



04/12/2020

LO52

# Rapport de travaux pratiques

Tatiana GASSER & Loïc MARTIN

## Table des matières

Table des matières .....	1
1 TP1 : Mise en place de l'environnement de développement .....	3
1.1 Présentation du fonctionnement des TPs et du projet .....	3
1.2 Prise de contact pour la création du groupe .....	3
1.3 Mise en place du GIT .....	3
1.4 Création d'une branche .....	3
1.4.1 Commandes .....	3
1.4.2 Récupération de la branche pour la deuxième personne .....	3
1.5 Installation d'Android studio .....	3
1.6 Ecriture du code .....	4
1.7 Résultat .....	4
2 TP 2 : Manipulation d'un Kernel Linux .....	5
2.1 Téléchargement d'une VM Linux .....	5
2.2 Ajout de repo sur la VM Linux .....	5
2.3 Enlever snap git repo .....	5
2.4 Réinstallation depuis le site source .....	5
2.5 Créer un dossier et se positionne dedans .....	6
2.6 Cloner le repo avec la bonne branche .....	6
2.7 Synchroniser le repo .....	6
2.8 Pour comparer les deux branches .....	6
2.9 Android Hikey Linaro .....	6
2.10 Modification du dossier de config .....	7
2.11 Création d'un script .....	7
2.11.1 Création d'un dossier pour le TP : .....	7
2.11.2 Initialisation du dossier repo et la branche hikey-linaro .....	7
2.11.3 Téléchargement .....	7
2.11.4 Copie du fichier .....	7
2.11.5 Activation du fichier config .....	8
2.11.6 Modification de la config .....	8
3 TP 3 : Création d'un device Android .....	9
3.1 Ecrire des Makefile Android relatifs à l'intégration d'un composant et d'un produit Android	9
3.2 Implémentation de la libusb : .....	9
3.2.1 Compilation : .....	9
3.2.2 Erreur sur la macro TIMESPEC_TO_TIMEVAL .....	9

3.2.3	Deuxième erreur.....	9
3.3	Implémentation d'un nouveau produit Android.....	10
3.3.1	Surcharge du fichier sym_keyboard_delete.png.....	10
4	TP 4 : Utilisation de JNI.....	11
4.1	Création du projet .....	11
4.2	Téléchargement du package NDK .....	11
4.3	Modification du fichier native-lib.cpp avec la fonction native.....	11
4.4	Création d'un fichier java avec les éléments graphiques à relier.....	11
4.5	Résultats .....	12

## 1 TP1 : Mise en place de l'environnement de développement

### 1.1 Présentation du fonctionnement des TPs et du projet

### 1.2 Prise de contact pour la création du groupe

### 1.3 Mise en place du GIT

```
git clone
```

### 1.4 Création d'une branche

#### 1.4.1 Commandes

```
git branch "NomPrenom_NomPrenom"
```

```
git checkout "Nom de la branche"
```

```
git push -u
```

Attente de la validation de la branche par le responsable du REPO GIT

#### 1.4.2 Récupération de la branche pour la deuxième personne

```
git branch //Vérification du positionnement sur la branche Master
```

```
git pull
```

```
git checkout GASSERTatiana_MARTINLoic
```

### 1.5 Installation d'Android studio

Choix de la version 8.0.0 pour Android pour respecter la version de notre téléphone.

Choix du langage de programmation : JAVA.

Prise en main de la documentation Android : <https://developer.android.com/docs/>

Téléchargement d'une version pour l'émulation.

## 1.6 Ecriture du code

Création de l'activité principale Bouton et l'activité Display Message dans le manifeste.

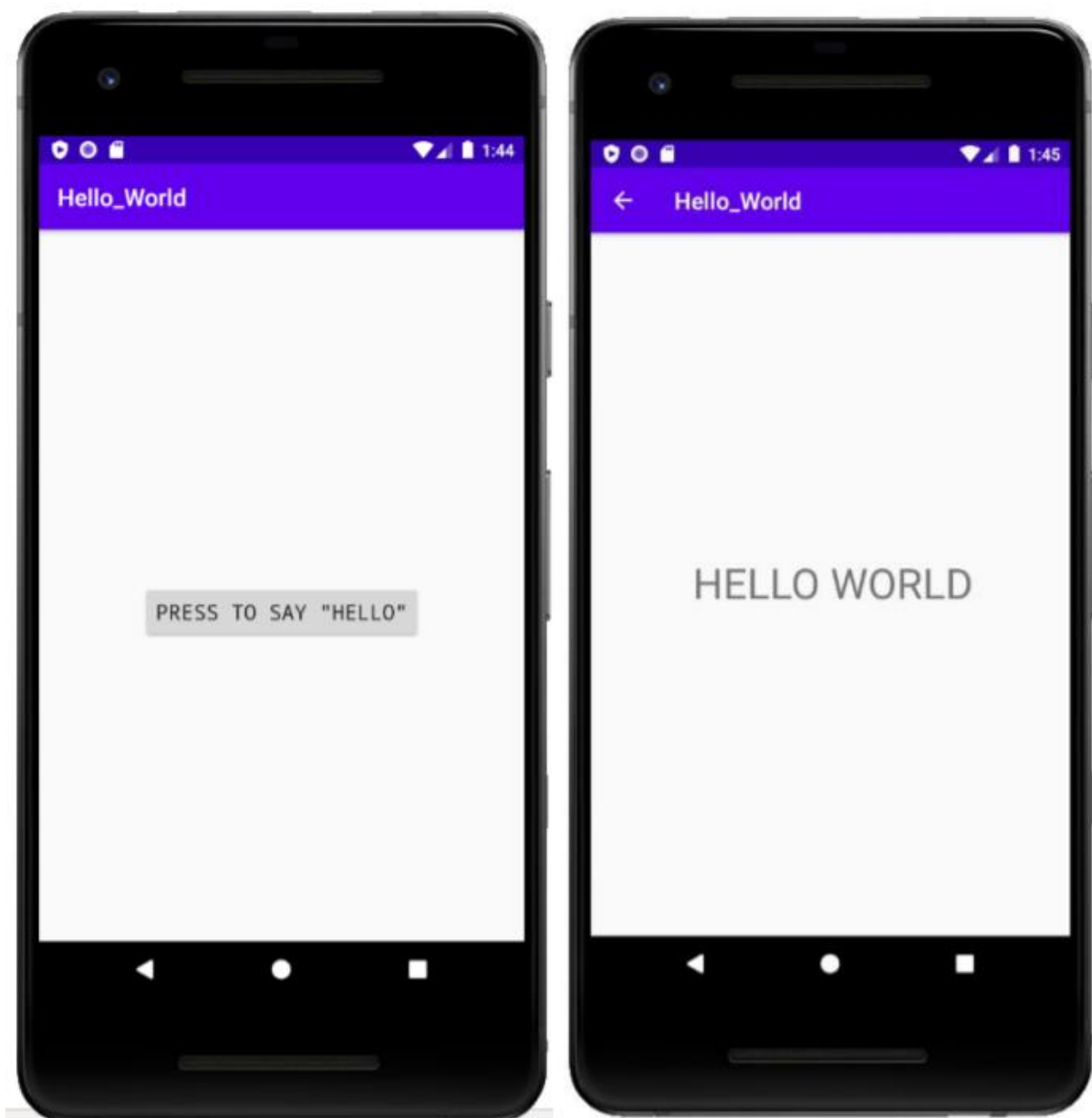
Utilisation du Layout Editor.

Création des xml pour le Layout.

Liaison du bouton à une fonction qui lance notre seconde activité.

Création de la classe DisplayMessageActivity.

## 1.7 Résultat



## 2 TP 2 : Manipulation d'un Kernel Linux

### 2.1 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
```

### 2.2 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.

### 2.3 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.

### 2.4 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.

On a donc 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.

## 2.5 Créer un dossier et se positionne dedans

```
mkdir android-kernel && cd android-kernel
```

## 2.6 Cloner le repo avec la bonne branche

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

## 2.7 Synchroniser le repo

```
repo sync
```

## 2.8 Pour comparer les deux branches

Noyau DL -> config

Identifier la config défaut -> charger

Faire un « diff » entre les deux

## 2.9 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
```

## 2.10 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
```

## 2.11 Création d'un script

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

### 2.11.1 Création d'un dossier pour le TP :

```
mkdir android-kernel && cd android-kernel
```

### 2.11.2 Initialisation du dossier repo et la branche hikey-linaro

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

### 2.11.3 Téléchargement

```
repo sync
```

### 2.11.4 Copie du fichier

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



#### 2.11.5 Activation du fichier config

```
make board-ranchu.config  
repo diff
```

#### 2.11.6 Modification de la config

On a modifié la config avec les caractéristiques. Voir fichier « newconf\_defconfig ».

### 3 TP 3 : Création d'un device Android

#### 3.1 Ecrire des Makefile Android relatifs à l'intégration d'un composant et d'un produit Android

#### 3.2 Implémentation de la libusb :

Les fichiers sources et les headers pour la compilation d'une libusb dans le Android.mk

```
LOCAL_PATH:= $(call my-dir)

include $(CLEAR_VARS)

# nom du module

LOCAL_MODULE := libusb

# .h

LOCAL_C_INCLUDES += $(LOCAL_PATH)/include $(LOCAL_PATH)/os

# Sources à compiler

LOCAL_SRC_FILES := core.c descriptor.c io.c sync.c

include $(BUILD_SHARED_LIBRARY)
```

##### 3.2.1 Compilation :

On compile en exécutant la commande :

```
sudo make
```

##### 3.2.2 Erreur sur la macro TIMESPEC\_TO\_TIMEVAL

Une erreur se produit sur la macro TIMESPEC\_TO\_TIMEVAL. Correction dans io.c :

```
#define TIMESPEC_TO_TIMEVAL(tv,ts)

{

    (tv)->tv_sec = (ts)->tv_sec;

    (tv)->tv_usec = (ts)->tv_nsec/1000;

}
```

##### 3.2.3 Deuxième erreur

build/tools/apriori/prelinkmap.c(137) : library "libusb.so" n'est pas en prelink map :

On ajoute la map libusb.so trouvée dans build/core/prelink-linux-arm.map

libusb.so 0xA8000000

### 3.3 Implémentation d'un nouveau produit Android

Le produit hérite du produit hikey de Linaro

```
$(call inherit-product, device/Linaro/hikey.mk)

PRODUCT_PACKAGES+= libusb

PRODUCT_PROPERTY_OVERRIDES := \
    ro.hw=lo_52 \
    net.dns1=8.8.8.8 \
    net.dns2=4.4.4.4

PRODUCT_PACKAGE_OVERLAYS := overlay/

PRODUCT_NAME := lo52_GASSERTatiana_MARINLoic

PRODUCT_BRAND := lo52

PRODUCT_DEVICE := lo52_GASSERTatiana_MARINLoic

PRODUCT_MODEL := GASSERTatiana_MARINLoic
```

#### 3.3.1 Surcharge du fichier sym\_keyboard\_delete.png

On surcharge le fichier sym\_keyboard\_delete.png

```
device/overlay/samples/SoftKeyboard/res/drawable-mdpi/sym_keyboard_delete.png
```

## 4 TP 4 : Utilisation de JNI

### 4.1 Création du projet

Ouverture d'Android studio

Création d'un nouveau projet en native C++

Sélection du standard C++ 17

### 4.2 Téléchargement du package NDK

Après une erreur lors du chargement, Android studio propose de télécharger le nécessaire pour le NDK

### 4.3 Modification du fichier native-lib.cpp avec la fonction native

### 4.4 Création d'un fichier java avec les éléments graphiques à relier

## 4.5 Résultats

