

## TP2 – Manipulation d'un kernel Linux

### 1) Installation d'une machine virtuelle Ubuntu

Dans un premier temps, pour réaliser ce TP, il nous a fallu installer une machine virtuelle Ubuntu et la déployer sur VirtualBox., le but étant en effet d'utiliser l'outil repo sur Linux. L'image Ubuntu est disponible sur le site suivant :

**<https://www.osboxes.org/ubuntu/>**

Nous avons sélectionné la version 20.04. Il s'agissait ensuite de monter la machine virtuelle directement sur VirtualBox.

### 2) Préparation d'un environnement permettant de configurer un noyau hikey-linaro

Dans un premier temps, il nous a fallu installer python sur la machine Ubuntu à l'aide de la commande suivante :

**sudo apt install python**

En effet, nous obtenions une erreur de type « Permission denied » lors de notre tentative d'initialisation de l'environnement (que nous décrirons par la suite).

```
osboxes@osboxes:~/android-kernel$ repo init -u https://android.googlesource.com/platform/manifest
Traceback (most recent call last):
  File "/home/osboxes/android-kernel/.repo/repo/main.py", line 49, in <module>
    import event_log
  File "/home/osboxes/android-kernel/.repo/repo/event_log.py", line 167, in <module>
    _EVENT_ID = multiprocessing.Value('i', 1)
  File "/snap/git-repo/18/usr/lib/python2.7/multiprocessing/__init__.py", line 253, in Value
    return Value(typecode_or_type, *args, **kwargs)
  File "/snap/git-repo/18/usr/lib/python2.7/multiprocessing/sharedctypes.py", line 108, in Value
    lock = RLock()
  File "/snap/git-repo/18/usr/lib/python2.7/multiprocessing/__init__.py", line 183, in RLock
    return RLock()
  File "/snap/git-repo/18/usr/lib/python2.7/multiprocessing/synchronize.py", line 172, in __init__
    SemLock.__init__(self, RECURSIVE_MUTEX, 1, 1)
  File "/snap/git-repo/18/usr/lib/python2.7/multiprocessing/synchronize.py", line 75, in __init__
    sl = self._semlock = _multiprocessing.SemLock(kind, value, maxvalue)
OSError: [Errno 13] Permission denied
```

Il nous a ensuite fallu modifier les droits du dossier android-kernel afin de pouvoir procéder à un repo init. Pour cela, il s'agissait d'effectuer 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
```

Il nous était ensuite possible de procéder au repo init afin de mettre en place la configuration de l'environnement pour le noyau hikey-linaro. A noter qu'il ne fallait pas ajouter de sudo à cette commande.

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

Ubuntu demandait ensuite de rentrer nos identifiants Git.

```
osboxes@osboxes:~/android-kernel$ repo init -u https://android.googlesource.com/
/kernel/manifest -b hikey-linaro-android-4.19

Your identity is: Hawlink <cempaul.sidawy@live.fr>
If you want to change this, please re-run 'repo init' with --config-name

Testing colored output (for 'repo diff', 'repo status'):
  black    red    green  yellow blue  magenta cyan  white
  bold    dim    ul      reverse
Enable color display in this user account (y/N)? y

repo has been initialized in /home/osboxes/android-kernel
```

L'environnement étant installé, il était ensuite possible d'effectuer un **repo sync**.

### 3) Identification de la configuration par défaut relative à un noyau pour une carte ranchu64.

Dans un premier temps, il nous a fallu localiser la config ranchu. Cette dernière se trouvait dans le dossier **arch/mips/configs/generic**. Le fichier en question était nommé **board-ranchu.config**.

```
CONFIG_VIRT_BOARD_RANCHU=y
CONFIG_BATTERY_GOLDFISH=y
CONFIG_FB=y
CONFIG_FB_GOLDFISH=y
CONFIG_GOLDFISH=y
CONFIG_STAGING=y
CONFIG_GOLDFISH_AUDIO=y
CONFIG_GOLDFISH_PIC=y
CONFIG_GOLDFISH_PIPE=y
CONFIG_GOLDFISH_TTY=y
CONFIG_RTC_CLASS=y
CONFIG_RTC_DRV_GOLDFISH=y

CONFIG_INPUT_EVDEV=y
CONFIG_INPUT_KEYBOARD=y
CONFIG_KEYBOARD_GOLDFISH_EVENTS=y

CONFIG_MAGIC_SYSRQ=y
CONFIG_POWER_SUPPLY=y
CONFIG_POWER_RESET=y
CONFIG_POWER_RESET_SYSCON=y
CONFIG_POWER_RESET_SYSCON_POWEROFF=y

CONFIG_VIRTIO_BLK=y
```

```
CONFIG_VIRTIO_CONSOLE=y
CONFIG_VIRTIO_MMIO=y
CONFIG_VIRTIO_MMIO_CMDLINE_DEVICES=y
CONFIG_NETDEVICES=y
CONFIG_VIRTIO_NET=y
```

Nous souhaitons donc passer sur une configuration ARM64. Il s'agissait de copier la configuration ranchu dans le dossier **arch/arm64/configs**. La commande à effectuer était donc la suivante :

```
cp board-ranchu.config ~/android-kernel/hikey-linaro/arch/arm64/configs
```

Il s'agissait ensuite de fusionner la configuration ranchu avec une configuration arm64. En effet, la configuration arm64 par défaut du noyau hikey-linaro (defconfig) contient bien plus d'éléments que la simple configuration ranchu. La configuration ranchu ajoute principalement des éléments goldfish et certains éléments virtio.

Tout d'abord, nous souhaitons travailler avec une configuration 64 bits. Il s'agissait donc de taper dans la console la commande suivante :

```
export ARCH=arm64
```

GCC n'étant pas installé, il nous a fallu également l'installer afin de pouvoir compiler le makefile.

```
osboxes@osboxes:~/android-kernel/hikey-linaro$ sudo apt install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 gcc : Depends: gcc-9 (>= 9.3.0-3~) but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
```

Un apt-cache policy nous a permis de constater que gcc-9 n'est pas installé :

```
osboxes@osboxes:~/android-kernel/hikey-linaro$ apt-cache policy gcc-9
gcc-9:
  Installed: (none)
  Candidate: 9.3.0-10ubuntu2
  Version table:
     9.3.0-10ubuntu2 500
        500 http://us.archive.ubuntu.com/ubuntu focal/main amd64 Packages
```

Or en essayant d'installer gcc-9, nous avons obtenu des dépendances en chaîne non reconnues :

```

osboxes@osboxes:~/android-kernel/hikey-linaro$ sudo apt install gcc-9
Reading package lists... Done
Building dependency tree
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
gcc-9 : Depends: cpp-9 (= 9.3.0-10ubuntu2) but 9.3.0-17ubuntu1~20.04 is to be
installed
        Depends: gcc-9-base (= 9.3.0-10ubuntu2) but 9.3.0-17ubuntu1~20.04 is t
o be installed
        Depends: libgcc-9-dev (= 9.3.0-10ubuntu2) but it is not going to be in
stalled
        Recommends: libc6-dev (>= 2.13-0ubuntu6) but it is not going to be ins
talled
E: Unable to correct problems, you have held broken packages.

```

Un **sudo apt-get update** a pu résoudre le problème pour installer GCC.

Lors de la tentative de compilation du Makefile avec la configuration par défaut, nous obtenions l'erreur suivante :

```

osboxes@osboxes:~/android-kernel/hikey-linaro$ make
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
YACC scripts/kconfig/zconf.tab.c
/bin/sh: 1: bison: not found
make[2]: *** [scripts/Makefile.lib:196: scripts/kconfig/zconf.tab.c] Error 127
make[1]: *** [Makefile:538: syncconfig] Error 2
Makefile:631: include/config/auto.conf.cmd: No such file or directory
make: *** [Makefile:643: include/config/auto.conf.cmd] Error 2

```

Il a fallu installer bison à l'aide d'un **sudo apt-get install bison** ainsi que flex à l'aide d'un **sudo apt-get install flex**.

#### 4) Script d'automatisation des commandes

Le script pourra être retrouvé sur le dépôt Git.

```

# Script file

# Pre-config

git config --global user.name = "User"
git config --global user.email = "user@mail.com"
sudo apt-get update sudo apt install python

sudo apt install gcc
sudo apt-get install bison
sudo apt-get install flex

```

```
# Installing repo and cloning repository

mkdir -p ~/.bin
PATH="${HOME}/.bin:${PATH}"
curl https://storage.googleapis.com/git-repo-downloads/repo%3E ~/.bin/repo
chmod a+rx ~/.bin/repo

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

# Configure kernel for a ranchu64 card

cp ~/android-kernel/hikey-linaro/arch/mips/configs/generic/board-ranchu.config ~/android-kernel/hikey-linaro/arch/arm64/configs

export ARCH=arm64
```

### 5) Modification de la configuration

L'utilisation de make oldconfig permet de lancer un éditeur de configuration en ligne de commandes pour éditer le fichier .config. Il s'agissait donc d'activer/de désactiver les options et quelques fois d'entrer des valeurs numériques (nombre de CPU, taille des pages ...).

Pour rendre la configuration compatible avec la carte ARMv8 Versatile, Qualcomm et Realtek, nous avons rentré « y » pour les lignes de configuration suivantes dans la catégorie « Platform Selection » :

```
Qualcomm Platforms (ARCH_QCOM) [N/y/?] (NEW) y
Realtek Platforms (ARCH_REALTEK) [N/y/?] (NEW) y
```

```
ARMv8 software model (Versatile Express) (ARCH_VEXPRESS) [N/y/?] (NEW) y
```

Pour activer l'option Frequency Scaling de la CPU, il s'agissait de valider les options de la catégorie « CPU Frequency scaling ».

```

*
* CPU Frequency scaling
*
CPU Frequency scaling (CPU_FREQ) [Y/n/?] y
CPU frequency transition statistics (CPU_FREQ_STAT) [Y/n/?] y
CPU frequency time-in-state statistics (CPU_FREQ_TIMES) [N/y/?] (NEW) y
Default CPUFreq governor
> 1. performance (CPU_FREQ_DEFAULT_GOV_PERFORMANCE)
   2. powersave (CPU_FREQ_DEFAULT_GOV_POWERSAVE)
   3. userspace (CPU_FREQ_DEFAULT_GOV_USERSPACE)
   4. ondemand (CPU_FREQ_DEFAULT_GOV_ONDEMAND)
   5. conservative (CPU_FREQ_DEFAULT_GOV_CONSERVATIVE)
   6. schedutil (CPU_FREQ_DEFAULT_GOV_SCHEDUTIL)
choice[1-6?]: 1
'performance' governor (CPU_FREQ_GOV_PERFORMANCE) [Y/?] y
'powersave' governor (CPU_FREQ_GOV_POWERSAVE) [Y/n/m/?] y
'userspace' governor for userspace frequency scaling (CPU_FREQ_GOV_USERSPACE)
[Y/n/m/?] y
'ondemand' cpufreq policy governor (CPU_FREQ_GOV_ONDEMAND) [Y/n/m/?] y
'conservative' cpufreq governor (CPU_FREQ_GOV_CONSERVATIVE) [Y/n/m/?] y
'schedutil' cpufreq policy governor (CPU_FREQ_GOV_SCHEDUTIL) [Y/n/?] y
*
* CPU frequency scaling drivers

```

Pour l'activation du support de l'HDMI CEC, il nous a fallu activer l'option suivante dans la catégorie « Graphics Support » :

```
Enable DisplayPort CEC-Tunneling-over-AUX HDMI support (DRM_DP_CEC) [Y/n/?] y
```

Concernant le support LED, il était nécessaire d'activer les options correspondantes à la catégorie éponyme un peu plus bas.

```

*
* LED Support
*
LED Support (NEW_LEDS) [Y/?] y
LED Class Support (LEDS_CLASS) [Y/m/?] y

```

#### 6) Génération d'une configuration par défaut

Pour générer la configuration par défaut, il était nécessaire de commencer par taper la commande suivante :

```
make savedefconfig
```

Ce afin de créer le fichier de configuration. Enfin, nous avons entré la commande suivante :

```
mv defconfig arch/arm64/configs/configTP2_LO52
```

