

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Analýza inštalačných APK súborov pre OS Android

BAKALÁRSKA PRÁCA

Martin Styk

Brno, jar 2016

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Analýza inštalačných APK súborov pre OS Android

BAKALÁRSKA PRÁCA

Martin Styk

Brno, jar 2016

*Namiesto tejto stránky vložte kópiu oficiálneho podpísaného zadania práce a
prehlásenie autora školského diela.*

Prehlásenie

Prehlasujem, že táto bakalárska práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Martin Styk

Vedúci práce: Ing. Mgr. et Mgr. Zdeněk Říha, Ph.D.

Podakovanie

Rád by som sa poďakoval vedúcemu práce Ing. Mgr. et Mgr. Zdeňkovi Říhovi, Ph.D. za venovaný čas, ochotu a cenné pripomienky, ktoré mi pomohli pri tvorbe tejto práce.

Zhrnutie

Práca sa zaoberá získavaním metadát o inštalačných APK súboroch pre mobilný operačný systém Android. V rámci práce je vytvorená rozsiahla databáza APK balíčkov. Na základe analýzy týchto súborov sú určené štatistické vlastnosti APK súborov a príslušných aplikácií. Ako súčasť tejto práce je implementovaný nástroj na hromadné sťahovanie APK súborov, ich analýzu a výpočet štatistických dát nad množinou APK súborov. Práca sa zaoberá aj bezpečnosťou aplikácií a detekciou modifikovaných APK súborov. V práci je navrhnutá metóda detekcie upravených a prebalených APK balíčkov, ktorá je aj prakticky implementovaná. V teoretickej časti je popísaná štruktúra APK balíčkov a súborov v nich obsiahnutých.

Klíčové slová

APK súbory, Android, Apktool, malvér, analýza aplikácií, AndroidManifest.xml

Obsah

1	Úvod	1
2	APK súbory	3
2.1	Priečinok META-INF	3
2.2	Priečinok res	5
2.3	Priečinok lib	7
2.4	Priečinok assets	7
2.5	resources.arsc	8
2.6	classes.dex	8
2.7	AndroidManifest.xml	8
2.7.1	Element manifest	9
2.7.2	Element uses-permission	10
2.7.3	Element permission	11
2.7.4	Element uses-sdk	12
2.7.5	Element uses-feature	13
2.7.6	Element supports-screens	14
2.7.7	Element activity	14
2.7.8	Element service	15
2.7.9	Element provider	15
2.7.10	Element receiver	15
2.7.11	Element uses-library	16
3	These are	17
3.1	the available	17
3.1.1	sectioning commands.	17
4	Floats and references	19
5	Mathematical equations	21
6	We have several FONTS at disposal	23
7	Inserting the bibliography	25
	Literatúra	27
8	Inserting the index	29
	Register	29
A	An appendix	31

Zoznam tabuliek

- 4.1 A weather forecast 20
- A.1 Lokalizácia aplikácií 31
- A.2 Najpoužívanéjšie vlastnosti 32
- A.3 Najpoužívanéjšie prístupové oprávnenia 32
- A.4 Hodnoty najnižšej vyžadovanej verzie Android SDK 33
- A.5 Hodnoty cieľovej verzie Android SDK 33

Zoznam obrázkov

- 2.1 Typická štruktúra APK súboru 4
- 4.1 The logo of the Masaryk University at 40 mm 19
- 4.2 The logo of the Masaryk University at $\frac{2}{3}$ and $\frac{1}{3}$ of text width 20

1 Úvod

TBD

2 APK súbory

APK súbory sú balíčky používané operačným systémom Android. Celý názov ukrytý za skratkou APK je Android application package file. Tieto súbory slúžia na distribúciu aplikácií v operačnom systéme Android. Ich použitie a význam je analogický ako pri MSI balíčkoch používaných v systéme Microsoft Windows, alebo DEB balíčkoch používaných v niektorých linuxových distribúciách. APK súbory sú asociované s príponou *.apk* a príslušný MIME typom *application/vnd.android.package-archive* [1].

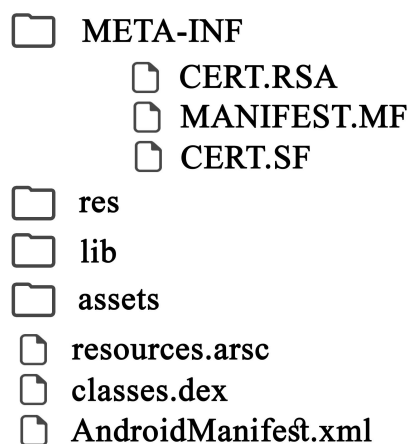
Štruktúra APK balíčkov vychádza z JAR¹ balíčkov – súborov používaných na distribúciu aplikácií alebo knižníc na platforme Java. Formát APK rozširuje všeobecnejší JAR formát o súbory, ktoré sú špecifické pre cieľovú platformu, ktorou je operačný systém Android. Zároveň si však ponecháva vlastnosti JAR súborov. APK balíčky sú archívne súbory v ZIP² formáte. Keďže APK používajú ZIP formát, k ich obsahu môžeme jednoducho prísť rozbalením archívu štandardným spôsobom. APK súbory vznikajú ako výstup kompletnej kompilácie a zabalenia aplikácií pre Android. APK súbor každej aplikácie obsahuje všetky potrebné súbory na jej inštaláciu a spustenie. Medzi týmito súbormi sa typicky nachádza *classes.dex* súbor obsahujúci skompilovaný zdrojový kód, *resources.arsc* súbor ktorý obsahuje skompilované zdroje aplikácie, súbor *AndroidManifest.xml* a neskompilované súbory ako sú napríklad obrázky [2].

2.1 Priečink METAINF

Priečink obsahujúci súbory, ktorých úlohou je zaručiť integritu ostatných súborov v APK balíčku a s ňou spojenú bezpečnosť celého systému. V prípade detekcie pozmenených súborov a narušenia integrity operačný systém Android nedovolí inštaláciu APK balíčku. Po každej zmene je nutné balíček digitálne podpísať.

1. Java archive

2. [https://en.wikipedia.org/wiki/Zip_\(file_format\)](https://en.wikipedia.org/wiki/Zip_(file_format))



Obr. 2.1: Typická štruktúra APK súboru

CERT.RSA

Súbor obsahujúci verejný kľúč ktorý slúži na overenie digitálneho podpisu balíčka.

MANIFEST.MF

Súbor obsahujúci relatívne cesty a SHA-1 hashe ³ všetkých súborov v APK balíčku. Tento súbor neobsahujú len APK súbory, je typický pre každý JAR archív.

Typický začiatok súboru *MANIFEST.MF* vyzerá nasledovne [3]:

```
Manifest-Version: 1.0
Built-By: 0.12.2
Created-By: Android Gradle 0.12.2
```

```
Name: res/drawable-xhdpi-v4/libraries.png
SHA1-Digest: Vvga01jpW3iS1nBBikD/urdbN58=
```

```
Name: res/layout/activity_settings.xml
```

3. reťazce v Base64 kódovaní

SHA1-Digest: 1coP1lt9Lmccc7SMZGHxNv4bbKs=

CERT.SF

Súbor podobný ako *MANIFEST.MF*, avšak namiesto SHA-1 hashov samotných súborov obsahuje SHA-1 hashe záznamov o týchto súboroch z *MANIFEST.MF*. Okrem toho obsahuje aj hash celého súboru *MANIFEST.MF*.

Záznam o jednom súbore v APK balíčku v súbore *CERT.SF* vyzerá nasledovne:

```
Name: res/drawable-xhdpi-v4/libraries.png  
SHA1-Digest: Slg56lqothjvmaBikD/urdb7q6=
```

Reťazec *Slg56lqothjvmaBikD/urdb7q6=* reprezentuje SHA-1 hash nasledujúceho záznamu zo súboru *MANIFEST.MF*:

```
"Name: res/drawable-xhdpi-v4/libraries.png  
SHA1-Digest: Vvga01jpW3iS1nBBikD/urdbN58=  
"
```

2.2 Priečínok res

Priečínok obsahujúci zdrojové súbory ako napríklad obrázky, zvuky alebo ikony. Okrem multimediálnych súborov obsahuje taktiež zdrojové XML súbory určujúce vzhľad obrazoviek, použité grafické štýly, alebo texty použité v aplikácii. Niektoré z týchto XML súborov môžu byť skompilované do binárneho formátu. V zdrojovom kóde sú tieto zdroje odkazované pomocou unikátnych identifikátorov. Identifikátory sú generované počas kompilácie nástrojom *aapt* a nachádzajú sa v projektovej triede *R*. Pre každý typ zdrojového súboru je generovaná podtrieda triedy *R* [4]. Všetky zdroje aplikácie by mali byť externalizované a uložené v špecifickom podpriečinku v tomto adresári.

Podporované podpriečinky [5]

2. APK SÚBORY

- animator - obsahuje XML súbory definujúce property animácie⁴
- anim - obsahuje XML súbory definujúce tween animácie⁵, môže obsahovať aj property animácie
- color - obsahuje XML súbory definujúce farby a ich zmeny na základe stavu objektov na ktoré sú aplikované
- drawable - obsahuje obrázky vo formáte PNG, 9.PNG, JPG, GIF alebo XML súbory skompilované do formy vykresliteľných obrázkov
- mipmap - ikony aplikácie s rôznou hustotou pixelov
- layout - obsahuje súbory vo formáte XML definujúce vzhľad rozmiestnenie prvkov na obrazovke
- menu - obsahuje súbory XML definujúce menu aplikácie
- raw - obsahuje súbory ktoré musia byť uložené a použité v neskomprimovanej forme a kvalite
- values - obsahuje súbory vo formáte XML definujúce hodnoty textových reťazcov, farieb, štýlov, základných rozmerov
- xml - obsahuje XML súbory, ktoré môžu byť načítané počas behu aplikácie

V spomenutých priečinkoch sú uložené základné zdroje aplikácie. Tieto zdrojové súbory určujú základný dizajn a obsah aplikácie. Avšak rôzne typy Android zariadení môžu využívať rôzne zdrojové súbory. Alternatívne zdroje sa využívajú na prispôbenie dizajnu a obsahu veľkosti a aktuálnej konfigurácií zariadenia. Sú umiestnené v priečinku, ktorého názov pozostáva z typu zdrojového súboru, ktorý korešponduje so základným názvom priečinku a názvu hodnoty konfiguračného atribútu pre ktorý je tento priečinok určený. Je možné kombinovať viacero konfiguračných atribútov [6].

4. Animácie definované pomocou zmeny atribútov vykreslovaných objektov

5. Animácie definované pomocou štartového bodu, koncového bodu, rotáciou a inými transformáciami vykreslovaného objektu

Najpoužívanéjšie atribúty

- Jazyk a región – jazyk je definovaný podľa ISO 639-1⁶ kódovania s možnosťou rozšírenia pomocou ISO 3166-1-alpha-2 regionálneho kódu. Napríklad obrázky špecifické pre zariadenia s francúzskym jazykom sa nachádzajú v priečinku *res/drawable-fr*
- Veľkosť obrazovky – v závislosti na veľkosti a rozlíšení obrazovky rozlišuje štyri možné hodnoty – *small, normal, large, xlarge*
- Orientácia obrazovky – v závislosti na orientácii zariadenia hodnota *port* pre zariadenie vo vertikálnej polohe, hodnota *land* pre polohu horizontálnu
- Hustota obrázkových bodov obrazovky - hodnoty určujúce vhodnosť zdrojových súborov vzhľadom na hustotu obrázkových bodov obrazovky daného zariadenia

2.3 Priečínok lib

Priečínok obsahujúci skompilovaný zdrojový kód natívnych knižníc. Tieto knižnice sú špecifické pre typ procesora. V závislosti na architektúre procesora obsahuje podpriečinky: *armeabi, armeabi-v7a, arm64-v8a, x86, x86_64, mips*.

2.4 Priečínok assets

Priečínok obsahujúci súbory uložené a používané v originálnej neskompimovanej forme. V tomto priečinku sa často nachádzajú textové súbory, html súbory, licenčné informácie, obrázky alebo textúry. Na rozdiel od priečinku *res/raw/*, zdrojovým súborom v umiestneným v priečinku *assets* nie sú pridelené unikátne identifikátory uložené v triede *R.java*. K súborom sa pristupuje ako k dátam uloženým v bežnom súborovom systéme. Trieda *AssetManager* poskytuje funkcionality na čítanie súborov ako prúdu bytov a navigovanie v tomto priečinku.

6. http://www.iso.org/iso/home/standards/language_codes.htm

2.5 resources.arsc

Súbor obsahujúci mapovanie medzi zdrojovými súbormi z priečinku *res* a ich identifikátormi. Vytvára sa počas kompilácie. Obsahuje XML súbory v binárnom formáte.

2.6 classes.dex

Classes.dex je súbor obsahujúci skompilovaný zdrojový kód aplikácie. Zdrojové súbory Android aplikácií sú napísané v jazyku Java. Java súbory sú skompilované do Java bytekódu pomocou bežného kompilátoru pre platformu Java. Výsledkom tejto kompilácie sú súbory s príponou *class*, ktoré sú následne preložené do Dalvik bytekódu pomocou nástroja *dx* ktorý je súčasťou *Android SDK* [7]. Výstupom nástroja *dx* je jediný súbor obsahujúci skompilovaný celý výkonný zdrojový kód aplikácie – *classes.dex*. Tento súbor je skomprimovanou a optimalizovanou verziou všetkých *class* súborov. Takto skompilovaný program môže byť vykonaný len vo virtuálnom stroji *Dalvik*, alebo v novšom prostredí *ART (Android Runtime)* používanom primárne od verzie *Android 5.0 Lollipop* [8].

2.7 AndroidManifest.xml

Súbor ktorý musí obsahovať každá Android aplikácia. Tento súbor poskytuje informácie o aplikácii operačnému systému Android. Neobsahuje žiadny výkonný kód. Definuje meno a verziu, ktoré slúžia ako unikátny identifikátor danej aplikácie. Popisuje všetky komponenty z ktorých sa aplikácia skladá, cesty k použitým knižniciam, minimálny vyžadovaný level Android API, oprávnenie vyžadované aplikáciu na prístup k chráneným častiam Android API a taktiež oprávnenia, ktoré sú vyžadované od iných komponent pri pokuse komunikovať s danou aplikáciou [9]. Súbor *AndroidManifest.xml*, ktorý nájdeme v APK balíčku je vo formáte binárneho XML súboru. Je ho však možné previesť do klasického čitateľného XML formátu.

Keďže *AndroidManifest.xml* je základným súborom poskytujúcim metadáta o Android aplikácii a APK súbore, informácie získané z tohto súboru tvoria veľkú časť štatistických dát zbieraných a vyhodno-

covaných v tejto práci. Preto sa detailnejšie pozrieme na jeho štruktúru a niektoré dôležité elementy, ktoré obsahuje.

Základná štruktúra XML súboru *AndroidManifest.xml* vyzerá nasledovne.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
    <uses-permission />
    <permission />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <application>
        <activity />
        <service/>
        <receiver/>
        <provider/>
        <uses-library />
    </application>
</manifest>
```

Kód 2.1: Základná štruktúra súboru AndroidManifest.xml

2.7.1 Element manifest

```
<manifest xmlns:android="URL"
    package="string"
    android:sharedUserId="string"
    android:sharedUserLabel="string resource"
    android:versionCode="integer"
    android:versionName="string"
    android:installLocation=["auto" | "internalOnly" |
        "preferExternal"] >
</manifest>
```

AndroidManifest.xml obsahuje element manifest ako koreňový prvok. Tento element je povinný a každý manifest ho obsahuje práve jeden.

Element manifest definuje atribúty [10]:

- `xmlns:android` – povinný atribút definujúci menný priestor
- `package` – nemenné meno balíku aplikácie, povinný atribút definujúci identitu aplikácie
- `android:sharedUserId` – identifikátor aplikácie zdieľaný s ostatnými aplikáciami za účelom vzájomnej komunikácie
- `android:sharedUserLabel` – čitateľná podoba `android:sharedUserId` identifikátoru
- `android:versionCode` – interná informácia o verzii aplikácie. Tento atribút je využívaný len na rozoznanie novších verzií od starších. Novšie aplikácie obsahujú vyššiu hodnotu
- `android:versionName` – informácia o verzii aplikácie prezentovaná užívateľom. Pre systém Android neposkytuje informáciu o verzii aplikácie, tú obsahuje atribút `android:versionCode`.
- `android:installLocation` – určuje miesto základné miesto inštalácie aplikácie. Pokiaľ je aplikáciu možné nainštalovať len do vnútornej pamäti zariadenia, obsahuje hodnotu `internalOnly`. Takáto aplikácia nemôže byť presunutá na externé pamäťové médium (typicky SD karta). Táto hodnota je základnou použitou možnosťou, ak tento atribút nie je definovaný. V prípade hodnoty `auto` sú aplikácie inštalované vo vnútornej pamäti, ale môžu byť presunuté do pamäti externej. Hodnota `preferExternal` zabezpečí, že systém sa pokúsi o inštaláciu na externé pamäťové médium. V prípade neúspechu sa použije interná pamäť

2.7.2 Element `uses-permission`

```
<uses-permission android:name="string"
    android:maxSdkVersion="integer" />
```

Prístup k niektorým dátam alebo častiam kódu je z dôvodu ochrany limitovaný. Android využíva princíp povolení⁷. Povolenia môžu byť definované samotnou aplikáciou, inou aplikáciou alebo systémom Android. Aplikácia, ktorá chce pristupovať k chráneným dátam alebo používať chránené časti kódu, musí pomocou tagu `uses-permissions` deklarovať vyžadované povolenia [9]. Prístupové povolenia sú aplikácii schválené užívateľom. Vo verzii Android 5.1 a starších, systém počas inštalácie oboznámi užívateľa so všetkými povoleniami, ktoré aplikácia vyžaduje. V prípade, že ich používateľ neschváli, aplikácia nebude nainštalovaná. Od verzie Android 6.0 užívateľ schvaľuje povolenia počas behu aplikácie.

Element `uses-permission` definuje atribúty [11]:

- `android:name` – definuje názov povolenia
- `android:maxSdkVersion` – najvyšší level Android API, pre ktorý je dané povolenie potrebné

2.7.3 Element `permission`

```
<permission android:description="string resource"
            android:icon="drawable resource"
            android:label="string resource"
            android:name="string"
            android:permissionGroup="string"
            android:protectionLevel=["normal"|"dangerous" |
                                   "signature" |
                                   "signatureOrSystem"] />
```

Definuje bezpečnostné povolenie, ktoré môže byť použité na obmedzenie prístupu ku komponente aplikácie. Toto povolenie je následne používané aplikáciami, ktoré vyžadujú prístup k chránenej časti danej aplikácie.

Najdôležitejšie atribúty definované v rámci elementu `permission` [12]:

7. angl. permissions

2. APK SÚBORY

- `android:name` – názov povolenia, aplikácie vyžadujúce dané povolenie uvádzajú túto hodnotu v atribúte `android:name` v tagu `uses-permission`. Názov musí byť unikátny a mal by dodržiavať konvencie pomenovávania jazyka Java.
- `android:permissionGroup` – priradí toto povolenie do skupiny povolení (angl. Permission group). Táto skupina povolení musí byť deklarovaná v rámci elementu `permission-group` v niektorých z nainštalovaných aplikácií. Slúži na logické zoskupenie významovo podobných oprávnení.
- `android:protectionLevel` – charakterizuje potenciálne riziko spojené s použitím daného povolenia. Táto hodnota určuje postup operačného systému pri rozhodovaní o udelení povolenia. Základnou hodnotou je „normal“, ktorá reprezentuje povolenia s nízkou mierou rizika. Tieto povolenia sú aplikácii automaticky schválené systémom Android počas inštalácie. Pre povolenia z zvýšenou mierou rizika je určená hodnota „dangerous“. Tieto povolenia typicky udeľujú aplikácii prístup k citlivým dátam alebo k prvkom Android zariadenia, ktoré môžu negatívne ovplyvniť jeho používanie. Pretože tento typ povolení prináša potenciálne riziko, systém ho nemôže udeliť automaticky, ale až po explicitnom súhlase používateľa. V prípade, že tento atribút obsahuje hodnotu „signature“, povolenie je udelené automaticky a to len aplikáciám, ktorých certifikát je rovnaký ako certifikát aplikácie definujúcej dané povolenie. Hodnota „signatureOrSystem“ rozširuje hodnotu „signature“ a povolenia sú udelené aj aplikáciám Android system image.

2.7.4 Element `uses-sdk`

```
<uses-sdk android:minSdkVersion="integer"  
          android:targetSdkVersion="integer"  
          android:maxSdkVersion="integer" />
```

Element vyjadrujúci kompatibilitu aplikácie s verziou Androidu pomocou čísla verzie Android API.

Atribúty definované elementom `uses-sdk` [13]:

- `android:minSdkVersion` – vyjadruje najnižší level API vyžadovaný aplikáciou. Pokiaľ je táto hodnota vyššia ako API úroveň zariadenia, systém zabráni inštalácií. Tento atribút by mala obsahovať každá aplikácia
- `android:targetSdkVersion` – informuje o úrovni API na ktorej bola aplikácia testovaná a pre ktorú je primárne určená. V prípade, že zariadenie podporuje vyššiu verziu Android API ako definuje spomínaný atribút, aplikácia môže byť spustená v móde kompatibility s touto verziou API
- `android:maxSdkVersion` – najvyššia úroveň API kompatibilná s aplikáciou. Z dôvodu spätnej kompatibility nie je používanie tohto atribútu doporučené. Tento atribút bol využívaný len do verzie Android 2.0.1

2.7.5 Element `uses-feature`

```
<uses-feature
  android:name="string"
  android:required=["true" | "false"] />
```

Deklaruje hardvérovú alebo softvérovú vlastnosť⁸ používanú aplikáciou. Tento element informuje externé entity o funkciách, ktoré aplikácia vyžaduje. Deklarované vlastnosti majú informačný charakter. Systém Android nekontroluje či zariadenie podporuje všetky vlastnosti deklarované aplikáciou. Tento element je využívaný službou Google Play na filtrovanie aplikácií vyhovujúcich danému zariadeniu, a preto by mala byť deklarovaná každá vyžadovaná vlastnosť.

Element obsahuje nasledujúce atribúty [14]:

- `android:name` – názov vyžadovanej vlastnosti
- `android:required` – určuje, či aplikácia vyžaduje danú vlastnosť pre korektné fungovanie. Pokiaľ obsahuje hodnotu `true`,

8. angl. feature

aplikácia nie je schopná korektne fungovať na zariadení nepodporujúcim danú vlastnosť

2.7.6 Element supports-screens

```
<supports-screens android:resizeable=["true" | "false"]
    android:smallScreens=["true" | "false"]
    android:normalScreens=["true" | "false"]
    android:largeScreens=["true" | "false"]
    android:xlargeScreens=["true" | "false"]
    android:anyDensity=["true" | "false"]
    android:requiresSmallestWidthDp="integer"
    android:compatibleWidthLimitDp="integer"
    android:largestWidthLimitDp="integer"/>
```

Špecifikuje podporované typy obrazoviek. V prípade inštalácie na zariadení s väčšou obrazovkou ako aplikácia podporuje, informuje systém o potrebe využitia módu obrazovej kompatibility.

Väčšina atribútov deklaruje podporované veľkosti obrazoviek. Ďalšie atribúty využívané v tejto práci sú [15]:

- android-resizeable – indikuje schopnosť aplikácie korektne sa prispôbiť rôznym veľkostiam obrazoviek
- android:anyDensity – indikuje či aplikácia obsahuje zdrojové súbory vhodné pre obrazovky s rôznou hustotou obrazových bodov

2.7.7 Element activity

Deklaruje aktivity. Aktivity sú základnými časťami aplikácie. Sú to triedy rozširujúce triedu *android.app.Activity*. Sú zamerané na jednotlivé prípady použitia aplikácie, implementujú časť grafického užívateľského rozhrania a zabezpečujú komunikáciu s užívateľom. Tieto triedy sú časťou zdrojového kódu a v skompilovanej forme sa nachádzajú v súbore *classes.dex*. Pri viacerých spustených aktivitách zohráva dôležitú úlohu životný cyklus aktivity [16]. Všetky aktivity musia byť

deklarované pomocou tagu `activity`. V prípade, že deklarované nie sú, systém ich bude ignorovať a nebudú spustené.

2.7.8 Element service

Deklaruje komponenty aplikácie typu služba⁹. Na rozdiel od aktivít, služby nemajú grafické používateľské rozhranie. Slúžia na implementáciu dlhodobých úloh na pozadí, ktoré môžu bežať aj v čase keď aplikácia nie je aktívna na popredí, alebo poskytujú funkcionality využívanú aplikáciou. Rozlišujeme dva typy služieb [17]. Služby typu `started` po spustení bežia pokým nedokončia svoju úlohu a ostatným komponentom nevracajú výsledok. Služby typu `bound` komunikujú s ostatnými komponentami pomocou modelu klient-server a sú ukončené keď pre nich neexistuje klient. Každá služba rozširuje základnú triedu `android.app.Service` a musí byť deklarovaná pomocou elementu `service`, inak bude ignorovaná.

2.7.9 Element provider

Deklaruje komponenty aplikácie ktorých úlohou je poskytovanie štruktúrovaného prístupu k dátam spravovaným aplikáciou. Poskytovatelia obsahu¹⁰ sú implementovaný ako podtriedy triedy `android.content.ContentProvider`. Využitie tejto komponenty je nutné len v prípade potreby zdieľania dát medzi viacerými aplikáciami. Operačný systém Android si ukladá referencie na jednotlivých poskytovateľov obsahu pomocou authority textového reťazca, ktorý je definovaný ako jeden z atribútov elementu `provider` [18]. Systém Android nerozpoznáva poskytovateľov obsahu nedefinovaných v `AndroidManifest.xml`.

2.7.10 Element receiver

Deklaruje komponentu aplikácie implementovanú ako podtriedu triedy `android.content.BroadcastReceiver`. Tieto komponenty umožňujú aplikácií prijímať a reagovať na informácie o zámere spustenia aktivity (`intent`) aj v čase, keď ostatné komponenty aplikácie nie sú spustené. Tieto informácie sú vysielané systémom alebo inou aplikáciou. Systém

9. angl. service

10. angl. content providers

2. APK SÚBORY

Android môžeme oboznámiť s existenciou komponent typu broadcast receiver pomocou tagu receiver alebo aj dynamicky pomocou volania *Context.registerReceiver()* v zdrojovom kóde aplikácie [19].

2.7.11 Element uses-library

```
<uses-library  
  android:name="string"  
  android:required=["true" | "false"] />
```

Špecifikuje zdieľanú knižnicu vyžadovanú aplikáciou. Tento element informuje systém o potrebe zahrnúť cestu k zdrojovému kódu knižnice medzi cesty v ktorých Dalvik Virtual Machine hľadá zdrojové súbory. V angličtine sú tieto cesty označované ako class path. Najpoužívanejšie android balíky ako napríklad android.app, android.content alebo android.view sú obsiahnuté v základnej knižnici, ktorá je automaticky pripojená ku každej aplikácii a nemusia byť deklarované týmto elementom. Balíčky, ktoré nie sú v základnej android knižnici, musia byť deklarované. Tento element ovplyvňuje inštaláciu aplikácie na konkrétnom zariadení a taktiež dostupnosť aplikácie v obchode Google Play.

Definuje atribúty [20]:

- android:name – názov knižnice, ktorý sa nachádza v dokumentácii príslušného použitého balíčku
- android:required – indikuje či aplikácia potrebuje knižnicu špecifikovanú atribútom android:name. V prípade hodnoty true aplikácia nie je schopná fungovať bez danej knižnice. Systém zamietne inštaláciu takejto aplikácie, pokiaľ zariadenie neobsahuje danú knižnicu. Ak je tento atribút nastavený na hodnotu false, aplikácia je schopná korektne fungovať aj bez danej knižnice

3 These are

3.1 the available

3.1.1 sectioning commands.

Paragraphs and

subparagraphs are available as well. Inside the text, you can also use unnumbered lists,

- such as
- this one
 - and they can be nested as well.
 - » You can even turn the bullets into something fancier,
 - § if you so desire.

Numbered lists are

1. very
 - (a) similar

and so are description lists:

Description list A list of terms with a description of each term

The spacing of these lists is geared towards paragraphs of text. For lists of words and phrases, the paralist package offers commands

- that
 - are
 - * better
 - suited
- 1. to
 - (a) this
 - i. kind of
 - A. content.

4 Floats and references

The logo of the Masaryk University is shown in Figure 4.1 and Figure 4.2 at pages 19 and 20. The weather forecast is shown in Table 4.1 at page 20. The following chapter is Chapter 5 and starts at page 21. Items 3, 3b, and 3(c)iv are starred in the following list:

1. some text
2. some other text
3. ★
 - (a) some text
 - (b) ★
 - (c) some other text
 - i. some text
 - ii. some other text
 - iii. yet another piece of text
 - iv. ★
 - (d) yet another piece of text
4. yet another piece of text

If your reference points to a place that has not yet been typeset, the `\ref` command will expand to `??` during the first run of `pdflatex thesis.tex` and a second run is going to be needed for the references to resolve. With online services – such as Overleaf – this is performed automatically.



Obr. 4.1: The logo of the Masaryk University at 40 mm



Obr. 4.2: The logo of the Masaryk University at $\frac{2}{3}$ and $\frac{1}{3}$ of text width

Day	Min Temp	Max Temp	Summary
Monday	13°C	21°C	A clear day with low wind and no adverse current advisories.
Tuesday	11°C	17°C	A trough of low pressure will come from the northwest.
Wednesday	10°C	21°C	Rain will spread to all parts during the morning.

Tabulka 4.1: A weather forecast

5 Mathematical equations

T_EX comes pre-packed with the ability to typeset inline equations, such as $e^{ix} = \cos x + i \sin x$, and display equations, such as

$$\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

L^AT_EX defines the automatically numbered equation environment:

$$\gamma Px = PAx = PAP^{-1}Px. \quad (5.1)$$

The package amsmath provides several additional environments that can be used to typeset complex equations:

1. An equation can be spread over multiple lines using the `multline` environment:

$$\begin{aligned} a + b + c + d + e + f + b + c + d + e + f + b + c + d + e + f \\ + f + g + h + i + j + k + l + m + n + o + p + q \end{aligned} \quad (5.2)$$

2. Several aligned equations can be typeset using the `align` environment:

$$a + b = c + d \quad (5.3)$$

$$u = v + w + x \quad (5.4)$$

$$i + j + k + l = m \quad (5.5)$$

3. The `alignat` environment is similar to `align`, but it doesn't insert horizontal spaces between the individual columns:

$$a + b + c + d \quad = 0 \quad (5.6)$$

$$e + f + g = 5 \quad (5.7)$$

4. Much like chapter, sections, tables, figures, or list items, equations – such as (5.8) and (My equation) – can also be labeled and referenced:

$$b_{11}x_1 + b_{12}x_2 + b_{13}x_3 \quad = y_1, \quad (5.8)$$

$$b_{21}x_1 + b_{22}x_2 \quad + b_{24}x_4 = y_2. \quad (\text{My equation})$$

5. The `gather` environment makes it possible to typeset several equations without any alignment:

$$\psi = \psi\psi, \tag{5.9}$$

$$\eta = \eta\eta\eta\eta\eta\eta, \tag{5.10}$$

$$\theta = \theta. \tag{5.11}$$

6. Several cases can be typeset using the `cases` environment:

$$|y| = \begin{cases} y & \text{if } z \geq 0, \\ -y & \text{otherwise.} \end{cases} \tag{5.12}$$

For the complete list of environments and commands, consult the `amsmath` package manual¹.

1. See <http://mirrors.ctan.org/macros/latex/required/amslatex/math/amslldoc.pdf>. The `\url` command is provided by the package `url`.

6 We have several FONTS *at disposal*

The serifed roman font is used for the main body of the text. *Italics are typically used to denote emphasis or quotations.* The teletype font is typically used for source code listings. The **bold**, SMALL-CAPS and sans-serif variants of the base roman font can be used to denote specific types of information.

We can also change the font size, although it is usually not necessary.

A wide variety of mathematical fonts is also available, such as:

$ABC, \mathcal{ABC}, \mathbf{ABC}, \text{ABC}, \textit{ABC}, \text{ABC}$

By loading the amsfonts packages, several additional fonts will become available:

$\mathfrak{ABC}, \mathbb{ABC}$

Many other mathematical fonts are available¹.

1. See <http://tex.stackexchange.com/a/58124/70941>.

7 Inserting the bibliography

After loading the `biblatex` package and linking a bibliography database file to the document using the `\addbibresource` command, you can start citing the entries. This is just dummy text [**inbook-full**] lightly sprinkled with citations [**incollection-full**]. Several sources can be cited at once [**whole-collection**, **manual-minimal**, **manual-full**]. **inbook-full** was written by **inbook-full** in **inbook-full**. We can also produce **inbook-full** or (**inbook-full**, **inbook-full**). The full bibliographic citation is: **inbook-full**. We can easily insert a bibliographic citation into the footnote¹.

The `\nocite` command will not generate any output, but it will insert its argument into the bibliography. The `\nocite{*}` command will insert all the records in the bibliography database file into the bibliography. Try uncommenting the command and watch the bibliography section come apart at the seams.

When typesetting the document for the first time, citing a work will expand to [**work**] and the `\printbibliography` command will produce no output. It is now necessary to generate the bibliography by running `biber thesis.bcf` from the command line and then by typesetting the document again twice. During the first run, the bibliography section and the citations will be typeset, and in the second run, the bibliography section will appear in the table of contents.

The `biber` command needs to be executed from within the directory, where the \LaTeX source file is located. In Windows, the command line can be opened in a directory by holding down the **[Shift]** key and by clicking the right mouse button while hovering the cursor over a directory. Select the **Open Command Window Here** option in the context menu that opens shortly afterwards.

With online services – such as Overleaf – all commands are executed automatically.

1. **inbook-full**.

Literatúra

1. Freed, Ned; Kucherawy, Murray; Baker, Mark; Hoehrmann, Bjoern. *Media Types* [online]. 2016 [visited on 2016-03-23]. Available from WWW: <http://www.iana.org/assignments/media-types/media-types.xhtml>.
2. *Building and Running Overview* [online]. 2016 [visited on 2016-03-23]. Available from WWW: <http://developer.android.com/tools/building/index.html>.
3. Yang, Herong. *META-INF Files - Digests, Signature and Certificate* [online]. 2015 [visited on 2016-03-24]. Available from WWW: <http://www.herongyang.com/Android/Project-META-INF-Files-Digest-Signature-and-Certificate.html>.
4. *Accessing Resources* [online]. 2015 [visited on 2016-03-24]. Available from WWW: <http://developer.android.com/guide/topics/resources/accessing-resources.html>.
5. *Providing Resources* [online]. 2015 [visited on 2016-03-24]. Available from WWW: <http://developer.android.com/guide/topics/resources/providing-resources.html>.
6. *Providing Alternative Resources* [online]. 2015 [visited on 2016-03-24]. Available from WWW: <http://developer.android.com/guide/topics/resources/providing-resources.html%5C#AlternativeResources>.
7. Reddy, Satheesh. *Android Application Build Process or Compilation Process* [online]. 2014 [visited on 2016-03-24]. Available from WWW: <http://www.c-sharpcorner.com/UploadFile/34ef56/android-application-build-process-or-compilation-process/>.
8. *ART and Dalvik* [online]. 2015 [visited on 2016-03-23]. Available from WWW: <https://source.android.com/devices/tech/dalvik/>.
9. *App Manifest* [online]. 2015 [visited on 2016-03-24]. Available from WWW: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>.
10. *Manifest element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/manifest-element.html>.
11. *Uses-permission element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/uses-permission-element.html>.

LITERATURA

12. *Permission element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/permission-element.html>.
13. *Uses-sdk element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>.
14. *Uses-feature element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/uses-feature-element.html>.
15. *Supports-screens element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/supports-screens-element.html>.
16. *Activity* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/reference/android/app/Activity.html>.
17. *Service* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/reference/android/app/Service.html>.
18. *ContentProvider* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/reference/android/content/ContentProvider.html>.
19. *Receiver element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/receiver-element.html>.
20. *Uses-library element* [online]. 2015 [visited on 2016-03-26]. Available from WWW: <http://developer.android.com/guide/topics/manifest/uses-library-element.html>.
21. Westenberg, Jimmy. *Gartner: Android and iOS dominate smartphone market with 98 percent marketshare* [online]. 2015 [visited on 2016-03-23]. Available from WWW: <http://www.androidauthority.com/android-ios-hold-98-percent-marketshare-656624/>.
22. Thomas, Daniel R.; Beresford, Alastair R.; Rice, Andrew. Security Metrics for the Android Ecosystem. *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices - SPSM '15*. 2015, pp. 87–98. Available also from WWW: <http://dl.acm.org/citation.cfm?doid=2808117.2808118>.

23. *Industry Leaders Announce Open Platform for Mobile Devices* [online]. 2007 [visited on 2016-03-23]. Available from WWW: http://www.openhandsetalliance.com/press_110507.html.
24. Rosoff, Matt. *Google's Biggest Acquisitions So Far, And What They Became* [online]. 2011 [visited on 2016-03-23]. Available from WWW: <http://www.gizmodo.com.au/2011/08/googles-16-biggest-acquisitions-so-far-and-what-happened-to-them/>.
25. Beavis, Gareth. *A complete history of Android* [online]. 2008 [visited on 2016-03-23]. Available from WWW: <http://www.techradar.com/news/phone-and-communications/mobile-phones/a-complete-history-of-android-470327>.
26. *An Overview of the Android Architecture* [online]. 2013 [visited on 2016-03-23]. Available from WWW: http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture.
27. Parmar, Ketan. *In Depth: Android Package Manager and Package Installer*. 2013. Available also from WWW: <https://dzone.com/articles/depth-android-package-manager>.
28. Elenkov, Nikolay. *Android security internals: an in-depth guide to android's security architecture*. San Francisco: No Starch Press, 2015. ISBN 978-1-59327-641-6.

8 Inserting the index

After using the `\makeindex` macro and loading the `makeidx` package that provides additional indexing commands, index entries can be created by issuing the `\index` command. It is possible to create ranged index entries, which will encompass a span of text. To insert complex typographic material – such as α or $\mathrm{T\!E\!X}$ – into the index, you need to specify a text string, which will determine how the entry will be sorted. It is also possible to create hierarchical entries.

After typesetting the document, it is necessary to generate the index by running

```
texindy -I latex -C utf8 -L <locale> thesis.idx
```

from the command line, where *<locale>* corresponds to the main locale of your thesis – such as `english`, and then typesetting the document again.

The `texindy` command needs to be executed from within the directory, where the $\mathrm{L\!A\!T\!E\!X}$ source file is located. In Windows, the command line can be opened in a directory by holding down the `[Shift]` key and by clicking the right mouse button while hovering the cursor over a directory. Select the `[Open Command Window Here]` option in the context menu that opens shortly afterwards.

With online services – such as Overleaf – the commands are executed automatically, although the locale may be erroneously detected, or the `makeindex` tool (which is only able to sort entries that contain digits and letters of the English alphabet) may be used instead of `texindy`. In either case, the index will be ill-sorted.

Register

α , 31

dummy text, 31

T_EX, 31

vehicles

 speed cars, 31

 trucks, 31

A An appendix

Kód	Jazyk	%
es	španielsky	61,7
de	nemecký	59,6
fr	francúzsky	59,4
ru	ruský	58,1
ja	japonský	57,6
it	talianky	57,4
ko	korejský	56,9
zh-rcn	čínsky (zjednodušený)	55,6
zh-rtw	čínsky (tradičný)	54,0
pt	portugalský	52,6

Tabuľka A.1: Lokalizácia aplikácií

Názov	%
android.hardware.camera	18,1
android.hardware.touchscreen	16,1
android.hardware.telephony	14,8
android.hardware.camera.autofocus	10,6
android.hardware.location.gps	10,2
android.hardware.location	8,8
android.hardware.wifi	8,4
android.hardware.location.network	7,0
android.hardware.bluetooth	6,6
android.hardware.touchscreen.multitouch	6,0

Tabuľka A.2: Najpoužívanéjšie vlastnosti

Názov	%
android.permission.internet	92,9
android.permission.access_network_state	87,9
android.permission.write_external_storage	75,2
android.permission.wake_lock	49,5
android.permission.read_phone_state	49,4
android.permission.access_wifi_state	44,7
android.permission.vibrate	43,6
android.permission.get_accounts	31,3
android.permission.receive_boot_completed	30,5
android.permission.vending.billing	27,1

Tabuľka A.3: Najpoužívanéjšie prístupové oprávnenia

Verzia Android SDK	%
9	21,3
8	18,4
7	14,2
14	10,5
10	8,1
4	7,0
3	5,6
15	3,7
5	3,7
11	2,1

Tabuľka A.4: Hodnoty najnižšej vyžadovanej verzie Android SDK

Verzia Android SDK	%
19	25,6
17	11,8
21	11,7
15	6,8
14	6,3
22	6,0
16	5,7
18	5,6
20	3,8
8	2,9

Tabuľka A.5: Hodnoty cieľovej verzie Android SDK