

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

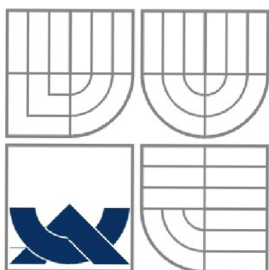
DÁLKOVÉ OVLÁDÁNÍ POČÍTAČE POMOCÍ  
MOBILNÍHO TELEFONU

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

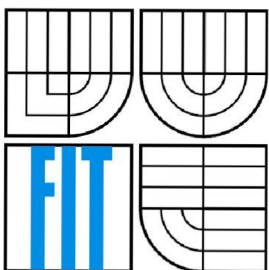
AUTOR PRÁCE  
AUTHOR

PROKOP ŘEHÁČEK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# DÁLKOVÉ OVLÁDÁNÍ POČÍTAČE POMOCÍ MOBILNÍHO TELEFONU

COMPUTER REMOTE CONTROL USING CELL PHONE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

PROKOP ŘEHÁČEK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. PETR POSPÍCHAL

BRNO 2011

## **Abstrakt**

Práce se zabývá tvorbou aplikace dálkového ovládání s využitím bezdrátové technologie Bluetooth. V práci je představen moderní operační systém Android a vývoj softwaru pro tento systém. Aplikace dálkového ovládání se skládá z částí: klient pro OS Android a server pro OS GNU/Linux. Pro implementaci klienta byl použit programovací jazyk Java a server vznikl za využití Qt toolkitu a jazyka C++. Vývoj programů je rozdělen na části: návrh, implementace a testování. Výsledkem práce je fungující aplikace, která ovládá multimediální programy pro přehrávání audio a video souborů.

## **Abstract**

This thesis deals with the creation of remote control application using Bluetooth wireless technology. The work introduces a modern Android operating system and software development for this system. Application of remote control consists of parts: client for Android OS and server for GNU / Linux. The client was implemented in Java and the server has been created using the Qt toolkit and C++. Software development is divided into parts: design, implementation and testing. The result of this thesis is software that can control multimedia programs for audio and video files.

## **Klíčová slova**

Dálkové ovládání, operační systém, Android, Linux, Qt, Bluetooth, komunikace, server, klient, socket, GUI, mobilní telefon

## **Keywords**

Remote control, operating system, Android, Linux, Qt, Bluetooth, communication, server, client, socket, GUI, cell phone

## **Citace**

Prokop Řeháček: Dálkové ovládání počítače pomocí mobilního telefonu, bakalářská práce, Brno, FIT VUT v Brně, 2011

# Dálkové ovládání počítače pomocí mobilního telefonu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Pospíchala. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Prokop Řeháček  
17.5. 2011

## Poděkování

Rád bych poděkoval Ing. Petru Pospíchalovi za odbornou pomoc, ochotu, čas a cenné rady. Také bych rád poděkoval svému bratrově MDDr. Adamu Řeháčkovi za pomoc při korekturách.

© Prokop Řeháček, 2011

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah.....	1
1 Úvod.....	3
2 Rozbor tématu vývoje mobilních aplikací a použitých technologií.....	4
2.1 Mobilní operační systémy.....	4
2.2 OS Android.....	4
2.2.1 Vývoj aplikací pro OS Android.....	5
2.3 Datové sítě.....	6
2.3.1 Bluetooth a BlueZ.....	6
2.3.2 Android Bluetooth API.....	7
2.4 Dálkové ovládání.....	8
3 Návrh Aplikace.....	10
3.1 Požadovaná funkčnost aplikace.....	10
3.1.1 Funkce klientské části na Android přístroji.....	10
3.1.2 Funkce serverové části na PC s OS GNU/Linux.....	11
3.2 Návrh systému dálkového ovládání.....	11
3.3 Návrh GUI.....	12
3.3.1 GUI dálkového ovládání.....	12
3.3.2 GUI serverové části.....	14
3.4 Server na PC.....	15
3.5 Klient na mobilním zařízení.....	16
3.6 Komunikace klient-server.....	17
3.6.1 Nastavení spojení mezi mobilním zařízením a PC.....	17
3.6.2 Vlastní komunikace.....	18
4 Implementace.....	19
4.1 Grafické prvky aplikací.....	19
4.2 Implementace aplikace klient pro mobilní zařízení.....	20
4.2.1 Vytvoření GUI klienta.....	20
4.2.2 Implementace Ovládání.....	21
4.3 Implementace aplikace server pro GNU/Linux.....	23
4.3.1 GUI.....	23
4.3.2 Implementace.....	24
5 Testování.....	26
5.1 Návrh testů.....	26
5.1.1 Testování kompatibility.....	26

5.1.2 Testování uživatelské přívětivosti.....	26
5.1.3 Testování programů.....	26
5.2 Vlastní testování.....	27
5.3 Výsledky testování.....	28
6 Závěr.....	31
6.1 Možnosti rozšíření a dalšího vývoje projektu.....	31
Literatura.....	32
Seznam příloh.....	33
Příloha 1.....	34
Příloha 2.....	35

# 1 Úvod

Mobilní telefon je novým osobním počítačem. Stolní počítače tu zůstanou, ale trh se *smartphony* roste velikou rychlostí. Telefony začínají být používány jak počítače čím dál více veřejností a pro různé účely. *Smartphony* jsou obecně levnější než počítače a vhodnější pro svou přenosnost a častěji užitečnější v kontextu geolokace [1].

S příchodem operačních systémů (dále OS), jako je Android od společnosti Google, jsou mobilní telefony blíže cíli vyrovnat se osobním počítačům. Tato práce se zabývá tvorbou aplikace pro tento operační systém, která má za úkol ovládat multimediální aplikace běžící na počítači s OS GNU/Linux. Pro komunikaci mezi zařízením s OS Android a GNU/Linux je použito Bluetooth, které má oproti WI-FI menší spotřebu elektrické energie [2]. Na ovládaném počítači musí být samozřejmě také nějaký program, který bude přijímat příkazy od mobilního telefonu a posílat je programu, který chceme ovládat. Tento serverový program, který bude nabízet připojení, je vytvořen multiplatformním toolkitem Qt. Jak je patrné z výše uvedeného, budeme se zabývat hned několika technologiemi a různými přístupy při tvorbě softwaru.

V druhé kapitole jsou podrobně rozebrány mobilní operační systémy a jejich trendy. Jedna samostatná podkapitola je věnována pro tuto práci nejdůležitějšímu OS, a to systému Android. Dozvíte se v ní něco o historii tohoto projektu, o jeho nasazování do zařízení a jeho neustálém vývoji. Důležitá část je věnována tvorbě aplikací na tomto OS. O technologii Bluetooth a vytváření Bluetooth aplikací pro OS Android a GNU/Linux se dozvíte v samostatné podkapitole věnované datovým sítím. Budou zde rozebrány konkrétně Android Bluetooth API a BlueZ stack pro GNU/Linux. V závěru druhé kapitoly je popsána historie dálkového ovládaní a důvod jeho vzniku.

Kapitoly 3-6 vysvětlují vlastní vývoj dvojice aplikací, a to tyto tři fáze: Návrh, Implementace, Testování. V návrhu jsou rozebrány podrobně všechny aspekty programu, jakými jsou grafické uživatelské rozhraní, Bluetooth komunikace, ukládání dat apod.

Hlavním cílem práce je seznámit se s tvorbou aplikací pro OS Android a technologii Bluetooth a vytvoření dvou vlastních aplikací – jedné pro OS Android a druhé pro OS Linux.

## 2 Rozbor tématu vývoje mobilních aplikací a použitých technologií

### 2.1 Mobilní operační systémy

Dnešní moderní mobilní a komunikační zařízení již obsahují plnohodnotný operační systém. Takto vybavené přístroje se nazývají též *smartphony*, tedy „chytré“ telefony. U starších nebo dnešních technologicky slabších modelů se využívalo vestavěných systémů pro konkrétní typ telefonu. Vestavěné systémy náročnému uživateli neposkytují dostatečnou volnost, například možnosti přidávání aplikací jsou velmi omezené. Z těchto důvodů jsou vyvíjeny operační systémy pro mobilní telefony, které stírají rozdíly mezi osobním počítačem a mobilním zařízením.

Výhodou mobilních operačních systémů je již zmiňovaná volnost uživatele při výběru aplikací (internetový prohlížeč, hudební přehrávač, IM klient), a možnost případného vývoje vlastní aplikace a s tím související nezávislost na konkrétním zařízení (pouze na OS). Možnosti, které jsou nám poskytovány mobilními OS, se dnes opravdu blíží schopnostem operačních systémů pro PC. U nejstarších mobilních telefonů byl uživatel odkázán pouze na vestavěné aplikace, v další generaci mobilních telefonů bylo možné stahovat do mobilů jednoduché Java aplikace (převážně se jednalo o hry). Do dnešních smartphonů se vyvíjí tisíce a tisíce aplikací různých druhů přes GPS navigace, čtečky knih, cizojazyčné slovníky, až po internetové prohlížeče, přehrávače hudby a spousty dalších. Jedinou nevýhodou je relativní obtížnost ovládnutí takového systému pro některé nezkušené uživatele (např. seniory), z hlediska vysokého počtu různých speciálních funkcí, které pro ně buď nemají praktické využití, nebo je považují za nadbytečné.

Mobilních operačních systémů bylo vytvořeno velké množství. Některé z nich vznikly jako mobilní verze známých počítačových OS např. Windows Mobile, Windows CE, Maemo (modifikovaná verze Debian Linuxu) a iOS (operační systém pro iPhone od firmy Apple). Dalšími systémy, které nevycházejí z OS pro PC, jsou například PalmOS, BlackBerry OS, Symbian OS a další [1]. Nás bude nejvíce zajímat operační systém vyvíjený firmou Google a to OS Android.

### 2.2 OS Android

Operační systém Android byl původně vytvořen firmou Android Inc., ale v roce 2005 byl koupen firmou Google. Spojení s Googlem je velmi viditelné na samotné technologii OS a pro užívání zařízení s Androidem je výhodné mít nějaký uživatelský účet na Googlu (např. Gmail). Předností je, že všechny produkty, které vyvíjí společnost Google, jsou Androidem velice dobře podporovány a některé jsou vytvořeny přímo pro Android telefony, například Google Sky Maps pro pozorování



noční oblohy. Programové vybavení se pořízuje pomocí Android Market, kam je umíst'ují vývojáři a uživatelé si mohou placené aplikace kupovat a některé stahovat zadarmo. Android se již velmi rozšířil a do svých přístrojů ho dávají předně výrobci mobilních zařízení a komunikátorů jako HTC, Samsung, SonyEricson, LG atd. [3]

Systém je založen na modifikovaném jádru OS Linux. Android je tedy napsán v C a C++, ale uživatelské rozhraní a aplikace jsou psány v programovacím jazyku Java. Android je stále rozvíjen do nových verzí. V současné době je nejnovější verze 2.3 Gingerbread pro mobilní zařízení a připravuje se verze 3.0 Honeycomb, která bude určena pro tablety. Zajímavostí je, že všechny verze od 1.5 jsou pojmenovány podle zákusků (Cupcake, Donut...) [3].

## 2.2.1 Vývoj aplikací pro OS Android

Přestože je Android velmi mladý operační systém, má již velkou základnu vývojářů. Přispěla tomu velkou měrou soutěž (*The Android Google Developer Challenge*), kterou vyhlásila společnost Google a vložila do ní 10 milionů dolarů [4].

Jak již bylo uvedeno v kapitole 2.2, aplikace pro OS Android jsou vyvíjeny převážně v jazyku Java, výsledný program je zabalen do balíku APK (*Android package*) a aplikace mají tudíž koncovku .apk. Existuje NDK (Native Development Kit), které umožňuje psát aplikace v nativním kódu v jazycích jako C a C++ a zabalí je také do balíku APK, ale zatím je to v experimentální fázi.

Android SDK obsahuje nástroje a API nezbytné k vývoji Android aplikací pomocí programovacího jazyku Java. Pro vytváření aplikací je doporučeno použít vývojové prostředí Eclipse s ADT pluginem, protože v něm může vývojář během programování přímo spouštět vývojové nástroje. ADT velmi usnadňuje implementaci aplikací funkcemi jako průvodce vytvářením projektu, přímé spouštění aplikace ve virtuálním zařízení i v telefonu, vlastní XML editor a další. IDE Eclipse není nutností, lze samozřejmě použít jiná IDE, nebo samotný textový editor, ale s ADT je práce mnohem snazší.

Na testování aplikací při vývoji není nutné vlastnit mobilní zařízení, je k dispozici AVD (Android Virtual Device), což je virtuální simulace zařízení s Android OS na kterém se spouští vytvářené programy. V konečné fázi vývoje je samozřejmě vhodné otestovat program i na fyzickém zařízení, ale AVD velice usnadňuje práci, protože programátor nemusí pokaždé program nahrávat přímo do zařízení. Další velkou výhodou AVD je, že je možné těchto zařízení vytvořit více a s různými verzemi OS Android, čímž můžeme testovat zpětnou kompatibilitu programů. Dají se nastavovat také hardwarové parametry AVD jako velikost RAM, podpora dotykového displeje, klávesnice atd. AVD je tedy velmi mocný nástroj, který velmi ulehčuje psaní kompatibilních programů pro různá zařízení a verze OS.

Vytváření grafického uživatelského rozhraní (dále GUI) se provádí pomocí objektů `View` a `ViewGroup`. `View` objekty jsou základními jednotkami pro sestavování GUI. `Widget` je objekt třídy `View`, který slouží jako rozhraní pro interakci s uživatelem. Android poskytuje velké množství již hotových komponent jako jsou např. tlačítka, zaškrtačací políčka nebo pole pro zadávání textu apod. Třída `ViewGroup` je základ pro podtřídu layouts, která nabízí různé druhy rozvržení grafických prvků např. absolutní, relativní, lineární atd. Rozvržení obrazovky a jejích grafických prvků se definuje pomocí XML. Vzhled standardních widgetů je možné měnit pomocí vlastně definovaných stylů a témat.

Po vytvoření GUI se dále přiřazuje funkčnost jednotlivým komponentám. Tuto problematiku řeší takzvané UI Events, neboli události uživatelského prostředí. Je nutné definovat takzvaného naslouchače událostí (event listener) a registrovat ho v objektu `View`. Například pokud chceme implementovat událost kliknutí, tak si naimplementujeme `OnClickListener`, definujeme metodu `onClick` a zaregistrujeme metedou `setOnClickListener()`. (informace k této kapitole čerpány z [5])

## 2.3 Datové sítě

Možnosti spojení OS Android a OS Linux jsou například pomocí PAN. Bluetooth PAN (Bluetooth Personal Area Network) slouží k bezdrátovému propojení mezi Bluetooth zařízeními (mobilní telefony, počítače) a sdílením internetového připojení. Součástí Android SDK je podpora Bluetooth Network Stack, který umožňuje výměnu dat mezi zařízeními Bluetooth a to buď *point-to-point* nebo *multipoint* spojení.

### 2.3.1 Bluetooth a BlueZ

Bluetooth umožňuje bezdrátovou komunikaci dvou zařízení na krátkou vzdálenost. Nejnovější specifikace Bluetooth má přes dva tisíce stránek a najdeme v ní nepřehledné množství informací a technických detailů, které při naší implementaci nejsou tak důležité. Pro naši problematiku se zaměříme zejména na samotné programování Bluetooth aplikací, které lze v mnoha ohledech srovnat s implementacemi ostatních druhů síťových/internetových aplikací.

5 hlavních částí komunikace je:

1. Vybrat zařízení, se kterým se bude komunikovat
2. Domluva na způsobu komunikace
3. Vytvoření odchozího spojení
4. Přijmutí příchozího spojení
5. Přijímání a odesílání dat

Každé Bluetooth zařízení má svou jedinečnou 48-bitovou adresu, která je srovnatelná s MAC adresou u síťových karet, a obě adresy jsou dokonce spravované stejnou společností IEEE. Klient většinou nemá k dispozici přímo adresu serveru, protože je těžce zapamatovatelná. Tak jako do internetového prohlížeče většinou nezadáváme IP adresy serverů, nebudeme ani v Bluetooth aplikacích zadávat tyto adresy, ale uživatelsky přívětivější jména zařízení (např. „Moje PC“). Tato jména se pak převedou na adresu zařízení porovnáním názvů všech zařízení v dosahu.

Dále musíme vybrat protokol, pomocí něhož budeme komunikovat. V programování síťových aplikací nad IP využíváme zejména protokol TCP pro spolehlivé doručení a UDP pro doručení nespolehlivé. V Bluetooth jsou s těmito protokoly srovnatelné protokoly RFCOMM k TCP a L2CAP k UDP.

Server určuje číslo portu, jemuž se v Bluetooth terminologii říká kanál. Pro RFCOMM protokol, který bude využíván v naší aplikaci, máme k dispozici kanály 1-30. Počet volných kanálů je tedy celkem malý, proto není dobré nastavovat číslo kanálu fixně, ale je výhodné využít protokolu SDP, jenž vybírá číslo kanálu.

UUID (Universally Unique Identifier) je identifikační číslo, které si vybírá vývojář při vytváření Bluetooth aplikace. Za běhu programu se registruje service ID pomocí SDP serveru. Klient se pak dotazuje SDP serveru na službu se stejným UUID.

Sockety zde fungují obdobně jako BSD sockety. Klientský socket je vytvořen funkcí `create` a připojí se k serveru pomocí funkce `connect`. Jakmile je server připojen, mohou se posílat a přijímat data. Serverový socket musí po vytvoření pomocí funkce `bind` určit, který Bluetooth adaptér a jaké číslo kanálu bude používat. Dále bude naslouchat příchozím připojením funkcí `listen` a přijme příchozí spojení funkcí `accept`, kterou se vytvoří nový socket pro komunikaci s klientem. Po ukončení komunikace se sockety zavírají příkazem `close`. (informace k této kapitole čerpány z [6])

## 2.3.2 Android Bluetooth API

Rozhraní pro programování Bluetooth aplikací Bluetooth API nám umožňuje tyto funkce: hledání jiných Bluetooth zařízení, procházení spárovaného zařízení, ustanovení RFCOMM kanálů, spojení se zařízeními, přenášení dat a mnohonásobné spojení.

Při Bluetooth komunikaci musíme nejprve vyžadovat jedno z povolení `BLUETOOTH` nebo `BLUETOOTH_ADMIN` (při `BLUETOOTH_ADMIN` musíme mít i povolení `BLUETOOTH`), aby bylo vůbec možné komunikovat. Dále je nutné zkontrolovat, jestli je zařízení schopné používat Bluetooth nebo zda má tuto funkci zapnutu. Pokud je Bluetooth na mobilním zařízení vypnuté, lze umožnit uživateli, aby jej zapnul, aniž by opustil aplikaci pomocí třídy `BluetoothAdapter` a jejích metod. K nalezení zařízení, s nímž se chceme spojit, můžeme buď projít seznam již spárovaných

přístrojů, nebo vyhledat zařízení v dosahu. Pokud nemáme zapnutou viditelnost našeho zařízení, můžeme vyzvat uživatele k jejímu aktivování. K vlastnímu spojení se používají BluetoothSockets, které jsou podobné TCP socketům. Na jednom z komunikujících zařízení musí být aplikace implementována jako server a bude vytvářet server socket. Klientská část iniciuje spojení. Po vytvoření spojení můžeme přenášet data pomocí funkcí InputStream a OutputStream, jež se starají o přenos přes sockety a funkce read a write, které zapisují a čtou data do streamů. (informace k této kapitole čerpány z [5])

## 2.4 Dálkové ovládání

První dálkové ovládání na televizi bylo vytvořeno v roce 1950 a neslo název „Lazy Bones“. K ovládání vedl kabel z televizního přijímače. V roce 1955 byl představen první bezdrátový ovladač „Flashmatic“ (viz. obrázek 2.1), který odstartoval celosvětovou revoluci v ladění TV [7].



Obrázek 2.1: První bezdrátové dálkové ovládání Flashmatic (Obrázek převzat a upraven z [8])

V dnešní době jsou filmy, seriály a televizní vysílání sledována z velké části na počítačích, ať už se jedná o klasický stolní počítač nebo notebook. Většina moderních notebooků a mobilních telefonů je vybavena Bluetooth adaptérem. To z nich činí potenciální dálková ovládání. Existují i speciální dálková ovládání, která lze použít na PC. Skládají se ze snímače, který se připojí přes USB, a klasického dálkového ovládání. Avšak použití mobilního telefonu vybaveného Bluetooth je

ekonomicky efektivnější, neboť je toto zařízení mezi uživateli velmi rozšířené a není proto třeba pořizovat další přístroj. Takovéto „dálkové ovládání“ disponuje i celou řadou výhod. Není například nutné na ovládané zařízení přímo fyzicky mířit, jako je tomu u konvenčních dálkových ovládání televizorů, ale stačí se v daný okamžik nacházet v dosahu Bluetooth připojení.

Ovládání přehrávání video a audio souborů v počítači na dálku pomocí Bluetooth by nám mělo umožnit kontrolovat především základní funkce (tzn. pause, stop, volume atd.). Řešení složitějších operací přes dálkové ovládání (např. změna pořadí přehrávaných audio souborů) není již uživatelsky natolik výhodná. Také dialogová okna na obrazovce počítače se z větší vzdálenosti jeví jako příliš malá (vzniká problém čitelnosti textu).

## 3 Návrh Aplikace

V této kapitole se budeme věnovat návrhu aplikace dálkového ovládání na OS Android. Návrhy programu jsou dva, protože pro komunikaci s počítačem potřebujeme program, který funguje jako server a mobilní zařízení s naší aplikací se k němu bude moci připojit. Návrh je vždy rozdělen na tyto části: grafické rozhraní, vlastní aplikace a síťová komunikace. K odlišení programů využijeme termínů klient a server, kde klient bude aplikace na mobilním zařízení a server aplikace na PC. Klient je implementován pro OS Android v programovacím jazyku Java a server pro OS GNU/Linux programovacím jazykem C++ s pomocí Qt toolkitu. Před vlastním návrhem budou rozebrány požadavky na funkčnost obou aplikací.

Aplikaci jsem kvůli zveřejnění pojmenoval TuXcontrol. Jméno je odvozeno od anglického Remote Control a Tux (pozn. název oficiálního maskota OS Linux).

### 3.1 Požadovaná funkčnost aplikace

Tato kapitola podrobněji pojednává o vlastní funkčnosti aplikace. Serverová část aplikace přijímá příkazy přes Bluetooth rozhraní, zajišťuje jejich následné vykonání a spravuje databázi podporovaných programů. Úkolem klienta je pouhé připojení se k serveru a zasílání příkazů, které se mají vykonat.

#### 3.1.1 Funkce klientské části na Android přístroji

**Hlasitost** – Nastavení hlasitosti je jedna z nejpoužívanějších funkcí dálkového ovládání. Při přehrávání hudby nebo video souborů na počítači mají softwarové přehrávače vlastní možnosti nastavení hlasitosti, ale v operačním systému je také možné ovládat hlasitost celkovou.

**Mute** – Funkce „umlčení“, která je spjata s kontrolou hlasitosti, má své široké praktické využití a měla by být součástí efektivního dálkového ovládání.

**Play, Pause, Stop** – Trojice esenciálních funkcí, které jsou samozřejmostí pro účinnou kontrolu přehrávání.

**Přetáčení a Přeskakování stop** – Další dvojice, potažmo čtveřice funkcí, jež jsou nezbytné.

**Shuffle, Loop** – Dvě zajímavé funkce umožňující v prvním případě náhodné přehrávání, ve druhém přehrávání ve smyčce.

**Fullscreen** – Režim celé obrazovky, zejména pro přehrávání videa.

**Volitelná funkce** – Uživatel bude mít možnost si na jednom z tlačítek nastavit vlastní funkci dle potřeby (např. ukončení ovládané aplikace).

### 3.1.2 Funkce serverové části na PC s OS GNU/Linux

**Ovládání programů** – server musí jednotlivé požadavky od klienta poslat spuštěnému programu, který zrovna přehrává nějaký multimediální soubor. Některé důležité informace jsou zasílány také klientovi např. seznam právě spuštěných aplikací.

**Databáze podporovaných programů** – Uživatel si může navolit, které multimediální programy bude schopen ovládat. U každého programu je nutné nastavit jeho klávesové zkratky na konkrétní funkce, jinak se použijí výchozí zkratky. Tato databáze bude implicitně obsahovat nejznámější a nejpoužívanější programy jako je Totem, MPlayer apod. Vlastní databázi si lze uložit do XML souboru, nebo importovat databázi již existující.

**Vytvoření nabídky spojení** - Server vytvoří serverový socket, na němž bude naslouchat žádostem o spojení. Pro naše potřeby není nutné, aby se k serveru připojovalo více zařízení najednou, tudíž nebudeme řešit konkurentnost serveru. Po ustanovení komunikace bude server přijímat kódy operací a vykonávat je.

## 3.2 Návrh systému dálkového ovládání

Jak již bylo uvedeno, program se skládá z částí klient na mobilním zařízení a server na PC, kde klient odesílá serveru příkazy k vykonání. Hlavními problémy s ovládáním aplikací, jež musí návrh systému vyřešit, jsou: jakým způsobem server předá příkaz ovládané aplikaci a jak klient zvolí ovládanou aplikaci. Pro vykonávání příkazů serverem se využije skutečnosti, že velká část aplikací pro přehrávání médií podporuje klávesové zkratky. Tímto vyvstává problém s odesláním klávesové zkratky programu. Řešením by bylo naimplementování konzolové aplikace, která by simulovala vstup z klávesnice. Taková aplikace je však již vytvořena, její název je xdotool a nachází se v repozitářích linuxové distribuce Ubuntu nebo na WWW stránkách tohoto nástroje [9]. Xdotool je

nástroj pro simulaci vstupu klávesnice a myši. Mezi další funkce xdotool patří vyhledávání oken a operace s nimi (např. změna velikosti okna) [9]. Tímto nástrojem tedy vyřešíme problém odesílání příkazu aplikaci. Vyhledáme okno aplikace a odešleme příslušnou klávesovou zkratku multimediální operace. Druhý problém spočívá ve výběru ovládané aplikace (na počítači může být spuštěno více multimediálních programů najednou). Řešení je jednoduché. Server vyhledá spuštěné aplikace, které dokáže ovládat, a odešle klientovi seznam těchto aplikací. Klient vybere jednu aplikaci a odešle její jméno serveru. Server od této chvíle ovládá pouze tento vybraný program. Klient může kdykoli znovu požádat o změnu ovládané aplikace. Tímto máme navržen systém pro ovládání klávesovými zkratkami, ale protože každý program může mít své specifické zkratky, musíme udržovat databázi těchto zkratk pro každou podporovanou aplikaci.

## 3.3 Návrh GUI

Uživatelské rozhraní je nejdůležitější částí každého počítačového systému. Je to systém pro nejvíce uživatelů. Může být viděn, slyšen a můžeme se jej dotknout. Cíle návrhu uživatelské rozhraní jsou jednoduché: udělat práci s počítačem jednoduchou, produktivní a zábavnou [10].

Tato kapitola popisuje návrhy rozhraní pro server na PC a klienta na mobilní telefon. Server by sice mohl být pouze konzolovou aplikací, ale pro uživatelskou přívětivost bude vhodnější program s grafickým rozhraním.

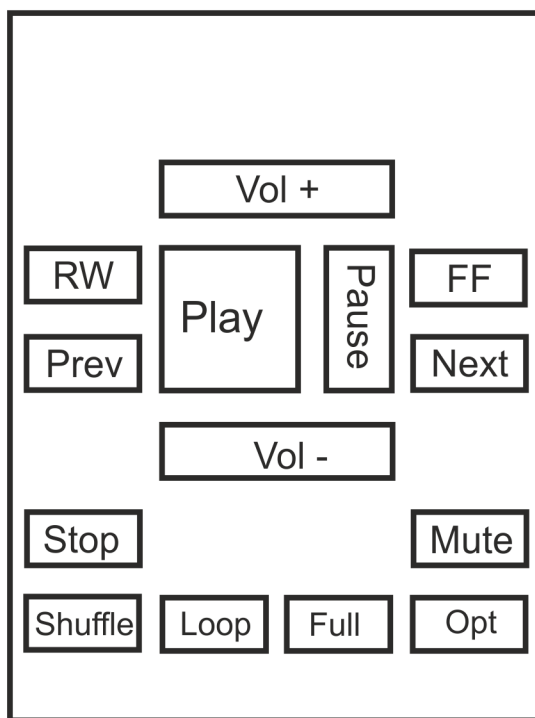
### 3.3.1 GUI dálkového ovládání

Uživatelské rozhraní dálkového ovládání se skládá z těchto částí: dialog pro vyhledávání a připojení k zařízení, hlavní obrazovka s ovládacími prvky a menu pro výběr zařízení a aplikace. Nejprve si rozebereme nejdůležitější část, kterou je obrazovka s tlačítky. Porovnáním výhod a nevýhod odlišných hardwarových dálkových ovládaní a přehrávačů, jsem navrhl co nejefektivnější rozložení ovládacích prvků (viz. obrázek 3.1).

Všechna tlačítka obsahují ikony, které reprezentují multimediální operace. Uprostřed obrazovky se nachází skupinka nejčastěji používaných tlačítek. Tlačítka *play* a *pause* jsou jedna z vůbec nejpoužívanějších funkcí při přehrávání audio a video souborů, proto jsou umístěna uprostřed a jsou nejmohutnější. Kontrola hlasitosti je pro svůj logický směr operace (zvyšujeme a snižujeme) umístěna nad a pod *play* a *pause*. Tlačítka pro přeskokování např. zvukových stop a posun po přehrávaném souboru, jsou umístěna dle směru jejich operace na levé nebo pravé straně dvojice *play* a *pause*. Na spodních okrajích výše zmíněné skupiny jsou operace *stop* a *mute*. Spodní řádek pak



patří tlačítkům s pokročilejší, a tudíž ne tak často používanou funkcí. Jedná se o náhodné přehrávání - *Shuffle*, smyčku - *Loop*, celou obrazovku - *Fullscreen* a volitelná funkce – *Optional*.



Obrázek 3.1: Návrh hlavní obrazovky

Menu pro výběr možností (připojit zařízení a vybrat aplikaci) se zobrazí po stisknutí hardwarového tlačítka menu jako takzvané Options Menu (viz. obrázek 3.2)



Obrázek 3.2: Options Menu

Skládá se pouze ze dvou ikon a jejich popisků. Ikony jsou standardní android ikony.

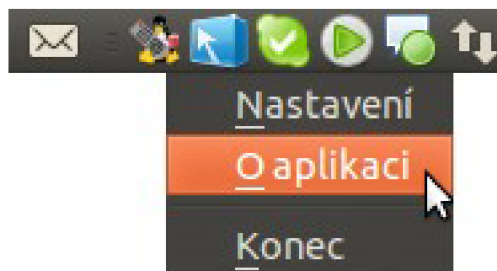
Dialog pro připojení k zařízení se objeví po zvolení volby připojit z Options Menu. Je tvořen dvěma seznamy a tlačítkem. První seznam obsahuje již v minulosti spárovaná zařízení a na druhém seznamu nalezneme po stisknutí tlačítka *vyhledat* názvy Bluetooth zařízení v dosahu. Jedno zařízení vybereme a potvrdíme připojení.

Po úspěšném připojení nám server pošle nabídku právě běžících a naší aplikací podporovaných programů. Uživatel si vybere z této nabídky jeden program. Po výběru aplikace přejdeme na hlavní obrazovku dálkového ovládání. Pokud budeme chtít změnit ovládanou aplikaci, zvolíme v Options Menu možnost vybrat aplikaci a program si od serveru znovu vyžádá seznam běžících aplikací. Uživatel po té může změnit svůj předchozí výběr.

### 3.3.2 GUI serverové části

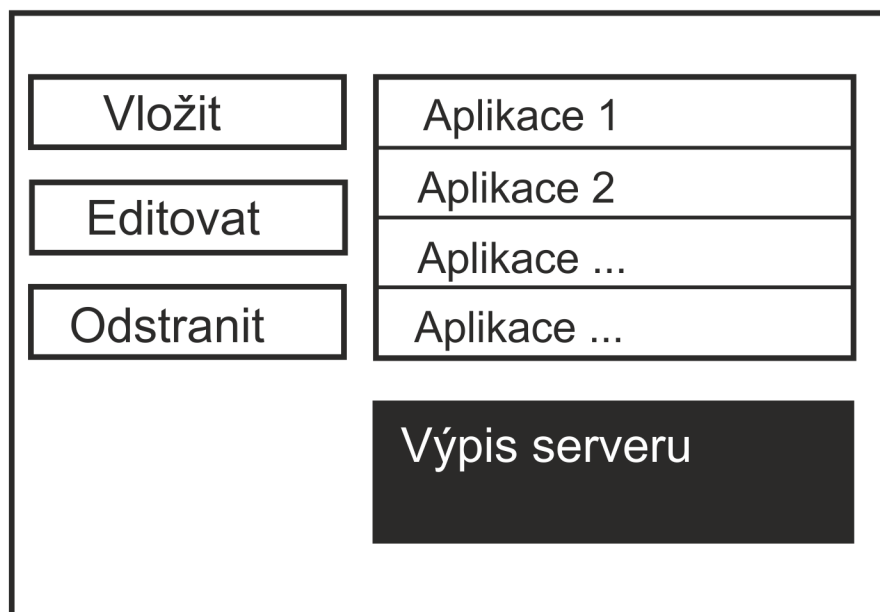
Serverová část aplikace bude vytvořena pomocí Qt toolkitu, který nám nabízí bohaté možnosti při tvoření grafického rozhraní. Pomocí Qt designeru si můžeme přímo graficky navrhnout jednotlivá okna, dialogy, formuláře, tlačítka atd.

Tato aplikace běží především na pozadí a uživatel (kromě spuštění) nepotřebuje zadávat nějaké další volby. Nejvhodnějším řešením je po spuštění aplikaci automaticky schovat a přistupovat k ní pomocí tzv. tray ikony. Moderní operační systémy většinou nabízejí speciální oblast zvanou *system tray*, kde dlouho běžící aplikace mohou zobrazovat ikony a krátké zprávy [11]. Pravým kliknutím na tray ikonu se nám vyvolá nabídka, z níž si uživatel může vybrat tyto možnosti: nastavení, o aplikaci a ukončení (viz. obrázek 3.3).



Obrázek 3.3: Tray ikona aplikace s nabídkou v prostředí Gnome

Volbou možnosti *nastavení* se objeví hlavní okno serverové aplikace (viz. schematický obrázek 3.4). Hlavním prvkem okna je tabulka, v níž se nacházejí všechny podporované aplikace. Každý řádek obsahuje jednu aplikaci (její název). Vedle tabulky se nalézají tlačítka upravit, odstranit a přidat záznam. Záznamy lze upravovat také dvojklikem na buňku tabulky. Při zvolení úpravy nebo vytvoření nového záznamu nějakého multimediálního programu se nám objeví dialogové okno, v němž vyplníme jednotlivé klávesové zkratky a další potřebné údaje. Pro usnadnění vytváření jsou všechna políčka vyplněna výchozími hodnotami.



Obrázek 3.4: Nákres hlavního okna

Další část hlavního okna je textové pole fungující jako výstup hlášení Bluetooth serveru, z něhož se dozvíme, jaké zařízení je připojeno, případně zda nedošlo k nějaké chybě.

Program obsahuje také nápovědu, kterou si může uživatel zobrazit přes nabídku Help a volbu *manuál*. Nápověda je HTML dokument, jenž se objeví jako dialog s QTextBrowserem. Další volby v nabídce Help jsou: O programu a O Qt. Obě tyto možnosti zobrazí dialog s informacemi.

## 3.4 Server na PC

Významnou částí serveru na PC je databáze aplikací a práce s ní. Jak již bylo zmíněno v kapitole 3.1.2, budou se data o aplikacích načítat a ukládat do XML souboru, který má takovou strukturu:

```

<data>
  <app>
    <name>MediaPlayer</name>
    <play>Enter</play>
    <pause>Ctrl+P</pause>
    ...
  </app>
  <app>
    ...
  </app>
</data>

```

Po spuštění programu se z XML souboru načtou do QListu struktur aplikace. Struktura aplikace se skládá z řetězce *název* a QListu řetězců *zkratky*. K načítání se využívá QdomDocumentu, z kterého si vytvoříme QDomNodeList skládajícího se z prvků <app>. NodeList se postupně prochází a jednotlivé zkratky jsou přidávány do naší struktury. Ukládání do souboru XML probíhá při každé změně ze strany uživatele. Při mazání je uživatel vždy vyzván k potvrzení operace.

Komunikace a vše, co s ní souvisí, popsána v kapitole 3.6.

Na tomto místě je nutné popsat mechanismus vykonávání příkazů klienta serverem. Pro simulaci klávesových zkratk je využíván terminálový program xdotool, který je spouštěn například s těmito parametry:

```
xdotool search "SMPlayer" windowactivate --sync key --clearmodifiers space
```

Nejprve se vyhledá spuštěná aplikace (v příkladu výše SMPlayer) a pošle se klávesa *space* nebo-li mezerník, která způsobí pauzu nebo přehrávání.

Jak určit, který program chce uživatel ovládat? Po připojení klienta k serveru pomocí databáze podporovaných programů, si vytvoříme seznam běžících aplikací a ten pošleme klientovi. Uživatel si vybere program, s nímž bude komunikovat, dokud si nezvolí program jiný nebo dokud jedna z komunikujících stran neukončí spojení.

Všechny terminálové příkazy se vykonávají příkazem systém („command“) za pomoci knihovny stdlib.h. Od klienta nám tedy přicházejí kódy příkazů, my je vybíráme z naší načtené databáze a ihned provádíme příkazy nástrojem xdotool.

## 3.5 Klient na mobilním zařízení

V kapitole 3.1.1 byla popsána funkce jednotlivých tlačítek z pohledu uživatele našeho programu. Klientská část programu z pohledu programátora má následující funkčnost:

1. Odchytávání jednotlivých stisknutí tlačítek a vytváření kódů pro serverový program
2. Navázání spojení se serverem
3. Možnost výběru ovládaného programu

Náš program je tedy vícevláknový. V hlavním vlákně je řešena interakce s uživatelem a v druhém vlákně komunikace se serverem a vše s ní spojené.

Odchytávání stisknutých tlačítek se provádí pomocí nastavení takzvaného „naslouchače událostí“, nebo-li event listener, který je registrován s třídou view. Pro každou funkci je určen příkaz, který se posílá aktivitě, jež se stará o komunikaci se serverem (viz. tabulka 3.1).

Funkce	Příkaz	Funkce	Příkaz
Přehrát	Play	Vypnout zvuk	Mute
Pauza	Pause	Zeslabit hlasitost	VolDown
Stop	Stop	Zesílit hlasitost	VolUp
Posun vpřed	FF	Náhodné přehrávání	Shuffle
Posun dozadu	RW	Smyčka	Loop
Další skladba	SkipL	Celá obrazovka	Full
Předchozí skladba	SkipR	Volitelná funkce	Optional

Tabulka 3.1: Hodnoty funkcí přehrávače posílané serveru

Návrh objektů vychází z příkladu Bluetooth chatu [12]. Pro naše účely nám postačí tři aktivity: hlavní s gui ovládáním, jedna pro vyhledávání zařízení a jedna pro Bluetooth komunikaci. Synchronizace mezi těmito aktivitami se provádí pomocí objektu Handler a zasíláním zpráv tomuto objektu z ostatních aktivit. Vlastní síťová komunikace a problematika s ní spojená bude popsána v kapitole 3.6.

## 3.6 Komunikace klient-server

Jak již bylo popsáno v kapitole 2.3.2, operační systém Android obsahuje vlastní Bluetooth API, které využijeme v našem případě k *point-to-point* spojení a vyhledávání zařízení v okolí. OS Linux obsahuje BlueZ a jeho C knihovny Bluetooth.

### 3.6.1 Nastavení spojení mezi mobilním zařízením a PC

Na straně serveru musíme pomocí SDP zaregistrovat službu, k tomu potřebujeme vytvořit jedinečné identifikační číslo služby UUID. Pro vytvoření UUID můžeme využít generátor například na webu [13] nebo si jej můžeme vymyslet. Pro registraci služby na lokálním SDP serveru je třeba naplnit potřebné struktury a proměnné parametry, které jsou serveru odeslány, a je vytvořeno sezení, při němž je nabízena služba.

Nastavíme RFCOMM kanál nebo-li port na kterém budeme komunikovat, jméno služby, poskytovatele a popis služby. Dále naplníme struktury pro nastavení UUID služby, viditelnost služby, l2cap a RFCOMM informace a připojení informací o protokolu. Po naplnění všech struktur a uvedených proměnných se připojíme k SDP serveru a vytvoříme sezení.

Jakmile je sezení nastaveno, je možné pokračovat jako při tvorbě klasického TCP serveru. Vytvoříme si Bluetooth RFCOMM socket. Pomocí funkce bind() nastavíme socketu adresu lokálního

Bluetooth adaptéru a funkcí `listen()` nasloucháme přichozím připojení od klientů. Přijetím spojení funkcí `accept()` je vytvořen klientský socket a v nekonečné smyčce jsou přijímána a odesílána data, dokud klient neukončí spojení, nebo nedojde k přerušení spojení z jiných příčin (např. překročení limitní vzdálenosti maximálního dosahu Bluetooth). Když se klient odpojí, nasloucháme znovu dalším spojením.

Na straně klienta je nutné nejprve vyhledat zařízení, na kterém běží server a připojit se k němu. Při návrhu jsem použil příklad Bluetooth Chat [12], kdy jsou využita dvě vlákna - jedno pro připojování, které po úspěšném připojení zrušíme a vytvoříme druhé vlákno, v němž se posílají a přijímají data. Vlákno pro připojování vytvoří RFCOMM socket s UUID naší služby, naváže spojení se serverem, předá tento socket druhému vláknu (pro správu připojení) a ukončí se. Synchronizaci s GUI kontroluje *handler*, jemuž jsou zasílány zprávy o změně stavu.

### 3.6.2 Vlastní komunikace

Ihned po připojení klienta k serveru obdrží klient seznam právě běžících aplikací, které je možné ovládat. Název vybrané aplikace se odešle zpátky serveru, který od této doby přijímá příkazy pro ovládání tohoto konkrétního programu (viz. tabulka 3.1). Pokud se klient rozhodne změnit výběr aplikace, pošle zprávu s žádostí o zaslání seznamu aplikací a server po přijetí této zprávy pošle seznam. Jakmile uživatel vypne na mobilním telefonu aplikaci, klient před samotným ukončením odešle zprávu serveru o ukončení spojení, aby server mohl uzavřít spojení na své straně. Formát zpráv komunikace je popsán v tabulce 3.2.

Funkce	Příkaz
Žádost o zaslání seznamu aplikací	<code>sendApps</code>
Seznam podporovaných aplikací	<code>app:aplikace1;aplikace2;</code>
Vybraná aplikace	<code>chosenApp:aplikace1</code>
Klient ukončil aplikaci dálkového ovládání	<code>konec</code>

Tabulka 3.2: Zprávy komunikace serveru a klienta

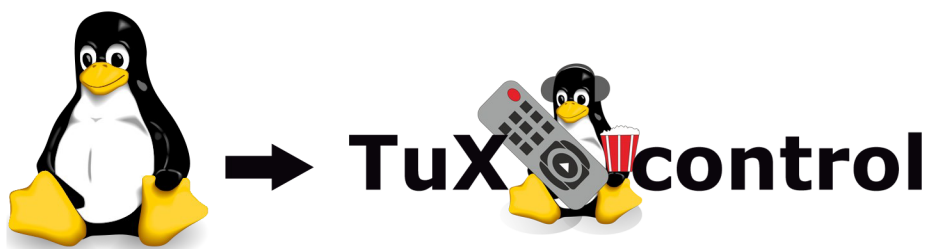
## 4 Implementace

V této kapitole je popsána realizace návrhu z kapitoly 3 a případné odklony od tohoto návrhu.

### 4.1 Grafické prvky aplikací

Pro lepší uživatelskou přívětivost a komfortnost užívání aplikace, jsem vytvořil speciální vzhled těchto prvků: logo aplikace, ikona, pozadí dálkového ovládání a veškerá tlačítka dálkového ovládání. Grafické prvky byly navrženy v grafickém programu Gimp.

Logo aplikace je modifikovaný obrázek Tuxe Larryho Ewinga, jenž byl vytvořen v aplikaci Gimp a je oficiálním maskotem operačního systému Linux. Na obrázku Tux (tučňák) drží v ruce dálkové ovládání a popcorn a na hlavě má nasazena sluchátka. V pozadí loga je nápis názvu aplikace. Transformace ve výsledné logo je demonstrována na obrázku 4.1. Ikona aplikace je zmenšeným logem bez nápisu.



Obrázek 4.1: Původní obrázek Larryho Ewinga a logo aplikace TuxControl

Tlačítka jsem také vytvořil v programu Gimp. Jsou černo-stříbrná a každé má na sobě ikonu funkce, jež jsou uživatelům známé z přehrávačů a jiných dálkových ovládání viz. tlačítko *play* je zobrazeno na obrázku 4.2. Pozadí aplikace a celkový vzhled dálkového ovládání je na obrázku 4.3.



Obrázek 4.2: Tlačítko pro přehrávání - *play*

## 4.2 Implementace aplikace klient pro mobilní zařízení

### 4.2.1 Vytvoření GUI klienta

Rozložení tlačítek dálkového ovládání bylo vytvořeno přesně podle návrhu z kapitoly 3.3.1. Pro sestavení rozhraní jsem použil relativní layout, aby byla aplikace snadno zobrazitelná na různých displejích. Aplikace díky své jednoduchosti umožňuje pouze jednu orientaci displeje. V horní části je umístěn ImageView, ve kterém je vyobrazeno logo aplikace. Pod ním se nachází TextView pro zobrazování stavu připojení. Dále následují ovládací tlačítka rozmístěná podle návrhu. Poloha tlačítek je vždy závislá na poloze jednoho z grafických prvků. Výsledné grafické rozhraní je zobrazeno na obrázek 4.3.



Obrázek 4.3: Grafické rozhraní dálkového ovládání

Dialogové okno pro výběr a hledání zařízení je tvořeno lineárním layoutem. Skládá se ze dvou listView, které mají nad sebou popisek z TextView. V dolní části je tlačítko pro vyhledávání Bluetooth zařízení v okolí. Oba listy i popisky jsou implicitně neviditelné a jejich viditelnost se nastavuje až programově, podle toho, jestli obsahují nějaké položky.



Options menu pro výběr aplikace a vyhledávání obsahuje dvě ikony s popiskami. Ikonu pro vyhledávání a připojení k zařízení jsem vybral `ic_menu_mylocation` a pro výběr aplikace `ic_menu_agenda`. Jsou to standardní ikony OS Android.

## 4.2.2 Implementace Ovládání

V následujícím textu je popsána implementace aplikace dálkového ovládání TuXcontrol pro OS Android v programovacím jazyce Java. Program se skládá ze tří aktivit, které jsou podrobně popsány.

### 4.2.2.1 RemoteActivity

Na úplném začátku je třeba vytvořit aktivitu, která bude pojmenována `RemoteActivity`, jež má za úkol obstarávat GUI. Přetížíme metodu `OnCreate`, která se volá jako první. V této metodě se nastaví *layout* okna, zruší se horní lišta s názvem aplikace a všem tlačítkům je přiřazen jeden `OnClickListener`, který „naslouchá“ kliknutím na tlačítka. V další metodě `OnStart` se kontroluje, zda je povolena funkce Bluetooth. Pokud tomu tak není, je uživateli nabídnuto zapnutí. Jestliže zapnutí není povoleno, program je automaticky ukončen. Jakmile je Bluetooth zapnuto, inicializuje se aktivita `BluetoothRemote`, která kontroluje Bluetooth komunikaci, a je jí přiřazena do proměnné `myRemote`. V metodě `OnResume` je voláno `myRemote.start()` (chování aktivity `myRemote` si popíšeme později). Metoda `OnFinish` odešle serveru před samotným ukončením příkaz „konec“. Ve funkci `setupRemote()` inicializujeme aktivitu `BluetoothRemote`, což bylo využito ve výše uvedené metodě `OnStart`. Dále implementujeme funkci `sendOrder`, která odesílá řetězce příkazů pomocí `myRemote`. Funkce `onClick` je nastavena tlačítkům jako `OnClickListener` a pouze zjišťuje, které tlačítko bylo stisknuto a funkcí `sendOrder` odešle řetězec příslušného příkazu.

Dále je implementován objekt třídy `Handler`, který obsahuje metodu `handleMessage`. Zpracovává tedy zprávy od `myRemote`. Jsou tři typy zpráv: `MESSAGE_STATE_CHANGE`, `MESSAGE_READ`, `MESSAGE_TOAST`. První indikuje změnu stavu připojení a její druhý argument indikuje nový stav. `MESSAGE_READ` znamená, že nám server poslal nějakou zprávu a druhý argument představuje tělo zprávy. Server posílá pouze zprávu se seznamem spuštěných aplikací, nebo zprávu `noApp`. Pokud se jedná o zprávu se seznamem, rozdělíme ji na pole se jmény aplikací a zobrazíme `AlertDialog`, ve kterém uživatel může zvolit jednu z aplikací. `MESSAGE_TOAST`, jak název napovídá, zobrazí zprávu typu `toast`.

`OnActivityResult` čeká na odpovědi z aktivit a to z té, kterou vyvoláme, pokud není povoleno Bluetooth a dále z aktivity, v níž vybíráme zařízení k připojení. V první variantě se kontroluje, jestli uživatel skutečně aktivoval Bluetooth. Pokud tomu tak není, aplikace se ukončí. Při výběru zařízení je druhým argumentem adresa, na kterou se chceme připojit. Toho se dosáhne pomocí objektu `myRemote`.

Jako poslední je v RemoteActivity programově vytvořeno Options menu a nastavena obsluha menu přetížením metody onOptionsItemSelected. Zde při výběru zařízení začneme aktivitu DeviceActivity a při znovu zvolení aplikace se odešle serveru příkaz sendApps.

#### 4.2.2.2 BluetoothRemote

Další důležitou aktivitou je BluetoothRemote, kde se uskutečňují veškeré operace spojené s komunikací. V této aktivitě jsou dvě vlákna - jedno pro připojování a druhé pro připojený stav. Nejprve se definuje UUID a konstanty stavu připojení. Jsou zde implementovány funkce setState a getState, které nastavují a získávají aktuální stav připojení. Ve funkci start se zruší všechna vlákna s připojením a je nastaven stav na STATE\_NONE. Metoda connect zruší všechna běžící vlákna a vytvoří nové vlákno pro připojování k zařízení. Connected() si vezme od připojovacího vlákna socket a vytvoří vlákno pro správu připojení a vlákno pro připojování zruší. Stop zruší všechna vlákna. Write je funkce volaná z RemoteActivity a zapisuje data do connected vlákna, které je posílá serveru. ConnectionFailed() a connectionLost() posílají GUI zprávu typu MESSAGE\_TOAST o nezdařeném připojení a nastaví stav na STATE\_NONE.

Vlákno connectThread vytvoří k požadovanému zařízení RFCOMM socket pro službu s UUID služby funkcí createRfcommSocketToServiceRecord. V metodě run se připojujeme funkcí connect a po úspěšném připojení je volána funkce connected. Cancel zavře socket a volá connectionFailed. Vlákno connectedThread vytvoří dva streamy pro vstup a výstup a v metodě run v nekonečném cyklu čte vstupní data ze serveru a posílá je GUI. Pokud při čtení dojde k nějaké chybě, volá connectionLost. Metoda write zapisuje do výstupního streamu.

#### 4.2.2.3 DeviceActivity

Poslední část programu je DeviceActivity - aktivita pro vyhledávání zařízení v okolí a spárovaných zařízení. Jsou vytvořena dvě pole pro spárovaná a hledaná zařízení. V onCreate je nastaven vzhled dialogu a layout pro jednotlivé prvky GUI. Vyhledávacímu tlačítku se přiřadí OnClickListener, který spustí vyhledávání. Dále je nastaven broadcastový přijímač, který vyhledává zařízení. Pokud nějaké najde, tak jej zapíše do příslušného listu i s MAC adresou zařízení. Na oba listy je implementován OnClickListener, který po výběru zařízení odešle jeho adresu a ukončí tuto aktivitu.

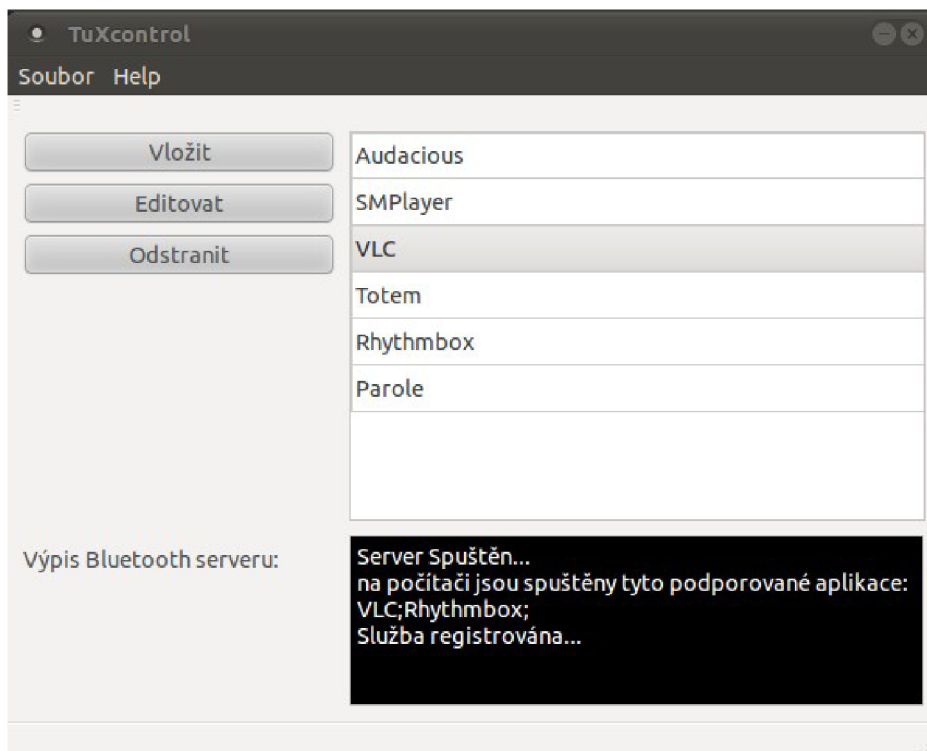
## 4.3 Implementace aplikace server pro GNU/Linux

### 4.3.1 GUI

Grafické rozhraní je vytvořeno pomocí WYSIWYG editoru Qt Designer. V této pomocné aplikaci si programátor snadno navrhne požadované GUI a okamžitě může vizuálně zhodnotit výsledek. Hlavní okno je vytvořeno podle návrhu z kapitoly 3.3.2. V levém horním rohu se nachází trojice tlačítek: vložit, editovat, odstranit. Vedle tlačítek je umístěn QTaleWidget s podporovanými aplikacemi. V dolní části aplikace nalezneme výstupní „pseudo-konzoli“ tvořenou QplainTextEdit. Výsledný vzhled aplikace je znázorněn na obrázku 4.4.

Tray ikona je vytvořena programově funkcí createTrayIcon. V této funkci je nejdříve navrženo Qmenu s položkami a pak samotná QsystemTrayIcon. Tray ikoně je přiřazeno Qmenu a soubor s obrázkem ikony aplikace. Výsledek je na obrázku 3.3.

Dialogové okno pro přidávání a editaci záznamů o klávesových zkratkách programů, je také vytvořeno pomocí nástroje Qt Designer. Skládá se vždy z dvojic: název funkce - QLabel a editační pole pro zkratku – QlineEdit. V dolní části okna je umístěn box se dvěma tlačítky pro přijetí nebo odmítnutí dialogu – QDialogButtonBox.



Obrázek 4.4: Hlavní okno serverové části aplikace

## 4.3.2 Implementace

### 4.3.2.1 Práce s databází programů a jejich klávesových zkratk

Nejprve popíše implementaci práce s databází aplikací. V kapitole 3.4 jsme si ukázali, jak bude vypadat formát XML souboru, z kterého se načítají data a zapisují se do něj. Pro práci s XML jsem implementoval dvě funkce `saveData` a `loadData`. Ve funkci `loadData` jsou načítána data ze souboru do `QListu` struktur (viz. Kapitola 3.4). Dále je vytvořen `QdomDocument` a jako obsah je stanoven náš XML soubor. Z dokumentu jsou vybrány všechny prvky se značkou `<app>` a tím je sestaven takzvaný `nodelist`. Ten se postupně prochází a data se ukládají do výše zmíněného `QListu`. Při tomto procesu se zobrazují jména aplikací do tabulky v hlavním okně. Funkce `saveData` naopak prochází `Qlist` a zapisuje do XML souboru.

Pro práci se záznamy jsou implementovány tři funkce (`createEntry`, `editEntry`, `deleteEntry`), které jsou zároveň sloty pro signály stisknutí příslušného tlačítka. `CreateEntry` zobrazí dialog pro vytvoření nového záznamu a pokud ho uživatel potvrdí, uloží data `saveData` a přidá řádek do tabulky s aplikacemi. Funkce `editEntry` má dvě varianty. Jedna je spojená se signálem dvojklik na buňku tabulky a druhá na signál stisknutí tlačítka. Do editačního okna se načtou hodnoty ze struktury a uživatel je může změnit podle svých potřeb. Po potvrzení editace se uloží znovu data do XML souboru. `DeleteEntry` smaže požadovaný záznam z `Qlistu`, tabulky a uloží změny do souboru.

### 4.3.2.2 Bluetooth komunikace.

Tímto se dostáváme k hlavní části programu a tou je vlastní Bluetooth komunikace. Při spuštění programu se vytvoří nové vlákno `keyThread`, které slouží pro komunikaci a vykonávání příkazů dálkového ovládání. Reimplementujeme funkci `run`. V této funkci definujeme strukturu `sdp_session_t`, kterou naplníme funkcí `registerService`, jež byla použita z knihy [6]. Vytvoříme `RFCOMM` socket a nabindujeme ho na lokální Bluetooth adaptér. Nasloucháme příchozím připojením funkcí `listen` a v nekonečném cyklu přijímané spojení funkcí `accept`. Pokud se připojí nějaký klient, ihned mu pošleme seznam běžících aplikací, který získáme funkcí `getRunningApps`, jejíž implementaci si popíšeme později. Dále následuje další nekonečná smyčka, v níž budeme přijímat data od klienta, dokud nepošle zprávu o svém ukončení nebo nenastane nějaká porucha spojení. Příchozí data analyzujeme porovnáním očekávaných řetězců. Příkazy pro přehrávače a jiné aplikace vykonáváme příkazem `system` („příkaz `xdotool`“), kde příkaz `xdotool` má formát uvedený v kapitole 3.4. Při přijetí zprávy „konec“, nebo při chybě čtení, vyskočíme z tohoto cyklu a budeme znovu naslouchat dalším připojením.

Nyní si popíšeme již zmíněnou funkci `getRunningApps`. Pomocí příkazu `ps -U root -u root -N , „r“` načteme výpis procesů, které nespustil `root`. Postupně zjišťujeme, jestli výpis neobsahuje nějaký námi podporovaný program. Funkce vrátí řetězec obsahující jména aplikací oddělená středníkem.

# 5 Testování

Po vlastní implementaci aplikace následuje její testování a odladění. V této podkapitole se budeme věnovat návrhu testů. Testovat se budou aspekty jako je kompatibilita, uživatelská přívětivost a funkčnost aplikace na různých programech.

## 5.1 Návrh testů

### 5.1.1 Testování kompatibility

Kompatibilita je testována u programu pro OS Android na různých zařízeních s odlišnými rozlišeními displeje, kde sledujeme správnost implementace grafického rozhraní. Aplikace byla primárně vyvíjena na mém osobním telefonu LG Optimus One s rozlišením displeje HVGA 320x480.

Protože distribucí GNU/Linux je velké množství a mají různá pracovní prostředí, bylo by vhodné otestovat funkčnost serverové části aplikace na různých distribucích. Program byl vyvíjen v distribuci Ubuntu 10.10 s pracovním prostředí Gnome 2.32.0. Testování bude probíhat na live distribucích Xubuntu s Xfce, Kubuntu s KDE.

### 5.1.2 Testování uživatelské přívětivosti

Testovány byly opět obě části aplikace. Dobrovolným testerům programu jsem vypůjčil svůj mobilní telefon a počítač s nainstalovaným TuXcontrol serverem. Jelikož velké procento uživatelů nemá zkušenosti s OS GNU/Linux, měli již na počítači spuštěn přehrávač SMPlayer a nastavené správné klávesové zkratky kromě funkce stop, kterou museli přednastavit na stisknutí klávesy q. Tester měl tedy za úkol na mobilním přístroji spustit program TuXcontrol, připojit se k PC, zkusit různé funkce a zastavit přehrávání tlačítkem stop, které měl za úkol si přednastavit. Na konci testování byl skupině testerů předán dotazník zkoumající jejich zkušenost s aplikací. Celý dotazník naleznete v příloze 2.

### 5.1.3 Testování programů

Program je určen pro ovládání multimediálních aplikací, proto by bylo vhodné vědět, které aplikace dokáže ovládat a zda je jejich ovládání efektivní. Ze stránek [14] a z programů nainstalovaných v testovaných distribucích jsem vybral 10 multimediálních přehrávačů (6 specializované na přehrávání audio souborů a 4 na přehrávání videa). Zkouška spočívá v otestování použitelnosti na těchto populárních přehrávačích a zároveň vytvoření použitelné databáze programu, tak aby mohl

být jednou distribuován jako plnohodnotná a fungující aplikace. Ovládání by také mohlo být použito pro ovládání prezentací, a proto jsem do testování začlenil i programy Document Viewer a Impress z řady OpenOffice.org.

## 5.2 Vlastní testování

Testování použitelnosti programu na různých druzích displejů jsem mohl otestovat pouze na virtuálním zařízení s různými parametry, protože jsem v době testování neměl k dispozici přístroje na vypůjčení. Ale pro účely testování zobrazení displeje je tato zkouška dostačující. Na obrázku 5.1 můžete pozorovat aplikaci s různými druhy rozlišení.



Obrázek 5.1: Aplikace v rozlišeních odleva QVGA, HVGA, QWVGA800

Distribuce OS GNU/Linux pro testování byly vždy live CD, na které se doinstaloval xdtool a Qt SDK, Bluetooth knihovna, případně nějaký multimediální program. První testovanou distribucí bylo Xubuntu 11.04 s pracovním prostředím Xfce. Program fungoval bezchybně a implicitní přehrávače ovládal bez větších problémů. Dalším operačním systémem bylo Kubuntu 11.04 s KDE, které také pracovalo bez obtíží stejně jako Ubuntu 10.10, na němž byl program vytvořen.

Uživatelskou přívětivost testovalo 8 testerů, z nichž většinu tvořili studenti vysoké školy. Testování probíhalo na Ubuntu 10.10. na programu SMPlayer. Všichni testeři splnili všechny úkoly testování bez závažných problémů. Data z vyplněných dotazníků jsou analyzována v další kapitole.

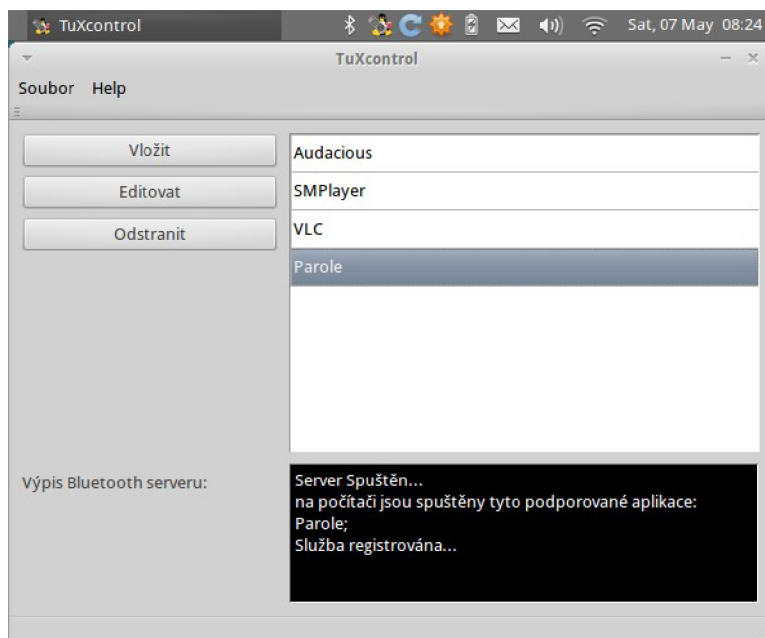
Testování programů jsem prováděl částečně s testováním funkčnosti programu na různých distribucích a prostředích, protože některé multimediální programy fungují nejlépe v pracovních

prostředích, pro která byly vytvořeny. Na Ubuntu 10.10 jsem testoval pouze aplikace pro prostředí Gnome. Přehrávač Totem má omezené množství klávesových zkratk a xdotool neumí zasílat příkazy maximalizovanému režimu tohoto přehrávače. V programu SMPlayer (populární nástavba na Mplayer) musí uživatel změnit klávesovou zkratku pro přehrávání - *play*, pokud chce odlišit operace *play* a *pause* (není to nutnost), protože xdotool neumí emulovat multimediální klávesy. Přehrávač VLC po přidání zkratk fungoval bezchybně. Hudební přehrávač Exaile bohužel vůbec nepodporuje klávesové zkratky a je našim programem neovladatelný. V prostředí Xfce jsem otestoval přehrávač Parole, který má pouze několik funkcí s nastavenými zkratkami a přenastavit je nelze. Gmusicbrowser, implicitní hudební přehrávač v Xubuntu, neměl nastavené žádné klávesové zkratky a ani po jejich nastavení na ovládání nereagoval.

U programů pro prohlížení dokumentů s prezentacemi bohužel vyvstal stejný problém jako u přehrávače Totem - neschopnost ovládat po maximalizaci prezentace, protože program xdotool poté nenalezne okno programu.

## 5.3 Výsledky testování

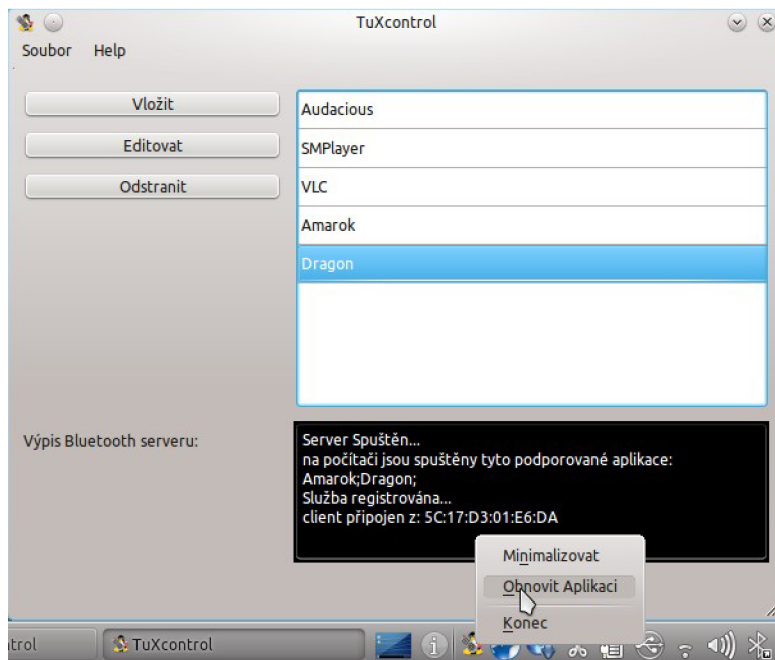
Výsledky testování funkčnosti aplikace na mobilních telefonech s různým rozlišením, jak je vidět na obrázku 5.1, jsou uspokojivé. GUI bylo dobře navrženo pro různé podmínky.



Obrázek 5.2: Serverová aplikace spuštěná v distribuci Xubuntu s prostředím Xfce

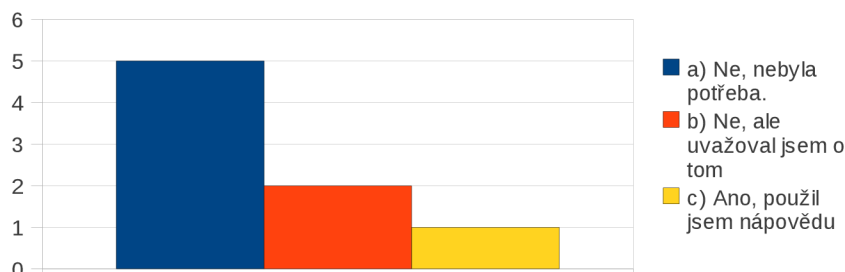


Program fungoval bezchybně na všech distribucích v testu. Pro podrobnější testování na jiných distribucích by bylo dobré využít zpětnou vazbu od koncových uživatelů, protože linuxových distribucí je opravdu velké množství. Na obrázku 5.2 je ukázka aplikací v prostředí Xfce a na obrázku 5.3 prostředí KDE.



Obrázek 5.3: Aplikace spuštěná v distribuci Kubuntu s prostředím KDE

Výsledky dotazníků z testu uživatelské přívětivosti si můžete prohlédnout v příloze 2. Pro statisticky významnější výsledky testů bude třeba shromáždit větší množství vyplněných dotazníků. Z výsledků vyplývá, že program je snadno použitelný a uživatelsky přívětivý. Hlavním nedostatkem, který v závěrečném prostoru pro vyjádření zmínilo 6 z 8 testerů, bylo manuální zadávání klávesové zkratky. Toto bude v příští verzi programu opraveno. V Grafu 5.1 jsou vyobrazeny výsledky na jednu z otázek dotazníku.



Graf 5.1: Výsledky otázky: Měl/a jste při úkolech potřebu použít nápovědu?

Výsledky testování použitelnosti aplikací můžete pozorovat v tabulce 5.1. Pro některé programy z důvodu absence klávesových zkratk a jejich nastavení, je TuXcontrol nepoužitelný.

Název	Zhodnocení
Amarok	V pořádku v prostředí KDE, v Gnome nefunguje ovládání
Audacious	V pořádku
Banshee	Nepodporuje klávesové zkratky pro některé základní funkce (volume, přetáčení) i pokročilé
Exaile	Nepodporuje klávesové zkratky, neovladatelný
gmusicbrowser	V Xubuntu nelze ovládat
Parole	Pouze několik ovladatelných funkcí
Rhythmbox	V pořádku
SMPlayer	V pořádku
Totem	Nepodporuje některé zkratky(shuffle, loop), po maximalizaci nereaguje na další příkazy
VLC	V pořádku

Tabulka 5.1: Výsledky testování multimediálních aplikací

Nestandardní chování se projevilo u aplikací pro prohlížení prezentací a programu Totem. Po přechodu do módu celé obrazovky nedokázal program xdotool zaslat zkratky. Program je tedy přednastaven pro tyto aplikace: Amarok, Audacious, Banshee, Parole, Rhythmbox, SMPlayer a VLC. Podpora dalších programů a oprava nedostatků se zasíláním zkratk bude v případné další verzi programu.

## 6 Závěr

Cílů této práce bylo několik. Seznámil jsem se dopodrobna s vývojem aplikací na mobilní telefony s OS Android a jednu takovou aplikaci jsem sám implementoval. Tato aplikace úspěšně komunikuje přes Bluetooth s PC s operačním systémem. Navrhl jsem systém ovládání, který využívá emulátoru klávesnice xdotool, kterým lze odesílat klávesové zkratky oknům aplikací. Tento systém spoléhá na přítomnost klávesových zkratk v ovládaném programu. Cíl práce, tzn. vytvořit funkční aplikaci na operační systém Android, která dokáže ovládat multimediální aplikace běžící na PC s GNU/Linux, byl splněn. Aplikace byla podrobena řadě různých testů a s menšími výhradami ji lze považovat za efektivně a plně fungující.

Hlavním přínosem vytvořené aplikace je její praktické využití. Díky tomuto programu je možné využít mobilní telefon jako dálkové ovládání multimediálních aplikací na PC bez nutnosti zakoupení samostatného přístroje majícího takovouto funkci. S neustálým zdokonalováním a zefektivňováním obslužnosti PC a zároveň s rostoucí náročností uživatelů nabývá takováto aplikace na praktickém významu.

### 6.1 Možnosti rozšíření a dalšího vývoje projektu

Program by mohl být po upravení nedostatků, které vyšly najevo při testování, nasazen na Android market. Pro její spjatost s operačním systémem GNU/Linux by bylo možné ji vydat jako *open-source* aplikaci, stejně tak její serverovou část na Linux. Aplikace nebyla vyzkoušena na Mac OS X od firmy Apple, ale port na tento OS by měl proběhnout bez obtíží, protože nástroj xdotool je kompatibilní s tímto systémem a Qt toolkit rovněž. Rozšíření aplikace na OS Windows 7 by už byl problematictější, ale ne nemožný. Musela by být použita komunikace pomocí jiné knihovny Bluetooth a vytvořen nástroj podobný xdotool. Použití Qt toolkitu pro jeho multiplatformnost otevírá více možností pro větší nasazení tohoto softwaru.

Aplikace by mohla být také rozšířena o další funkce a přiblížit se tak konkurenci, jakou je např. Gmote nebo RemoteDroid. Program Gmote podporuje přímé procházení souborů v PC pomocí mobilního telefonu. Tato funkce by byla vhodná pro další rozšíření TuXcontrol. Dalšími rozšířeními by mohlo být: Spouštění aplikací z mobilního přístroje, používání aplikace jako klávesnice nebo myši a další.

V této fázi vývoje je TuXcontrol použitelnou aplikací pro jednoduché ovládání multimediálních programů na PC, ale s potenciálem stát se využívanou aplikací pro mnoho uživatelů.

# Literatura

- [1] ALLEN, Sarah; GRAUPERA, Vidal; LUNDRIGAN, Lee. *Pro Smartphone Cross-Platform Development : iPhone, BlackBerry, Windows Mobile and Android Development and Distribution*. [s.l.] : Apress, 2010. 275 s. ISBN 978-1-4302-2868-4.
- [2] KAMMER, David, et al. *Bluetooth Application Developer's Guide : The Short Range Interconnect Solution*. Rockland (MA) : Syngress, 2002. 520 s. ISBN 1-928994-42-3.
- [3] GARGENTA, Marko. *Learning Android : Building Applications for the Android Market*. [s.l.] : O'Reilly Media, 2011. 268 s. ISBN 978-1-4493-9050-1.
- [4] DARCEY, Lauren; CONDER, Shane. *Sams Teach Yourself Android Application Development in 24 Hours*. 2nd ed. [s.l.] : Sams, 2010. 480 s. ISBN 978-0-321-67335-0.
- [5] *Android Developers* [online]. c2011 [cit. 2011-05-14]. The Developer's Guide. Dostupné z WWW: <<http://developer.android.com/guide/index.html>>.
- [6] HUANG, Albert S.; RUDOLPH, Larry. *Bluetooth Essentials for Programmers*. 1st ed. Cambridge (Massachussets) : Cambridge University Press, 2007. 208 s. ISBN 978-0-521-70375-8.
- [7] *Zenith* [online]. c2011 [cit. 2011-05-13]. About Us - History. Dostupné z WWW: <<http://www.zenith.com/about/>>.
- [8] *Electronic House* [online]. c2010 [cit. 2011-05-13]. 1955 - Zenith's Flashmatic Remote Control. Dostupné z WWW: <<http://www.electronichouse.com/slideshow/category/2206/424>>.
- [9] *Semicomplete* [online]. Fri, 12 Nov 2010 [cit. 2011-05-12]. Xdotool - fake keyboard/mouse input, window management, and more. Dostupné z WWW: <<http://www.semicomplete.com/projects/xdotool/#id54264>>.
- [10] GALITZ, Wilbert O. *The Essential Guide to User Interface Design : An Introduction to GUI Design Principles and Techniques*. 3rd ed. Indianapolis (Indiana) : Wiley Publishing, 2007. 888 s. ISBN 978-0-470-05342-3
- [11] *Trolltech* [online]. 2007 [cit. 2011-05-12]. System Tray Icon Example. Dostupné z WWW: <<http://doc.trolltech.com/4.2/desktop-systray.html>>.
- [12] *Android Developers* [online]. c2009 [cit. 2011-05-12]. BluetoothChat - Bluetooth Chat. Dostupné z WWW: <<http://developer.android.com/resources/samples/BluetoothChat/index.html>>.
- [13] HUMBAD, Shailesh N. *Somacn* [online]. 2004-11-20, 2009-03-19 [cit. 2011-05-13]. Generate GUID Online. Dostupné z WWW: <<http://www.somacn.com/p113.php>>
- [14] AUZA, Jun. *TechSource* [online]. 4/29/2008 [cit. 2011-05-14]. 12 of the Best Media Players for Linux. Dostupné z WWW: <<http://www.junauza.com/2008/04/12-of-best-media-players-for-linux.html>>.

# Seznam příloh

Příloha 1. Manuál k instalaci aplikací

Příloha 2. Dotazník pro testování uživatelské přívětivosti s výsledky

Příloha 3. DVD

# Příloha 1

## Manuál k instalaci aplikací

### Instalace serveru na PC s GNU/Linux

Pro správný chod aplikace musíte nejprve ve své distribuci nainstalovat nástroj `xdotool`.  
Uživatelé systémů Debian a Ubuntu mohou tento nástroj doinstalovat příkazem:

```
apt-get install xdotool
```

Ostatní uživatelé jej mohou stáhnout z internetových stránek:

<http://www.semicomplete.com/projects/xdotool/#id54264>

Aplikace musí být nejprve přeložena příkazy (je potřeba mít nainstalovaný Qt toolkit a nástroj `make`):

1. `qmake -project`
2. `qmake`
3. `make`

Po úspěšném přeložení, můžete spustit soubor `TuXcontrol` a používat tuto aplikaci.

### Spuštění klienta na mobilním zařízení s OS Android

Pro spuštění aplikace na mobilním zařízení budete potřebovat tyto nástroje a pomůcky: USB kabel k vašemu přístroji, mobilní přístroj s OS Android nastavený v režimu ladění přes USB, Vývojové prostředí Eclipse s nainstalovaným pluginem Android ADT. Dále postupujte podle těchto kroků :

1. Otevřete projekt `TuXcontrol` v IDE Eclipse
2. Připojte svůj mobilní přístroj k PC a nastavte režim ladění přes USB
3. V Eclipse zvolte z nabídky Run možnost Run

Aplikace se nyní spustí na vašem mobilním přístroji a je připravena k používání.

# Příloha 2

## Dotazník pro testování uživatelské přívětivosti s výsledky

### Dotazník

**1) Měl/a jste při úkolech potřebu použít nápovědu?**

- a) Ne, nebylo potřeba.
- b) Ne, ale uvažoval jsem o tom.
- c) Ano, použil jsem.

**2) Byl/a byste si schopen/a nastavit sám/a zkratky pro novou aplikaci?**

- a) Ano, není to pro mě problém.
- b) Ano, ale raději nic nenastavuji.
- c) Nepřeji si nic nastavovat, proto bych chtěl/a obsáhlou databázi aplikací.

**3) Vyhovuje vám minimalizace aplikace do Tray oblasti?**

- a) Ne.
- b) Je mi to jedno.
- c) Ano.

**4) Rozumíte ikonám dálkového ovládní?**

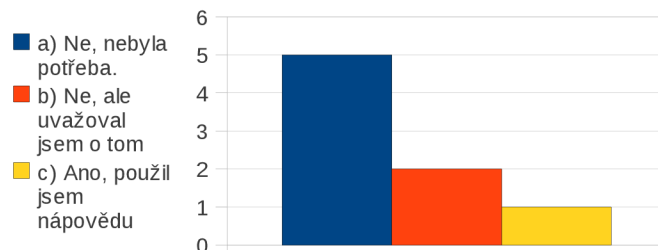
- a) Ano, jsem s nimi obeznámen/a.
- b) Rozuměl/a jsem všem základním, ale některé pokročilé funkce jsem nepoznal/a.
- c) Ne, potřeboval/a bych nápovědu.

**5) Využil/a byste tuto aplikaci?**

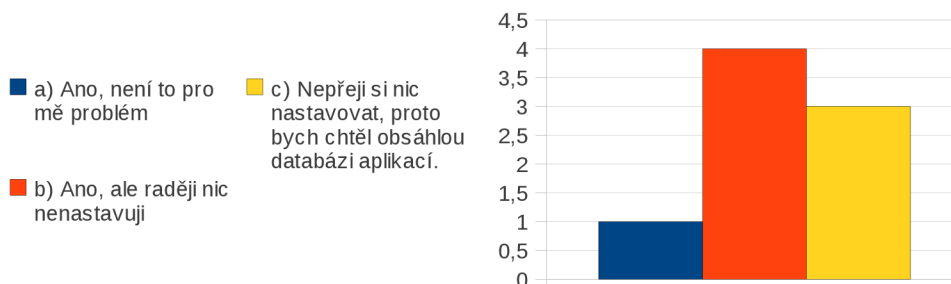
- a) Ano, zaujala mě a určitě bych jí využil/a.
- b) Možná ano.
- c) Ne.

# Výsledky

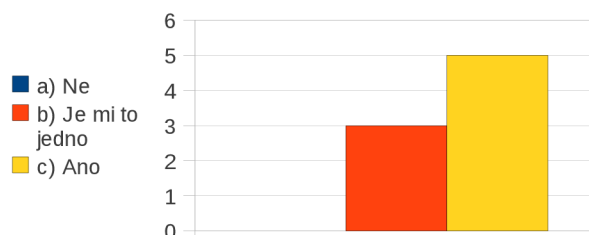
## 1) Měl jste při úkolech potřebu použít nápovědu?



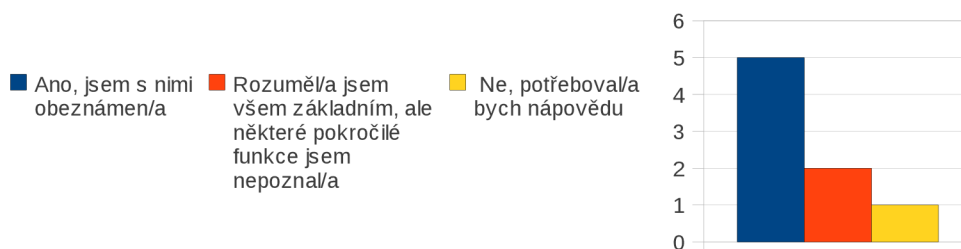
## 2) Byl byste si schopen nastavit sám zkratky pro novou aplikaci?



## 3) Vyhovuje vám minimalizace aplikace do Tray oblasti?



## 4) Rozumíte ikonám dálkového ovládání?



## 5) Využil/a byste tuto aplikaci?

