



# Département Informatique

## LO51 Semestre A19

### Rapport de TP

**Auteur**

Alban PEZZOLI  
Anthony BOUTEILLER

**Encadrant**

Fabien BRISSET

# Table des matières

---

<b>TP 1 : Environnement de développement Android .....</b>	<b>3</b>
<b>TP 2 : Manipulation Kernel Linux .....</b>	<b>3</b>
<b>TP 3 : Création d'un device Android .....</b>	<b>3</b>
<b>TP 4 : Utilisation JNI.....</b>	<b>4</b>

## TP 1 : Environnement de développement Android

---

### 1. Principe

La mise en place de l'environnement de développement Android pour l'UV LO52 était le but de ce TP. Elle se décompose en deux parties et une mise en pratique avec une première application. En premier lieu, nous organiserons la branche Git, ensuite nous installerons Android Studio.

### 2. Réalisation

Nous avons créé une branche Equal depuis master pour réaliser nos TPs et notre projet. Pour cela, nous avons utilisé **Tower** et **SourceTree** comme utilitaire graphique pour Git.

Ensuite, nous n'avons pas eu le besoin de configurer Android Studio. En effet, nous avons tous deux déjà eu l'occasion de développer des applications Android.

Pour finir, nous avons réalisé une application simple pour vérifier que l'environnement avait été correctement configuré.

### 3. Conclusion

Le but du TP étant de se familiariser avec l'environnement Android. Étant donné que nous avons déjà eu une expérience dans le développement Android, nous n'avons pas eu de difficultés.

## TP 2 : Manipulation Kernel Linux

---

Impossible à faire, DSI fautive ?

## TP 3 : Création d'un device Android

---

### 1. Principe

Le principe sera de comprendre et d'intégrer le composant libusb dans un device Android qui sera à créer.

### 2. Réalisation

Nous avons dans un premier temps merge sur notre branche Equal les fichiers sources et les headers nécessaire à l'implémentation de libusb.

Nous avons ensuite cherché à écrire le fichier Android.mk, dans un premier temps avec pour challenge de ne pas utiliser la variable LOCAL\_PATH pour inclure les différents fichiers source et les headers, utilisant à la place le chemin depuis le /external/libusb/

On a par la suite résolu le problème de la macro non définie `TIMESPEC_TO_TIMEVAL` en la redefinissant ainsi que le problème de `libusb.so` non pré-liée en la liant à la variable `LOCAL_SHARED_LIBRARIES`.

Pour finir, on a implémenté un nouveau produit Android en définissant son nom, en le faisant hérité du produit `hikey` de Linaro puis en personnalisant ses propriétés via la variable `PRODUCT_PROPERTY_OVERRIDES`.

### 3. Conclusion

Le TP3 permet de rentrer dans la partie noyau d'Android. La création d'un appareil Android et l'intégration d'un composant simple permet de comprendre et de se familiariser avec les `makefiles`.

## TP 4 : Utilisation JNI

---

### 1. Principe

Durant ce TP nous devons créer une application avec le NDK. De ce fait, nous pourrons utiliser le processus de JNI et implémenter des fonctions en C.

### 2. Réalisation

Pour réaliser cette application, nous devons premièrement créer une application native C++. Ensuite dans le fichier `native-lib.cpp`, nous rajoutons nos fonctions `read`, `write` et `direction` qui seront à implémenter en natif.

Pour finir, nous avons juste à les rendre visible depuis notre activité en kotlin avec le mot clé « `external` » pour pouvoir les utiliser dans notre code

### 3. Conclusion

Le TP nous a permis de mettre en pratique le mécanisme d'appel entre Java et le natif avec JNI avec une application simple.