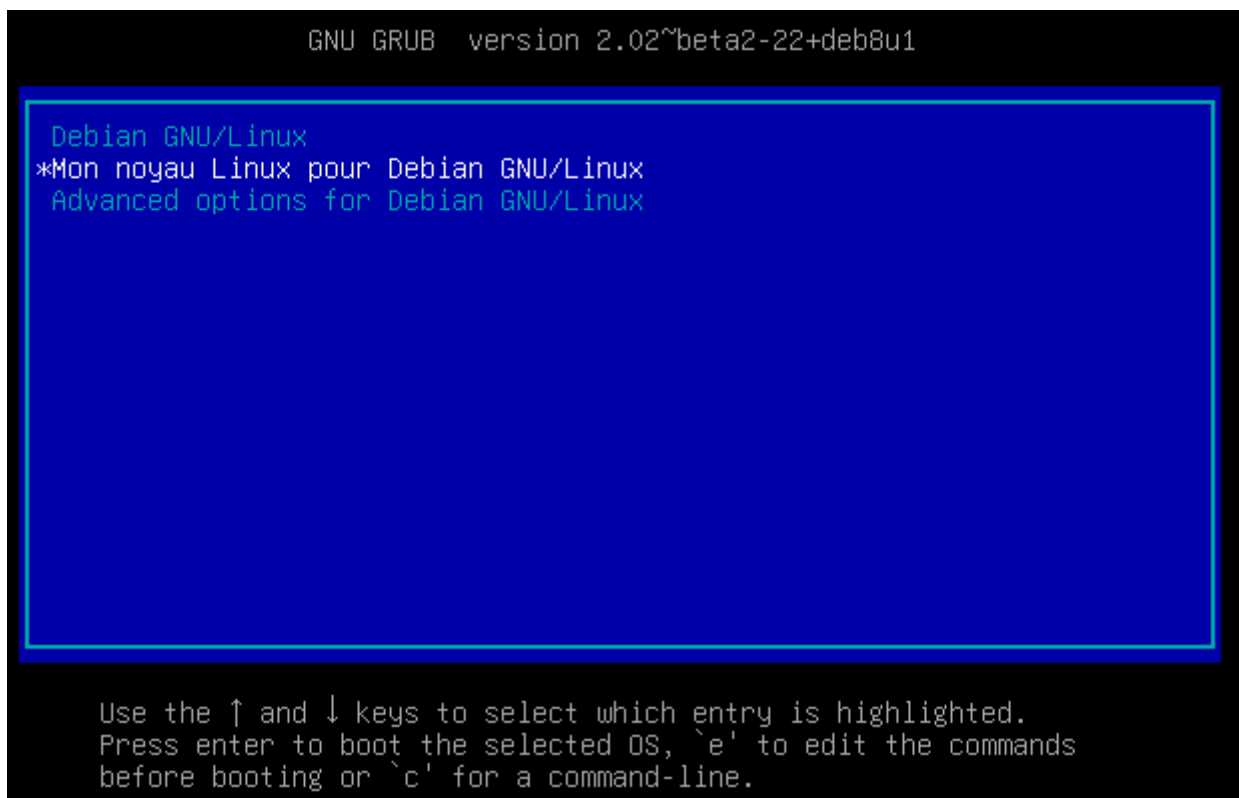
En cas d'erreur c'est que vous n'avez pas inclus toutes les fonctionnalités nécessaires pour démarrer votre système. A vous de diagnostiquer la source possible de l'erreur.

#### 3.4.2 Installation du noyau sur votre système

**A ne pas faire sur votre système, mais pour information !**

Vous pouvez aussi installer ce noyau pour votre machine virtuelle de travail (les pilotes pour les matériels émulés ont été inclus dans cette configuration).

Commencez par copier les différents fichiers nécessaires (noyau, cartographie, configuration) puis modifiez le chargeur de noyau qui est installé sur votre système (`/boot/grub/grub.cfg`) afin de vous proposer ce nouveau noyau au prochain reboot avec les bons paramètres. Ce noyau ne devra pas être le choix par défaut (pour continuer à travailler avec le système complet et bien configuré), mais un des choix possibles du démarrage du système.



# TD n° 1

## Installation de l'environnement

**Attention !** La dénomination de la partition root de votre système utilise le système de nommage UUID. Celui-ci ne peut être utilisé dans notre cas présent (utilisation via un script dans un système de fichier alternatif pour le boot que l'on verra lors d'un prochain cours). Vous devrez donc remplacer dans le fichier de configuration de grub la définition de la partition root du système de fichier en remplaçant l'UUID par `root=/dev/sda1`

Supprimez la référence à l'UUID pour la remplacer par `/dev/sda1` et supprimez l'option `quiet` afin de pouvoir profiter de tous les messages du noyau lors de son démarrage.

Arrivez-vous à démarrer le système ? Non ? Si vous avez une erreur lors du démarrage, cherchez à trouver l'origine de celle-ci pour ajouter le support de la fonctionnalité manquante dans votre noyau et testez à nouveau celui-ci.

## 4 Appels système

### 4.1 Ajouts d'appels systèmes au noyau

Toujours sur votre noyau 2.6.32.68, nous allons tenter d'ajouter un appel système et de l'invoquer par un programme de test.

Pour bien comprendre le fonctionnement d'un appel système, il suffit d'en créer un. Pour être sûr que vous avez bien compris, vous allez en créer deux nouveaux.

En prenant modèle sur l'exemple donné en cours, ajoutez les appels système suivants au noyau Linux :

```
void addition(int x, int y, int *res) ;
void increment(int *x) ;
```

Une fois ces nouveaux appels systèmes ajoutés au noyau, réamorcer la machine en prenant soin de démarrer sur le noyau auquel vous avez ajouté ces nouveaux appels système.

### 4.2 Programme testant les appels systèmes créés

Écrire un programme de test pour vos nouveaux appels systèmes (c'est le code qui devrait être ajouté à la bibliothèque C pour utiliser vos nouveaux appels système).

Ecrivez donc un simple programme `test_syscall` (comme présenté en cours) pour faire ce test. Simple ne veut pas dire simpliste ; lors de vos tests, vous prendrez soin d'essayer des adresses impossibles (ou protégées en écriture) pour le paramètre `res` qui est passé par adresse (NULL par exemple) et une constante pour le paramètre d'appel de la fonction `increment` (protection en écriture). Ainsi vous pourrez vraiment tester si vous avez correctement réalisé l'implémentation de ces appels systèmes en faisant correctement les copies de données de l'espace utilisateur vers l'espace noyau et vice versa).

Nous allons tester notre nouveau noyau avec ces deux appels systèmes ajoutés. Pour cela, vous lancerez `qemu` comme précédemment, mais en ajoutant des paramètres supplémentaires à la ligne de commande. Cela démarrera le réseau de la machine virtuelle `qemu` et nous permettra de communiquer avec elle.

```
qemu-system-i386 ... -device ne2k_pci,netdev=mynet0 -netdev
user,id=mynet0,net=192.168.10.0/24,dhcpstart=192.168.10.10 -redir tcp:5555::22
```

Vous utiliserez la commande `scp` depuis votre machine virtuelle de travail (et pas la machine `qemu` qui nous permet de tester notre noyau uniquement) pour vous connecter par `ssh` à votre machine virtuelle `qemu` et copier le fichier de test que vous avez créé via la commande :

```
scp -P 5555 test_syscall root@localhost:/root/
```

Puis depuis la machine virtuelle, vous vérifierez que votre programme fonctionne bien.