|  |  |  |
| --- | --- | --- |
| REPUBLIQUE DU CAMEROUN  Paix – Travail – Patrie  \*\*\*\*\*\*\*  UNIVERSITE DE NGAOUNDERE  \*\*\*\*\*\*\*  FACULTE DES SCIENCES  \*\*\*\*\*\*\*  DÉPARTEMENT DE MATHEMATIQUE ET INFORMATIQUE  \*\*\*\*\*\*\*  BP 454 Ngaoundéré |  | REPUBLIC OF CAMEROON  Peace-Work-Fatherland  \*\*\*\*\*\*  THE UNIVERSITY OF NGAOUNDERE  \*\*\*\*\*\*\*  FACULTY OF SCIENCES  \*\*\*\*\*\*\*  DEPARTMENT OF INFORMATIC AND MATHEMATICAL  \*\*\*\*\*\*\*  PO Box P 454 Ngaoundéré |

**CYCLE:** I

**NIVEAU:** III

**OPTION: INFORMATIQUE**

**Unité d’enseignement : SYSTEME D’ESPLOITATION MOBILE**

**Rapport des travaux pratiques**

***Liste des membres du groupe de TALA***

|  |  |  |
| --- | --- | --- |
| REPUBLIQUE DU CAMEROUN  Paix – Travail – Patrie  \*\*\*\*\*\*\*  UNIVERSITE DE NGAOUNDERE  \*\*\*\*\*\*\*  FACULTE DES SCIENCES  \*\*\*\*\*\*\*  DÉPARTEMENT DE MATHEMATIQUE ET INFORMATIQUE  \*\*\*\*\*\*\*  BP 454 Ngaoundéré |  | REPUBLIC OF CAMEROON  Peace-Work-Fatherland  \*\*\*\*\*\*  THE UNIVERSITY OF NGAOUNDERE  \*\*\*\*\*\*\*  FACULTY OF SCIENCES  \*\*\*\*\*\*\*  DEPARTMENT OF INFORMATIC AND MATHEMATICAL  \*\*\*\*\*\*\*  PO Box P 454 Ngaoundéré |

|  |  |  |
| --- | --- | --- |
| **NOMS** | **PRENOMS** | **MATRICUL** |
| **TALA SIMO** | **Cécile Aude** | **16B739FS** |
| **OUSSOUMANOU** | **Benoit** | **16B156FS** |
| **ZOUFANE YOUGOUDA** | **Emmanuel** | **16A014FS** |
| **WANPOO WANE** | **Pierre** | **15A470FS** |
| **SANDA GINO** | **Mohamed** |  |
| **TCHUIGANG** | **Deric** | **16A998FS** |
| **MADJIREBAYE** | **Nanra** | **15B179FS** |
| **ALHADJE MADJOU** | **Kome** | **17B517FS** |
| **SIABA NGUELIFACK** | **Dilane** | **16A970FS** |
| **TCHANG YAHBO** | **Onesinus** | **15A995FS** |
| **MAHAMAT HAROUM** | **Adoum** | **16A002FS** | |
| **DJEKORNOM** | **Josué** | **16B122FS** | |

**Sous l’encadrement de:** **Dr.-ing Franklin TCHAKOUNTE**

**Année académique 2018-2019**

Sommaire

[SUJET 2 3](#_Toc530927711)

[Titre : décompilation et compilation d’une application sous androïde 3](#_Toc530927712)

[1. Les objectifs 3](#_Toc530927713)

[2. Les méthodes utilisées 3](#_Toc530927714)

[3. Résultats obtenus lorsqu’on decompile un apk 4](#_Toc530927715)

[4. Méthodes utilisées Pour la recompilation d’un apk : 5](#_Toc530927716)

[5. Résultats obtenus : 5](#_Toc530927717)

[6. Méthodes utilisées et résultats obtenu Pour trouver le code source 6](#_Toc530927718)

[SUJET3 7](#_Toc530927719)

[Titre : Simulation d’une intercommunication entre deux applications Android via les Intents. 7](#_Toc530927720)

[1. Objectifs : 7](#_Toc530927721)

[2. Méthodes utilisées et résultats obtenus 7](#_Toc530927722)

[3. Inter Communication Via Intent Implicite 9](#_Toc530927723)

[4. Inter communication via intent explicite 10](#_Toc530927724)

[BIBLIOGRAPHIE 11](#_Toc530927725)

En vue d’une compréhension plus pratique du cours de Système d’Exploitation mobile, les travaux pratiques ont été assignés aux groupes des étudiants qui ont chacun pour responsable une fille. A cet effet, nous présentons notre rapport de travail que nous avons eu à faire :

# SUJET 2

## Titre : décompilation et compilation d’une application sous androïde

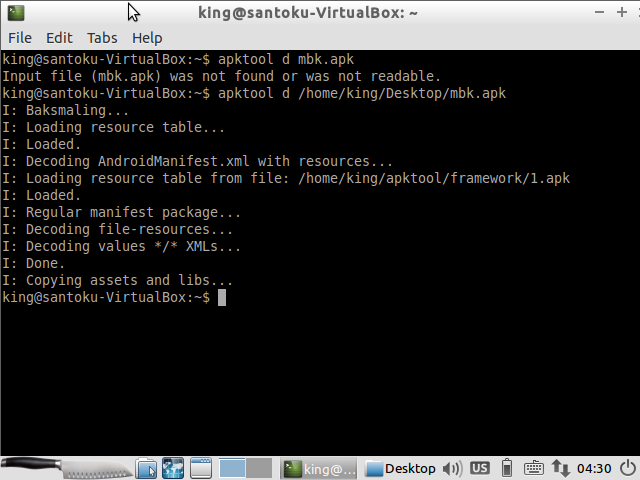
### Les objectifs

Ces travaux pratiques ont pour objectifs :

* Décompiler et recompiler une application androïde, retrouver le contenu du manifeste ;
* Retrouver le code source et les classes du code source d’une application ;
* Recompiler pour trouver l’application de départ

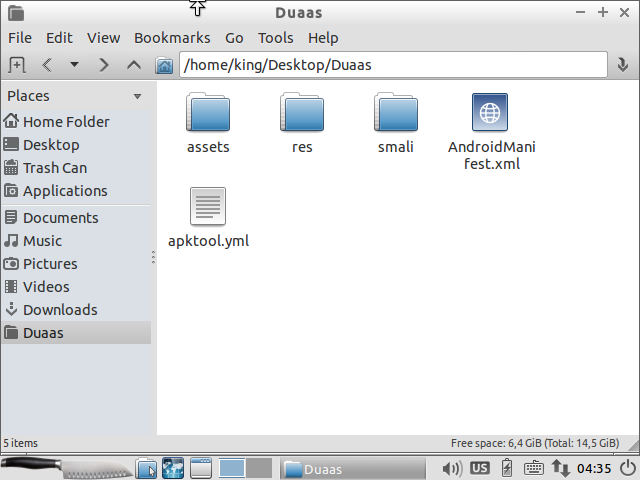
##### Les méthodes utilisées**Pour la décompilation d’un apk :**

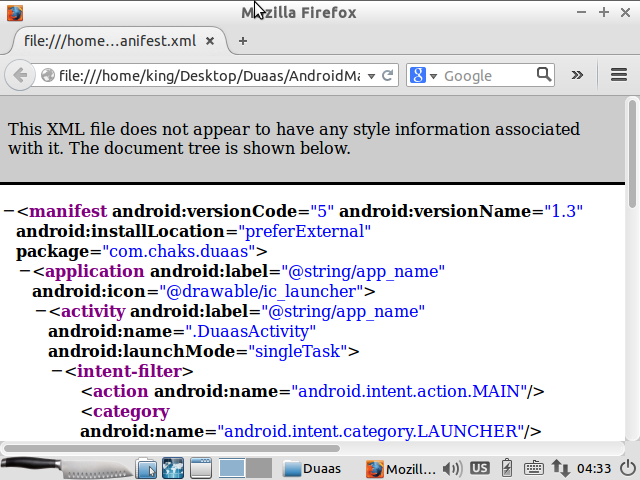
* Ouvrir le terminal de Santoku en utilisant la combinaison des touches Ctrl+Alt+T
* Saisir la commande **: apktool d –f /chemin d’accès au fichier/le nom de l’application suivi de l’extension .apk**
* Le message ‘’**Done**’’ confirme que la décompilation a été effectuée avec succès. s’il est absent cela signifie que soit l’apk n’est pas **Opensource** soit la commande a été mal saisie .



1. Résultats obtenus lorsqu’on decompile un apk

Une fois la décompilation effectuée, un répertoire se crée automatiquement, ayant pour nom celui de l’application décompilée, dans le gestionnaire de fichiers de la VM. Ce répertoire contient des dossiers tels que : **res** qui contient les ressources de l’application, **asset**, **smali** etc., et deux fichiers **AndroidManifest.xml et apktool.yml**.Le fichier AndroiManifest.xml contient les manifestes de l’application.



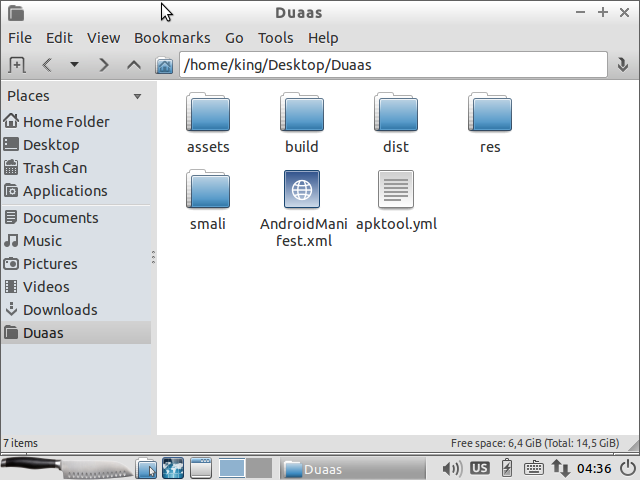
****

### Méthodes utilisées Pour la recompilation d’un apk :

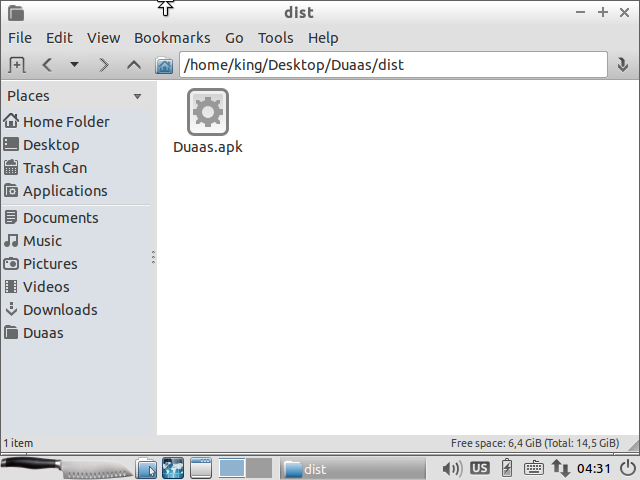
* Ouvrir le terminal de Santoku en utilisant la combinaison des touches **Ctrl+Alt+T**
* Saisir la commande **: apktool b /chemin d’accès au dossier**
* Le message ‘’**Building apk file…**’’ confirme que la recompilation a été effectuée avec succès.

### Résultats obtenus :

Apres la recompilation effectuée, nous retrouvons deux dossiers supplémentaires **build** et **dist**



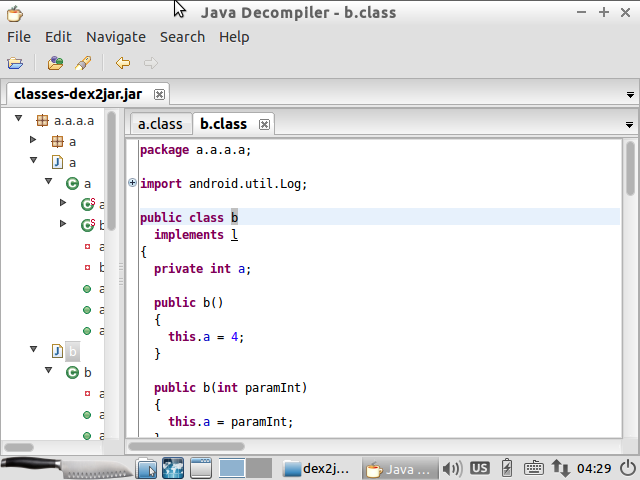
Alors le dossier dist qui contient l’application recompilée.

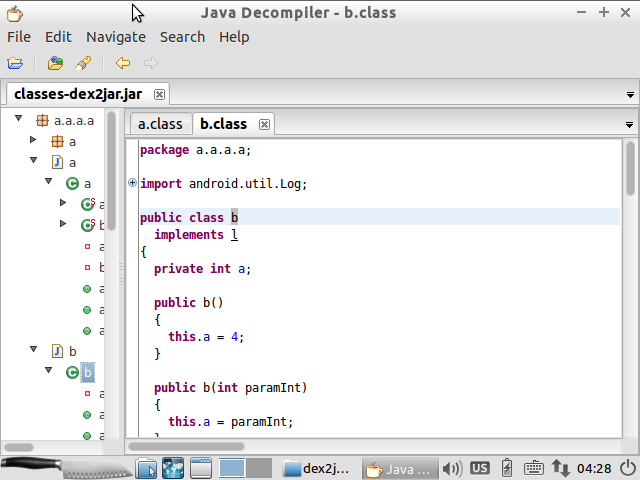


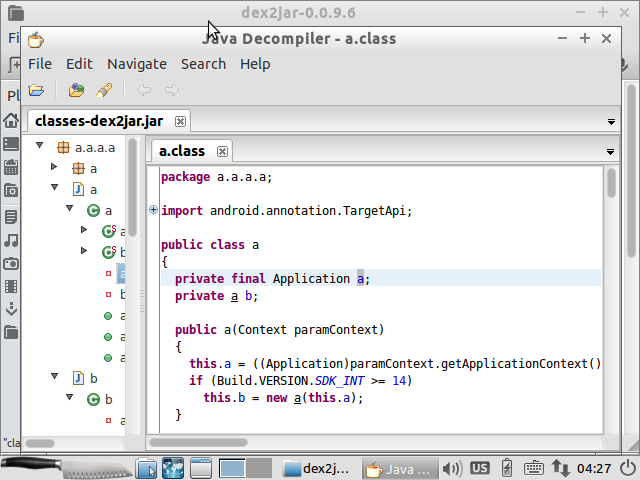
### Méthodes utilisées et résultats obtenu Pour trouver le code source

A partir de l’outil Dex2jar depuis le terminal Linux :

* On ouvre le fichier **.apk** à partir du gestionnaire d’archives, copier le fichier ‘**’classes.dx**’’ et le coller dans le dossier ‘’**dex2jar**’’ ;
* Ouvrir le terminal Linux(LxTerminal) et taper la commande « **cd/chemin/dex2jar** » où chemin est le chemin d’accès vers le dossier dex2jar sur lequel on veut le focus :
* Taper ensuite la commande «**d2j-dex2jar  classes.dex** » afin de vérifier les fichiers .dex et .jar ;
* Ouvrir le dossier ‘’**dex2jar**’’ et sélectionner le fichier .jar créée et l’ouvrir avec **JD-GUI**.On obtient les codes sources.







# SUJET3

## Titre : Simulation d’une intercommunication entre deux applications Android via les Intents.

### 1. Objectifs :

* Comprendre les spécificités des intents;
* Illustrations de l’inter communication avec les intents implicites et les intents explicites

### Méthodes utilisées et résultats obtenus

Le Système d’exploitation Mobile Androïde pour mesure de sécurité restreint les applications à des actions bien définis, il protège l’application dans son propre environnement. Ainsi, les applications utilisent les « ***intents*** » pour communiquer entre ses différents composants et d’autres applications.

Les ***intents*** sont en quelque sorte des objets messagers qui fournissent une certaine facilité à une application de communiquer entre ses différents composants ou des différentes applications.

* **LES TYPES D’INTENT DANS LE SYSTEME ANDROID**

Les intents contiennent des informations telles que : le **nom**, **composant**, **l’action à réaliser**, les **données**, **les données supplémentaires**, la **catégorie** et les **drapeaux**. En fonction de l’absence ou de la présence des données dans le champ on peut distinguer deux catégories d’intents à savoir : les **intents implicites** et les **intents explicites.**

Nous avons utilisés une application nommée ‘’Pro des Mots ‘’ Comme suite. Après avoir lancé l’application, nous entrons dans la page d’accueil où Plusieurs boutons nous sommes présentés (Messages, Paramètre ,Boutique, un gros bouton vert,…)

* ***Les intents implicites***

A l’inverse de l’intent explicite, l’intent implicite eux n’ont pas de destinataire prédéfini de façon précise, mais néanmoins d’autres champs doivent impérativement être remplis pour laisser à Android le soin de déterminer les applications capables d’exécuter cet intent et ce sera à l’utilisateur de choisir quelle applications il désire. Les champs essentiels obligatoires sont :

* **Action :** Ce que l’on désire que le destinataire fasse
* **Ensemble de données :** Sur quelles donnéesle destinataire doit effectuer son action (représenté sous forme d'URI)

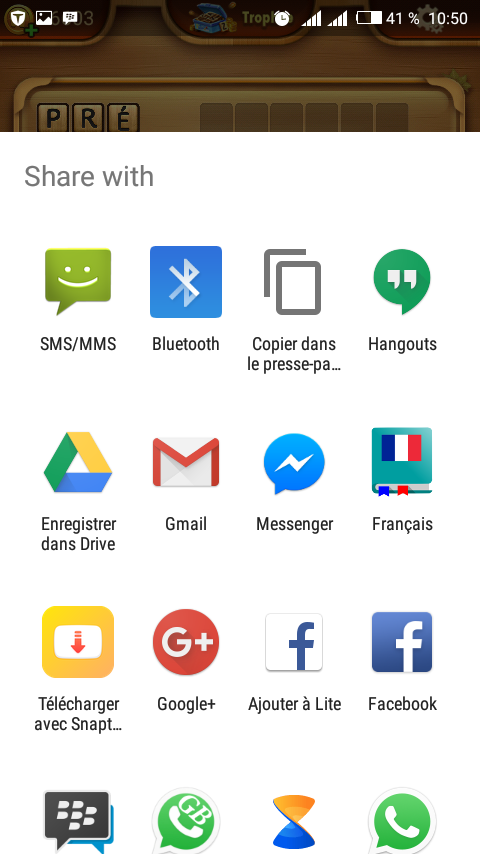
Les autres champs peuvent bien évidemment être remplis pour un renseignement beaucoup plus complet ce qui facilitera la proposition de liste d’applications par Android. Cet intent est majoritairement utilisé pour la communication d’une application à une autre et il permet de résoudre la difficulté pour le programmeur de connaitre à l’avance quelle application sera installée sur l’appareil de l’utilisateur.

### Inter Communication Via Intent Implicite

Lorsqu’on clique sur le gros bouton vert, une activité A s’ouvre contenant les mots à constituer

Juste à côté nous avons un bouton interroger un ami lorsqu’on appuie dessus, une activité B s’ouvre et nous présente les différentes applications (Facebook, twitter, WhatsApp, Sms, Email…) : c’est la manifestation d’un intent implicite.

L’application Pro des Mots qui s’ouvre et l’activité B comportant les applications (Facebook, WhatsApp, …) illustre l’inter communication entre l’application Pro des Mots et l’application Facebook par exemple. Comme le montre le ScreenShot suivant :



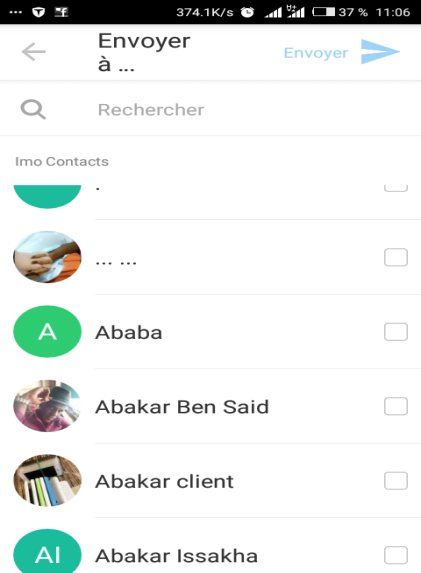
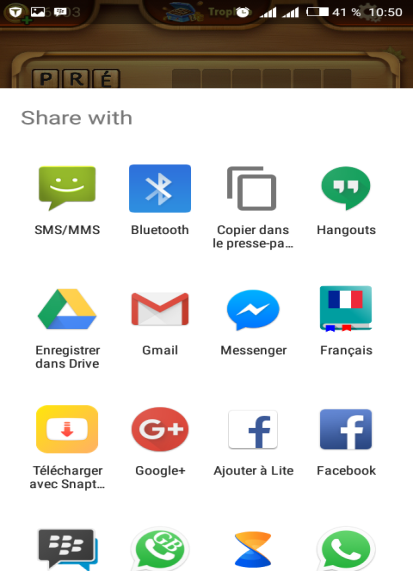
* ***Les intents explicites***

Pour qu’un intent soit dit « **explicite** », il faut que son champ composant soit renseigné, ce qui permet de définir le destinataire de l’intent, celui qui devra le gérer. Le champ composant est constitué de deux informations : le package où se situe le composant et le nom du composant. Ainsi donc, quand l’intent sera exécuté, androïde pourra retrouver le composant de destination de manière précise. En un mot, l’intent explicite est un intent ciblé. Cet intent est plus utilisé pour la communication entre les différents composants d'une même application puisque le programmeur connait d’ores et déjà les différentes classes.

### Inter communication via intent explicite

Lorsqu’on clique Le bouton partage, une activité A nous est présentée avec les applications (Facebook, WhatsApp, …). En sélectionnant Imo par exemple, une activité B s’ouvre ; ces deux interfaces A et B illustrent l’inter communication entre l’application Pro des Mots et Imo. Comme l’illustre-le

Screenshot suivant :



# BIBLIOGRAPHIE

* **ANDROID DEVELOPER/F.MICHEL** [**Frederic.Guinand@Univ-Lehavre.Fr**](mailto:Frederic.guinand@univ-lehavre.fr)
* **L’art Du Développement Android Par Mark L. Murphy**
* **Professional NFC Application Développement For Android By Vedat Coskun**
* **Android Developer Http://Developer.Android.Com/Index.Html**