

# LO52 – TP2

## Tatiana GASSER & Loïc MARTIN

### 1. Téléchargement d'une VM Linux

Installation de Virtual Box et téléchargement de Ubuntu 20.4.

On ajoute également à la VM python et git.

```
Sudo apt install python
```

```
Sudo apt install git
```

### 2. Ajout de repo sur la VM Linux

```
sudo snap install git-repo
```

On a essayé d'utiliser « `sudo apt install` » repo mais le paquet repo n'est pas trouvé. Donc on est passé par snap.

### 3. Enlever snap git repo

```
sudo snap remove git-repo
```

On est dans l'impossibilité d'utiliser repo. Soit les droits sont refusés, soit-il ne le trouve pas. On a donc essayé de le supprimer et le réinstaller depuis le site source mais sans succès. On pense que le problème vient en partie de la version d'Ubuntu utilisée pour la VM qui serait trop récente. De plus, la VM a tendance à crasher pendant une dizaine de minute, donc on a préféré télécharger la version 18.4 d'Ubuntu en attendant. Mais comme on ne peut pas se connecter au réseau de l'UTBM et qu'on utilise la 4G par partage de réseau, cela prend du temps.

### 4. Réinstallation depuis le site source

Finalement, on a réussi à trouver un moyen de télécharger repo directement depuis le site. Donc on est resté sur la version 20.4 de Ubuntu. Le site qui nous a débloqué est :

<https://askubuntu.com/questions/1247103/why-is-the-repo-package-unavailable-in-ubuntu-20-04-how-can-i-install-it>

*On a exécuté les commandes suivantes :*

```
mkdir -p ~/.bin
```

```
PATH="${HOME}/.bin:${PATH}"
```

```
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/.bin/repo
```

```
chmod a+rx ~/.bin/repo
```

En testant cette méthode, on s'est rendu compte que « `curl` » n'était pas présent sur la VM. Donc on a installé curl avec « `sudo apt-get install curl` ».

Après 40 minutes d'installation, l'espace libre de stockage était plein, donc il a fallu tout recommencer...

Une fois installé, on a eu accès a repo et on a enfin pu commencer le TP.

#### 5. Créer un dossier et se positionne dedans

*mkdir android-kernel && cd android-kernel*

#### 6. Cloner le repo avec la bonne branche

*repo init -u <https://android.googlesource.com/kernel/manifest> -b hikey-linaro-android-4.19*

#### 7. Synchroniser le repo

*repo sync*

#### 8. Pour comparer les deux branches

Noyau DL -> config

Identifier la config défaut -> charger

Faire un « diff » entre les deux

#### 9. Android Hikey Linaro

La dernière version de Hikey Linaro est disponible ici :

<https://android.googlesource.com/kernel/hikey-linaro/+/refs/heads/android-hikey-linaro-4.19>

*cd hikey-linaro*

#### 10. Modification du dossier de config

Pour modifier le dossier de config, on s'est inspiré de :

<https://android.googlesource.com/platform/external/qemu/+/emu-master-dev/android/docs/ANDROID-KERNEL.TXT>

Il a fallu fouiller dans les dossiers téléchargés.

*cp arch/mips/configs/generic/board-ranchu.config .*

*make board-ranchu.config*

*repo diff*

#### 11. Création d'un script

On a réalisé un script « tp.sh » pour automatiser les commandes (voir également dans le fichier tp.sh) :

*Création d'un dossier pour le TP :*

*mkdir android-kernel && cd android-kernel*

*Initialisation du dossier repo et la branche hikey-linaro*

*repo init -u <https://android.googlesource.com/kernel/manifest> -b hikey-linaro-android-4.19*

*Téléchargement*

*repo sync*

*Copie du fichier*

*cp hikey-linaro/arch/mips/configs/generic/board-ranchu.config hikey-linaro/arch/arm64/configs/*

*Activation du fichier config*

*make board-ranchu.config*

*repo diff*

## 12. Modification de la config

On a modifié la config avec les nouvelles caractéristiques demandées. Voir fichier

« newconf\_defconfig ».