

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## BLUETOOTH DÁLKOVÝ OVLADAČ

BAKALÁŘSKÁ PRÁCE

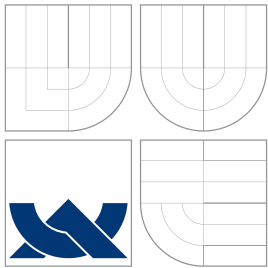
BACHELOR'S THESIS

AUTOR PRÁCE

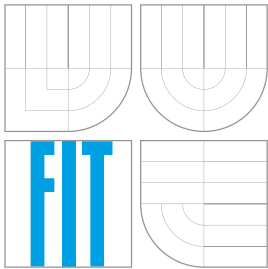
AUTHOR

MARTIN KRATOCHVÍL

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# **BLUETOOTH DÁLKOVÝ OVLADAČ**

BLUETOOTH REMOTE CONTROLLER

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**MARTIN KRATOCHVÍL**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JAROSLAV ŠKARVADA**

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Kratochvíl Martin**

Obor: Informační technologie

Téma: **Bluetooth dálkový ovladač**

Kategorie: Počítačové sítě

Pokyny:

1. Seznamte se s platformou J2ME, nejpoužívanějšími rozšiřujícími specifikacemi JSR a se způsobem programování zařízení založených na J2ME. Seznamte se s technologií Bluetooth a se standardy ETSI relevantními pro tuto práci.
2. Navrhňte systém, který umožní použít mobilní telefon jako dálkový ovladač PC. Při návrhu se zaměřte zejména na snadnou použitelnost a rozšiřitelnost systému, možnost interaktivní obousměrné komunikace mezi klientem a serverem, vícenásobný přístup a nízkou energetickou náročnost.
3. Navržený systém implementujte.

Literatura:

- Prathap Reddi, A. CSR Prabhu: Bluetooth Applications with Java & J2ME, Prentice Hall India Ltd, 2003, 390 stran
- <http://interval.cz/vyvoj-aplikaci/j2me/>

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1-2 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Škarvada Jaroslav, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačových systémů a sítí  
602 00 Brno, Božetěchova 2



---

doc. Ing. Zdeněk Kotásek, CSc.  
vedoucí ústavu

**LICENČNÍ SMLOUVA  
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Martin Kratochvíl**  
Id studenta: 78913  
Bytem: Močovice 150, 286 01 Čáslav  
Narozen: 13. 03. 1986, Prostějov  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1  
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Bluetooth dálkový ovladač  
Vedoucí/školitel VŠKP: Škarvada Jaroslav, Ing.  
Ústav: Ústav počítačových systémů  
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1  
elektronické formě                počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel

*Krenbochová*  
.....

Autor

## **Abstrakt**

Cílem práce bylo vytvoření univerzálního systému pro ovládání počítače dálkovým ovladačem v podobě mobilního telefonu. Systém využívá technologie Bluetooth pro bezdrátovou komunikaci. Zaměřuje se na snadnou rozšiřitelnost a konfigurovatelnost, jednoduché použití a nízkou energetickou spotřebu na straně mobilního telefonu.

## **Klíčová slova**

Bluetooth, server, GNU/Linux, BlueZ, MPD, klient, dálkový ovladač, J2ME, mobilní telefon

## **Abstract**

The goal of this thesis was to create an universal system for a remote computer control by a cellular phone. The system uses Bluetooth technology for wireless communication. It focuses on extensibility and ease of configuration, simple usage and low power requirement on the cellphone.

## **Keywords**

Bluetooth, server, GNU/Linux, BlueZ, MPD, client, remote controller, J2ME, cellular phone

## **Citace**

Martin Kratochvíl: Bluetooth dálkový ovladač, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Bluetooth dálkový ovladač

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jaroslava Škarvady

.....

Martin Kratochvíl

13. května 2008

## Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Jaroslavu Škarvadovi za jeho odbornou pomoc při zpracování práce.

© Martin Kratochvíl, 2008.

*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

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Bluetooth</b>	<b>4</b>
2.1	Vlastnosti . . . . .	4
2.2	Protokoly . . . . .	5
2.2.1	HCI . . . . .	5
2.2.2	L2CAP . . . . .	5
2.2.3	BNET . . . . .	5
2.2.4	RFCOMM . . . . .	5
2.2.5	SDP . . . . .	6
2.2.6	OBEX . . . . .	6
2.3	Profily . . . . .	6
2.3.1	GAP . . . . .	6
2.3.2	PAN . . . . .	6
2.3.3	HID . . . . .	7
2.3.4	SPP . . . . .	7
2.3.5	DUN . . . . .	7
2.3.6	OPP . . . . .	7
<b>3</b>	<b>J2ME</b>	<b>8</b>
3.1	Konfigurace . . . . .	8
3.1.1	CDC . . . . .	9
3.1.2	CLDC . . . . .	9
3.2	Virtuální stroje . . . . .	9
3.2.1	KVM . . . . .	9
3.2.2	CVM . . . . .	10
3.3	Profily . . . . .	10
3.3.1	Základní profil . . . . .	10
3.3.2	Osobní profil . . . . .	10
3.3.3	RMI profil . . . . .	10
3.3.4	MIDP . . . . .	10
3.3.5	PDA profil . . . . .	11
3.3.6	Java API pro Bluetooth . . . . .	11
<b>4</b>	<b>Bluetooth a GNU/Linux</b>	<b>12</b>
4.1	BlueZ . . . . .	12
4.2	Affix . . . . .	12



<b>5</b>	<b>Návrh systému</b>	<b>13</b>
5.1	Využití . . . . .	13
5.2	Funkčnost . . . . .	13
5.2.1	Ovládací obrazovky . . . . .	14
5.3	Komunikační protokol . . . . .	14
5.3.1	Získání ovládacích obrazovek . . . . .	14
5.3.2	Výběr určité ovládací obrazovky . . . . .	14
5.3.3	Práce s vybranou ovládací obrazovkou . . . . .	15
5.4	Konfigurační soubory . . . . .	16
<b>6</b>	<b>Implementace systému</b>	<b>18</b>
6.1	Klient . . . . .	18
6.1.1	Jádro Bluetooth komunikace . . . . .	18
6.1.2	Klientská aplikace . . . . .	21
6.1.3	Podpůrné třídy . . . . .	22
6.2	Server . . . . .	23
6.2.1	Spuštění/ukončení aplikace . . . . .	23
6.2.2	Bluetooth . . . . .	24
6.2.3	Dispečer . . . . .	24
6.2.4	Konfigurační soubory . . . . .	24
6.2.5	Moduly . . . . .	24
<b>7</b>	<b>Závěr</b>	<b>26</b>
<b>A</b>	<b>Specifikace funkcí modulů</b>	<b>28</b>
A.1	Modul MPD . . . . .	28
A.1.1	Řídící funkce . . . . .	28
A.1.2	Funkce pro získání zpětné informace . . . . .	28
A.2	Modul XTE . . . . .	29
A.2.1	Řídící funkce . . . . .	29
A.2.2	Funkce pro získání zpětné informace . . . . .	29
<b>B</b>	<b>Ikony podporované klientem</b>	<b>30</b>
<b>C</b>	<b>Obsah CD</b>	<b>31</b>

# Kapitola 1

## Úvod

Tématem této práce je vytvoření univerzálního systému pro bezdrátové ovládání počítače pomocí mobilního telefonu za využití technologie Bluetooth. Navržený systém se zabývá zejména stránkou univerzálnosti (systém je jednoduše rozšiřitelný), velmi jednoduchou použitelností (systém je připraven k použití během velmi krátkého času) a snížením spotřeby energie na straně mobilního telefonu. Systém je možné použít na počítačích s operačním systémem GNU/Linux s dálkovým ovladačem, což je libovolný mobilní telefon s podporou Javy pro malá zařízení (J2ME) a s podporou Bluetooth (standard JSR-82).

V kapitole 2 je rozebrána technologie Bluetooth, její základní charakteristika, vlastnosti a protokoly, které se v souvislosti s ní používají. Využití technologie Bluetooth je velmi široké, proto je zde uveden jen základní přehled.

Další technologií využívanou systémem je technologie J2ME, která je rozebrána ve 3. kapitole. Tuto technologii je možné využít v širokém spektru zařízení, proto je zde popsáno základní rozdělení do konfigurací (více kapitola 3.1) a detailnější rozdělení do profilů (viz kapitola 3.3).

V kapitole 4 je nastíněn způsob využití technologie Bluetooth v operačním systému GNU/Linux.

Obsahem 5. kapitoly je návrh systému a hlavně je zde navržen vlastní komunikační protokol mezi dálkovým ovladačem a ovládaným počítačem (více kapitola 5.3). S tím souvisí i návrh podoby konfiguračních souborů pro definování ovládacích a ovládaných prvků (podrobnosti v kapitole 5.4).

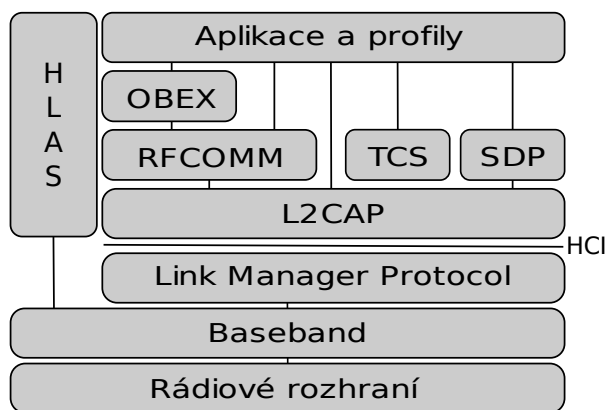
V kapitole 6 je ve dvou částech popsána implementace systému. První vysvětluje implementaci klienta (program určený pro mobilní telefon) a druhá implementaci serveru (program spuštěný na ovládaném počítači a zajišťující provádění příkazů z dálkového ovladače).

## Kapitola 2

# Bluetooth

Bluetooth [1] je nová technologie využívající rádiového spojení s krátkým dosahem k nahrazení kabelů propojujících různá zařízení (například osobní počítač, PDA, mobilní telefon, tiskárny a další). Tato technologie je definována standardem IEEE 802.15.1 [4].

Bluetooth patří do skupiny sítí WPAN (*Wireless Personal Area Network*), což jsou osobní bezdrátové sítě s malým dosahem. Tato technologie získala název odvozením od přezdívky dánského krále Haralda II, přezdívaného „Modrozub“, údajně podle jeho velké záliby v borůvkách a ostružinách. Harald díky svým diplomatickým schopnostem sjednotil skandinávský lid. Stejně tak i tato nová technologie má sloužit k usnadnění komunikace. Struktura technologie Bluetooth – rozhraní mezi hardwarem zařízení a jednotlivými aplikacemi je zobrazeno na obrázku 2.1.



Obrázek 2.1: Bluetooth rozhraní

### 2.1 Vlastnosti

Mezi základní vlastnosti Bluetooth patří:

- Rádiová komunikace v bezlicenčním pásmu ISM na frekvenci 2,4 GHz s využitím metody rozprostřeného spektra s přeskokováním kmitočtů FHSS (*Frequency Hopping Spread Spectrum*), kdy rádiový signál velmi rychle (1600-krát za sekundu) náhodně přeskakuje mezi 79 jedno-MHz kanály.

- Nízká energetická spotřeba díky nízkému vysílacímu výkonu.
- Teoretická rychlost 1 Mbit/s (skutečná propustnost dat se pohybuje okolo 720 kbit/s).
- Podpora různých přenosů, nejen datových, ale i hlasových.

Každé Bluetooth zařízení spadá do jedné z následujících výkonnostních kategorií.

Kategorie	Maximální povolený vyzářený výkon	Dosah (přibližně)
Class 1	100 mW (20 dBm)	100 metrů
Class 2	2,5 mW (4 dBm)	10 metrů
Class 3	1 mW (0 dBm)	1 metr

## 2.2 Protokoly

Bluetooth používá řadu různých počítačových protokolů [6]. Každý z těchto protokolů poskytuje možnosti komunikace na různé úrovni abstrakce, následuje krátký přehled.

### 2.2.1 HCI

HCI (*Host Controller Interface*) je určité rozhraní zajišťující komunikaci nižších vrstev s vyššími.

### 2.2.2 L2CAP

L2CAP (*Logical Link Control and Adaptation Protocol*) je protokol, který poskytuje jednotnou správu přenosu ke skupině jiných Bluetooth zařízení. Zapouzdřuje pakety do formátu vhodného pro nižší vrstvy a multiplexuje spojení tak, aby mohlo být využíváno více aplikacemi. Základní vlastnosti L2CAP jsou:

- Poskytuje vyšším vrstvám spojově a bezspojově orientované datové služby.
- Maximální velikost L2CAP paketu je 64 KB.
- Kvalita služeb - monitorování prostředků pro zajištění parametrů dohodnutých při navázání spojení.
- Přenášení dat mezi různými protokoly na vyšších vrstvách.

### 2.2.3 BNET

BNET (*Bluetooth Network Emulation Protocol*) je využíván v PAN profilech a slouží k dopravě síťových paketů.

### 2.2.4 RFCOMM

RFCOMM (*Radio Frequency Communication*) protokol poskytuje emulaci sériového portu pomocí funkcí L2CAP protokolu. RFCOMM umožňuje až 60 současných připojení Bluetooth zařízení. Protokol vychází ze standardu ETSI TS 07.10.

### 2.2.5 SDP

SDP (*Service Discovery Protocol*) je protokol, který umožňuje zjistit poskytované služby, včetně jejich vlastností.

### 2.2.6 OBEX

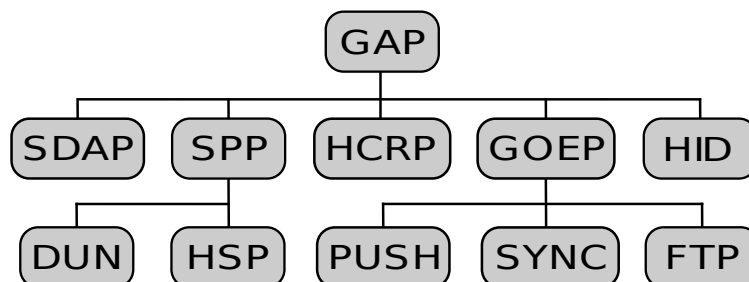
OBEX (*Object Exchange*) je komunikační protokol, který usnadňuje výměnu binárních souborů mezi zařízeními.

## 2.3 Profily

Profily slouží k reprezentaci určitého způsobu využití technologie Bluetooth tak, aby se předešlo možné nekompatibilitě vyvíjených programů u Bluetooth zařízení od různých výrobců. Každé Bluetooth zařízení musí umět interpretovat jednotlivé profily, proto jsou určitá pravidla pro jejich specifikaci [5]. Jejich specifikace musí zahrnovat:

- Závislosti na jiných profilech.
- Návrh formátu uživatelského rozhraní.
- Specifické části vrstev Bluetooth protokolu použité v profilu.

Dále následuje přehled několika Bluetooth profilů a jejich hierarchie (obrázek 2.2).



#### 2.3.1 GAP

GAP (*Generic Access Profile*) je základní profil pro všechny ostatní, aby se tím zajistila všeobecná kompatibilita. Zajišťuje například navázání spojení.

#### 2.3.2 PAN

PAN (*Personal Area Networking Profile*) popisuje, jak můžou dvě nebo více zařízení s podporou Bluetooth vytvořit ad-hoc síť a jak pomocí stejného mechanismu může být přistupováno ke vzdálené síti přes přístupový bod.

### 2.3.3 HID

HID (*Human Interface Device Profile*) poskytuje podporu pro zařízení jako jsou klávesnice, myši, joysticky a další. Je navržen na poskytování spojení s nízkým zpožděním a s nízkými energetickými požadavky. Bluetooth HID vychází z HID protokolu definovaného pro USB, což zvyšuje kompatibilitu v různých operačních systémech. Zařízení podporující USB HID podporuje i Bluetooth HID.

### 2.3.4 SPP

SPP (*Serial Port Profile*) používá RFCOMM protokol. Emuluje sériovou linku, a tak poskytuje jednoduchou bezdrátovou implementaci náhrady za existující RS-232 sériovou komunikační linku. Poskytuje základ pro další profily (DUN, FAX, HSP, AVRCP).

### 2.3.5 DUN

DUN (*Dial-Up Networking Profile*) je profil, který poskytuje standardní přístup k Internetu a jiným vytáčeným službám přes Bluetooth. Scénář možného použití je například přístup k Internetu na notebooku vytáčeným spojením bezdrátově přes mobilní telefon.

### 2.3.6 OPP

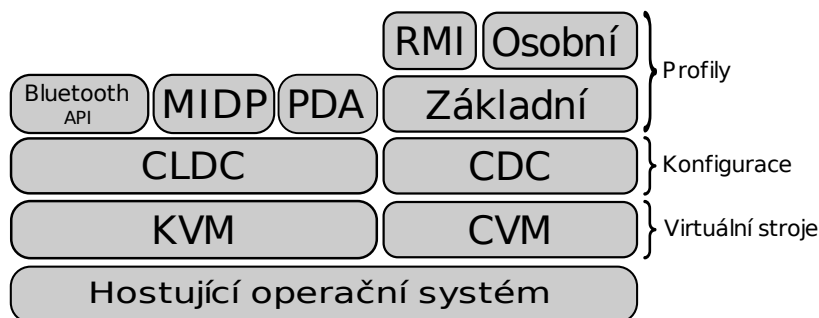
OPP (*Object Push Profile*) je základní profil pro posílání „objektů“ jako jsou obrázky, elektronické vizitky a další. OPP používá API z profilu OBEX. Přenos je vždy iniciován odesílatelem (klient) a ne příjemcem (server).

# Kapitola 3

## J2ME

Java 2 Micro Edition (J2ME) je verze produktu Java pro malá zařízení, jakými jsou například mobilní telefony, osobní digitální asistenti (*PDA*), pagery a další [2]. Vývoj J2ME zajišťuje projekt JCP (*Java Community Process*). JCP umožňuje každému podílet se na tomto vývoji. Struktura J2ME je zobrazena na obrázku 3.1. J2ME definuje tyto komponenty:

- Java virtuální stroje pro různá zařízení s odlišnými nároky.
- API a knihovny spustitelné na každém virtuálním stroji, zvané konfigurace a profily.
- Nástroje pro vývoj a nastavení zařízení.



Obrázek 3.1: Základní členění J2ME

Velmi důležité je rozlišovat mezi konfigurací a profilem, následuje vysvětlení pojmů.

### 3.1 Konfigurace

Malá zařízení se liší svými funkcemi a vlastnostmi, přesto používají podobné procesory a mají podobné množství paměti. Proto vznikly konfigurace. Konfigurace slouží ke členění jednotlivých zařízení na základě dostupné paměti a výkonu procesoru. Z těchto informací konfigurace zjistí, co všechno je zařízením podporováno, knihovny jazyka Java, API a rysy Java virtuálního stroje. Existují dvě standardní konfigurace CLDC (*Connected Limited Device Configuration*) a CDC (*Connected Device Configuration*).

### 3.1.1 CDC

CDC (standard JSR-36) je konfigurace určena pro relativně výkonná zařízení, s možností připojení k síti [8]. Patří sem zařízení jako navigační systémy pro vozidla, domácí spotřebiče a další. CDC používá verzi Java virtuálního stroje podobného tomu, který se používá u J2SE (*Java 2 Standard Edition*), rozdíl je v paměti a zobrazovacích schopnostech zařízení. Požadavky na CDC zařízení dle oficiální specifikace J2ME jsou:

- Minimálně 512 KB paměti ROM.
- Minimálně 256 KB paměti RAM.
- Připojení k nějakému typu sítě.
- Podpora kompletní implementace Java virtuálního stroje.
- Uživatelské rozhraní s různým stupněm propracovanosti.

### 3.1.2 CLDC

CLDC (standard JSR-30) je typ konfigurace obvyklejší pro J2ME [9]. Je to konfigurace s mnohem menšími požadavky na zařízení. CLDC je určena pro zařízení jako jsou PDA, osobní organizátor, mobilní telefon a další. Požadavky pro J2ME CLDC podle oficiální specifikace J2ME jsou:

- 128 – 512 KB celkové paměti, z toho méně než 256 KB ROM/Flash a méně než 256 KB RAM. Ve většině případů mají zařízení více ROM paměti než RAM nebo Flash.
- Omezený zdroj energie, často bateriové napájení.
- Připojení k nějakému typu sítě, často s omezenou šířkou pásma (9600 bps nebo méně).
- Různě propracované uživatelské rozhraní.

## 3.2 Virtuální stroje

Jak už bylo zmíněno, každá konfigurace stanovuje vlastní skupinu podporovaných rysů Java virtuálního stroje. To znamená, že pro každý z nich je potřeba jiný Java virtuální stroj. Virtuální stroj pro CLDC je KVM (*Kilobyte Virtual Machine*) a virtuální stroj pro CDC je CVM (*Compact Virtual Machine*).

### 3.2.1 KVM

KVM je virtuální stroj vytvořen speciálně pro malá zařízení s omezenými zdroji. Přesto se jedná o skutečný virtuální stroj podle specifikace pro virtuální stroje Javy. Obsahuje specifické odchylky, nutné pro správné fungování na malých zařízeních.



### 3.2.2 CVM

CVM je virtuální stroj vytvořený pro větší zařízení, například ta, která používají konfiguraci CDC. Podporuje všechny rysy JVM (*Java Virtual Machine*).

## 3.3 Profily

Profil je vlastně API (*sada programových rozhraní*) tvořící nadstavbu konfigurace. Poskytuje programu přístup ke specifickým vlastnostem daného zařízení. Následuje přehled několika zástupců profilů pro obě konfigurace (CLDC, CDC). Profilů je mnohem více, zde jsou uvedeny jen ty základní a ty, které se přímo týkají této práce.

### 3.3.1 Základní profil

Základní profil (*Foundation Profile*) je vázán na CDC konfiguraci a slouží jako základ pro další profily. Základní profil nedefinuje žádné API pro uživatelské rozhraní, důvodem je typ zařízení, pro které je CDC určeno (primárně pro autonomní zařízení jako jsou set-top boxy, TV apod.). Tato zařízení nejsou na použití GUI tak fixována.

### 3.3.2 Osobní profil

Osobní profil (*Personal Profile*) je rozšíření Základního profilu o grafické uživatelské rozhraní. Jedná se o kompletní prostředí s plnou podporou AWT (*Abstract Window Toolkit*). Tento profil je závislý na základním profilu.

### 3.3.3 RMI profil

RMI profil (*Remote Method Invocation*) je další rozšíření Základního profilu. Poskytuje prostředí pro vzdálené volání metod Javy kompatibilní s RMI API ze standardní edice Javy (J2SE) [11].

### 3.3.4 MIDP

MIDP (*Mobile Information Device Profile*) je profil poskytující API v zařízeních jako mobilní telefony, tedy zařízeních s konfigurací CLDC. Obsahuje třídy pro uživatelské rozhraní, práci v síti a trvalé ukládání dat. Aplikacím pro tuto kategorii se říká **Midlet** podle základní třídy profilu MIDP. Zařízení s MIDP musí splňovat následující parametry:

- Velikost displeje minimálně 96x54 pixelů, 2 barvy.
- Klávesnice, případně dotyková obrazovka.
- Minimálně 32 KB paměti pro samotnou Javu.
- Minimálně 128 KB stálé paměti pro MIDP komponenty.
- Minimálně 8 KB stálé paměti pro ukládání dat aplikací.
- Obousměrné síťové spojení.

Profil MIDP řeší správu průběhu aplikací, uživatelské rozhraní, připojitelnost k síti a ukládání dat v zařízení.

### 3.3.5 PDA profil

PDA profil, jak už název napovídá, je rozšíření určené pro zařízení jako PDA, zařízení s větším displejem, definuje uživatelské rozhraní (podskupina AWT) a API pro ukládání dat v zařízení.

### 3.3.6 Java API pro Bluetooth

Bluetooth je rozšiřujícím prostředkem pro bezdrátové propojení malých zařízení jako mobilní telefony, pagery, atd. Specifikace Java API pro Bluetooth (JSR-82 [10]) je určena pro konfiguraci CLDC a obsahuje podporu základních Bluetooth protokolů. Mezi tyto protokoly patří například RFCOMM, OBEX a SDP. Nároky na zařízení s těmito API jsou:

- Minimálně 512 KB celkové paměti (bez paměťových nároků samotné aplikace).
- Bluetooth síťové připojení.

## Kapitola 4

# Bluetooth a GNU/Linux

Bluetooth rozhraní poskytuje implementaci sady protokolů a nástrojů pro použití v určitém operačním systému. V tuto dobu jsou pro operační systém GNU/Linux k dispozici dvě různé implementace Bluetooth rozhraní:

- BlueZ, který je obsažen v oficiálním linuxovém jádře.
- Affix, vyvinutý Nokia Research Center.

### 4.1 BlueZ

BlueZ je oficiální Bluetooth rozhraní pro GNU/Linux [7]. Je zaměřen na implementaci Bluetooth bezdrátových standardů specifických pro GNU/Linux.

BlueZ byl původně vyvinut společností Qualcomm a je dostupný v linuxovém jádře ve verzi 2.4.6 a vyšší.

BlueZ poskytuje podporu pro všechny základní Bluetooth protokoly jako jsou L2CAP, SCO, RFCOMM, BNET, HIDP. Jedná se o kompletně modulární implementaci, která podporuje více Bluetooth zařízení a zároveň zajišťuje opravdovou hardwarovou abstrakci, rozhraní pro standardní sokety, atd.

BlueZ je dostupný pro mnoho různých hardwarových architektur jako x86, AMD64, SPARC, PowerPC, ARM, Hitachi/Renesas SH processors, Motorola DragonBall, atd.

### 4.2 Affix

Affix je robustní open-source Bluetooth rozhraní pro operační systém GNU/Linux vyvinutý Nokia Research Center [3]. Podporované Bluetooth protokoly jsou: HCI, L2CAP, RFCOMM, SDP, BNEP a profily: GAP, SDAP, SPP, GEOP, FTP, OPP, PAN, LAN, DUN, HID, FAX a BPP.

Affix je implementován v jazyce C. Je dostupný pro několik hardwarových architektur jako jsou x86, SPARC, ARM, PowerPC, atd.

## Kapitola 5

# Návrh systému

Cílem této práce je vytvořit Bluetooth dálkový ovladač. Jinými slovy systém pro ovládání počítače pomocí přenosného zařízení, např. mobilního telefonu a to bezdrátově s využitím technologie Bluetooth. Podobné aplikace již jistě existují, tento systém se ale snaží řešit nedostatky ostatních řešení, jako je například udržování Bluetooth spojení po celou dobu komunikace. Technologie Bluetooth činí obrovské energetické nároky na mobilní zařízení, která jsou napájena pouze z baterie. Proto se tento systém snaží omezit spotřebu energie přerušováním Bluetooth spojení, kdykoli není potřeba. Další rozdílem je možnost připojení více klientů zároveň.

### 5.1 Využití

Systém může sloužit při prezentacích jako nástroj pro přepínání snímků. Může sloužit také k zábavě, kdy ho využijeme jako náhradu za dálkový ovladač televizoru, DVD přehrávače nebo Hi-Fi systému. Navíc oproti jmenovaným zařízením má tu výhodu, že je univerzální, tzn. pomocí jednoho zařízení, které je jednoduše konfigurovatelné, je možné ovládat libovolný program na osobním počítači, a tak jím nahradit všechny zmiňované přístroje. Pomocí pluginů je možné systém rozšířit o ovládání další aplikace. Systém navíc umožňuje obousměrnou komunikaci, tu lze využít ke zpětné vazbě. Například ovládáme-li přehrávač hudby a zvýšíme hlasitost, systém umožňuje zobrazit na displeji mobilního zařízení informaci o aktuálním stavu hlasitosti.

### 5.2 Funkčnost

Systém se skládá ze serverové části, která je spuštěna na ovládaném počítači a provádí příkazy přijaté od „dálkového ovladače“ (klientské části, spuštěné na mobilním telefonu). Základními požadavky na klienta (mobilní telefon) jsou podpora J2ME (Java edice pro malá zařízení) a standardu JSR-82 (Java Bluetooth API).

Jádro systému jako takové zajišťuje pouze komunikaci a nezajímá se o to, jaké aplikace a jak jsou ovládány. Od toho jsou zde pluginy, které zajišťují samotné ovládání počítače a aplikací na něm. Vzhledem k rozdílným ovládacím prvkům na zařízeních od různých výrobců umožňuje systém výběr z  $N$  tzv. ovládacích obrazovek. Na každé obrazovce je možno posílat příkazy 12-ti klávesami a to klávesami s čísly nula až devět a znaky hvězdičky a dvojitého křížku (to jsou základní klávesy, kterými jsou vybaveny všechny mobilní telefony).

### 5.2.1 Ovládací obrazovky

Ovládací obrazovky slouží k definování, která klávesa provede který příkaz s využitím určitého pluginu. Obrazovky se definují pomocí konfiguračních souborů a jsou uloženy na serveru. Komunikační protokol umožňuje jejich získání. Tak i klient, který se spojuje se serverem poprvé, má k dispozici informaci o tom, které aplikace a jak může ovládat.

## 5.3 Komunikační protokol

Komunikace probíhá pomocí textového protokolu a to tak, že klient vyšle požadavek a server po zpracování generuje odpověď. Komunikace se skládá z několika fází:

- Získání seznamu ovládacích obrazovek definovaných na serveru.
- Výběr určité z nich.
- Práce s vybranou ovládací obrazovkou.

Je samozřejmě možné kdykoli přejít zpět a vybrat jinou obrazovku nebo i jiný dostupný server v okolí.

### 5.3.1 Získání ovládacích obrazovek

Po připojení k serveru je nutné získat seznam ovládacích obrazovek, abychom vůbec věděli, které funkce nám server poskytuje. Každá ovládací obrazovka je definována pomocí svého jména a navíc je k dispozici i popis, který je přívětivější pro uživatele. Při připojení k serveru klient pošle příkaz `GET_SCREEN`s a server mu jako odpověď vygeneruje seznam obrazovek ve formátu:

```
(jméno ovládací obrazovky):(popisek)
```

Pokud je obrazovek více, jsou odděleny znakem pro konec řádku (`\n`). Nejlepší bude ukázat to na příkladu.

```
Klient: GET_SCREEN
```

```
Server: mpd:Ovládání aplikace MPD\nmplayer:Ovládání aplikace MPlayer
```

### 5.3.2 Výběr určité ovládací obrazovky

Ted, když klient zná seznam ovládacích obrazovek, je potřeba ještě zjistit, co díky nim může posílat za příkazy a jaké zpětné informace získávat. Tyto informace jsou vyžádány vždy při výběru určité ovládací obrazovky, tedy jsou získány jen pro tu určitou právě zvolenou ovládací obrazovku a to příkazem `GET_SCREEN param`, kde parametr `param` je jméno ovládací obrazovky. Jako odpověď na tento příkaz server generuje seznam definic kláves a zpětných informací oddělených znakem pro konec řádku (`\n`). Jednotlivé definice mají následující formát:

```
key:(klávesa na mobilním zařízení):(popisek)
```

```
text:(kód pluginu):(popisek):(obnovovací čas)
```

```
progress:(kód pluginu):(popisek):(obnovovací čas)
```

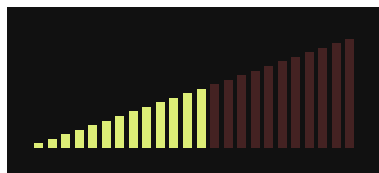
Klíčové slovo `key` s parametry klávesa na mobilním zařízení (0-9, POUND, STAR) a popisek (text, který se zobrazí v nápovědě na mobilním zařízení) slouží pro registraci obsluhy klávesy.

Klíčové slovo `text` s parametry kód pluginu (určitá funkce některého z modulů), popisek a nakonec obnovovací frekvence slouží pro registraci textové zpětné informace.

To samé platí i pro `progress`, jen se jedná místo textové o zpětnou informaci typu ukazatel průběhu. Opět lépe je to vidět na konkrétním příkladu.

```
Klient: GET_SCREEN mpd
Server: key:2:TOGGLE\n
        key:5:STOP\n
        text:song:Název písničky:50\n
        progress:volume:Hlasitost:60
```

Podle této odpovědi od serveru klient vygeneruje a zobrazí ovládací obrazovku, na které bude možno stisknout klávesy 2 a 5. Jak je vidět, klient vůbec neví, co stisk kláves provede na straně serveru, tuto informaci má server uloženou u sebe. Jedinou informací, kterou má klient k dispozici a také jedinou potřebnou, je popisek akce tlačítka a to v podobě čitelné pro uživatele nebo kód pro zobrazení ikony místo textu. Dále se budou na obrazovce zobrazovat zpětné informace a to zpětná textová informace s názvem písničky, která se bude obnovovat každých 50 sekund a informace o hlasitosti, která se zobrazí v podobě ukazatele průběhu (viz obrázek 5.1), s obnovovacím intervalem jedné minuty.



Obrázek 5.1: Ukazatel průběhu

### 5.3.3 Práce s vybranou ovládací obrazovkou

S vygenerovanou obrazovkou klient může začít využívat poskytnuté služby. Těmi je zasílání příkazů pomocí registrovaných kláves a požadavků na zpětné informace.

#### Posílání příkazů k ovládní

Při stisku registrované klávesy klient generuje například následující kód příkazu.

```
Klient: SEND mpd 2
Server: YES/NO
```

Tento kód server interpretuje tak, že byla stisknuta klávesa 2 na mobilním zařízení, na ovládací obrazovce `mpd`. Poté vykoná příslušnou akci a generuje odpověď `YES`, pokud se akci podařilo provést, jinak `NO`.

## Žádosti o zpětné informace poskytované serverem

Žádosti o zpětnou informaci jsou posílány podle požadovaného intervalu a mají podobu ukázanou na následujícím příkladu.

```
Klient: GET mpd volume
Server: volume 100
Klient: GET mpd album
Server: album Get A Grip
```

Požadavek na zpětnou informaci začíná klíčovým slovem `GET` a má dva parametry. Prvním je jméno aktuálně zvolené ovládací obrazovky a druhým klíčové slovo určující funkci pro získání požadované informace. Odpověď začíná klíčovým slovem získávané informace se získanou informací jako parametrem.

V prvním příkladu je získávána informace pro ovládací obrazovku `mpd` a tou informací je `volume` (hlasitost). Odpovědí je, že hlasitost je 100. V druhém příkladu je získávána pro tu samou ovládací obrazovku informace o albu, ze kterého je aktuálně přehrávaná písnička (`album`). Odpovědí je „Get A Grip“.

## 5.4 Konfigurační soubory

Konfigurační soubory slouží k definování podoby a funkce ovládacích obrazovek. Jeden konfigurační soubor přísluší k jedné ovládací obrazovce. Mezi povinné položky patří:

- `name` - název ovládací obrazovky pro potřeby identifikace
- `title` - popis ovládací obrazovky, ten je zobrazen na místech, kde uživatel systému pracuje s obrazovkami

Příklad definice povinných položek v konfiguračním souboru ovládací obrazovky.

```
name:mpd
title:Ovládání aplikace MPD
```

Tyto položky jsou nezbytně nutné, ale jen s nimi by ovládací obrazovka neměla žádný smysl, protože by neposkytovala žádnou funkčnost. Proto je vhodné definovat některé z nepovinných položek a těmi jsou:

- `key` - pro definici klávesy mobilního zařízení jako prvku pro provedení určité akce s ovládaným počítačem
- `text` - pro definici zpětné informace textového typu
- `progress` - pro definici zpětné informace typu ukazatel průběhu

Každá z těchto položek musí být definovaná dohodnutým formátem.

```
key:mpd:5:stop:Zastavit
```

Tento záznam přiřadí klávese 5 na mobilním zařízení funkci `stop` z modulu `mpd` a jako popisek na klientu bude zobrazen text „Zastavit“. Tedy v obecném případě má definice tvar: klíčové slovo `key`, modul, ze kterého se bude volat požadovaná funkce, klávesa na mobilním zařízení (možná klávesy jsou 0–9, `POUND` pro křížek a `STAR` pro hvězdičku), volaná funkce z předtím definovaného modulu a nakonec popisek pro zobrazení v klientské aplikaci (místo popisku může být definováno klíčové slovo, které klient interpretuje jako určitou ikonu, která vystihuje funkci klávesy i bez textu, seznam podporovaných ikon viz dodatek B).

```
text:mpd:song::50
text:mpd:album:Album:100
```

První příklad definuje zobrazování zpětné textové informace získané funkcí `song` z modulu `mpd` a s obnovovací frekvencí 50 sekund. Druhý je podobný, ale definuje zobrazení informace získané funkcí `album` ze stejného modulu s obnovovací frekvencí 100 sekund. Navíc bude před touto informací na klientu zobrazen identifikující text „Album“. Definice začíná klíčovým slovem `text`, první parametr je modul, ze kterého je funkce pro získání informace. Druhým parametrem je samotná funkce pro získání informace. Poté následuje nepovinný parametr s identifikujícím textem. Posledním parametrem je obnovovací frekvence.

```
progress:mpd:volume::30
```

Tento příklad definuje zobrazení zpětné informace typu ukazatel průběhu. Zobrazovanou informací je výsledek získaný z funkce `volume` z modulu `mpd`, bez popisku s obnovovací frekvencí 30 sekund. Parametry jsou úplně stejné jako u položky `text`.



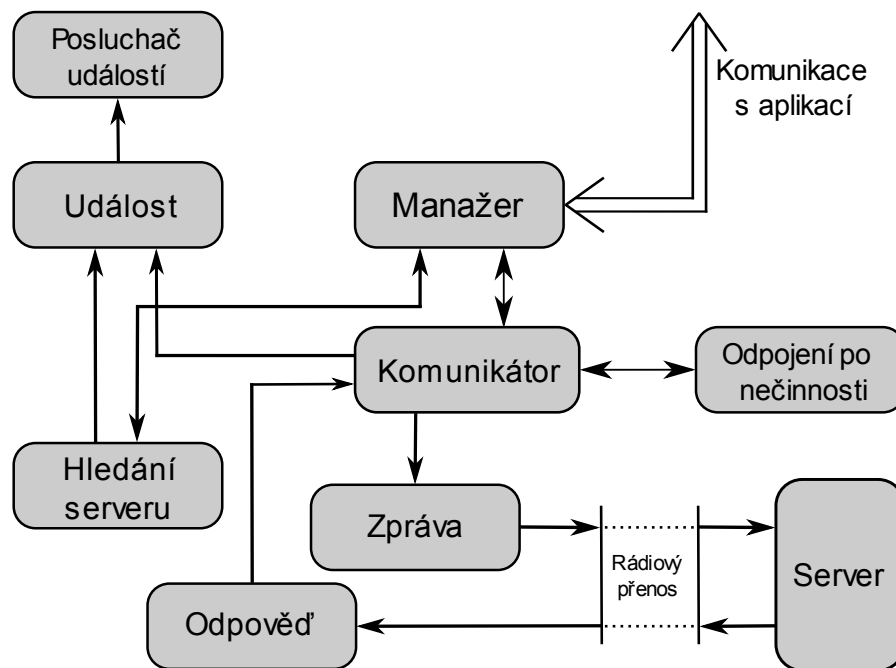
## Kapitola 6

# Implementace systému

### 6.1 Klient

Klientská aplikace je implementována v J2ME a je určena pro provoz na mobilním telefonu s podporou Java standardu JSR-82 (Java Bluetooth API). Skládá ze tří balíčků a to z jádra Bluetooth komunikace `btcore`, balíčku se samotnou aplikací `btremote` a balíčku s pomocnými třídami `support`.

#### 6.1.1 Jádro Bluetooth komunikace



Obrázek 6.1: Blokové schéma jádra Bluetooth komunikace

Aby síťová komunikace nezdržovala práci s uživatelským rozhraním, je její běh oddělen do zvláštních vláken. Proto je nutné nějak řídit synchronizaci mezi základním vláknem programu a vlákny pro síťovou komunikaci. K tomu slouží manažer, třída `BTManager`. Kromě

řízení je také důležité získat informace od serveru v samotné aplikaci a to je možné velmi jednoduše díky tomu, že `btcore` poskytuje rozhraní `BTEventListener`, které obsahuje metody, jež jsou vyvolány při určitých událostech během komunikace. Na obrázku 6.1 je znázorněno blokové schéma jádra Bluetooth komunikace.

## Manažer

Manažer tedy zajišťuje komunikace se serverem. Slouží jako rozhraní pro ovládání síťové komunikace ze samotné aplikace. Při vytvoření manažeru jsou vytvořeny dvě další vlákna, jedno pro samotnou komunikaci, třída `BTTransmitter` a druhé pro hledání okolních zařízení, třída `BTDiscoverer`. Jako řídicí třída poskytuje manažer pro každé z těchto vláken metody pro práci s nimi.

Jak již bylo zmíněno, pro získávání informací od serveru je speciální rozhraní. Ale k tomu, aby příslušná třída mohla služby tohoto rozhraní využívat, musí se zaregistrovat k poslechu tohoto rozhraní, k tomu poskytuje manažer potřebné metody.

Mimo řízení dalších vláken manažer také umožňuje ověření, zda vůbec dané zařízení podporuje Bluetooth tedy standard JSR-82.

## Rozhraní pro příjem Bluetooth událostí

Rozhraní `BTEventListener` poskytuje metody pro jednotný způsob zjišťování, že nastala nějaká Bluetooth událost. V případě, že nastala Bluetooth událost, jsou všichni registrovaní posluchači implementující toto rozhraní informováni vyvoláním příslušné metody rozhraní `BTEventListener`. Následuje seznam typů událostí definovaných tímto rozhraním:

- Přijetí zprávy od serveru.
- Úspěšné odeslání zprávy serveru.
- Objevení nového okolní zařízení.
- Diagnostická zpráva.
- Chybová zpráva.

## Bluetooth událost

Každá Bluetooth událost nese informaci o svém typu pro zpracování a textovou informaci s popisem pro uživatele. To znamená například, když nastane chyba, je generována Bluetooth událost typu chyba s popisem, o jakou chybu se jedná. Bluetooth událost je implementována ve třídě `BTEvent`.

## Vlákno pro hledání okolních zařízení

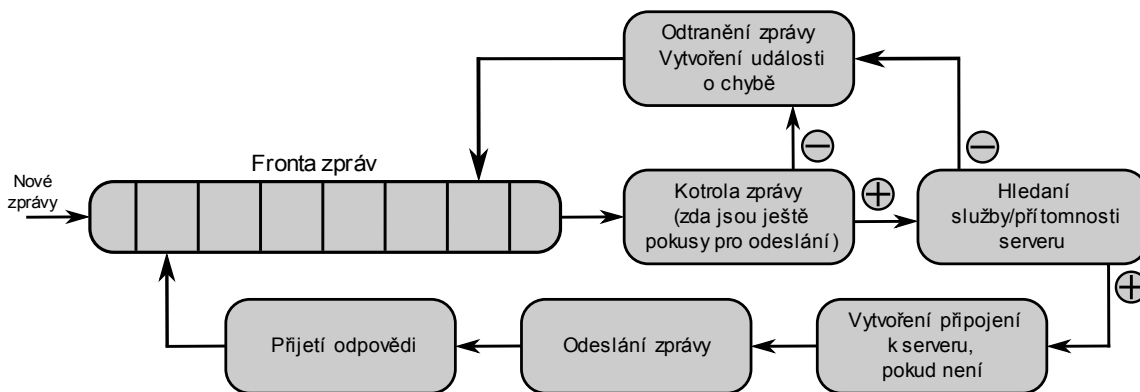
Jako každé vlákno i toto běží nezávisle na zbytku aplikace a jen kontroluje, zda nebyl vznesen požadavek vyhledat okolní zařízení. Pokud ano, spustí hledání. Informuje o každém nalezeném zařízení generováním Bluetooth události a nakonec o tom, že hledání skončilo. Seznam nalezených zařízení je možné získat pomocí metod manažeru.

## Vlákno pro komunikace se serverem

V tomto vláknu probíhá komunikace se serverem. Probíhá zde práce s frontou zpráv, což je delikátní záležitost, jelikož k jedné datové struktuře najednou přistupuje více vláken. Toto vlákno z ní čte zprávy a poté je odesílá, zatímco jiná vlákna zapisují do fronty zpráv nové zprávy k odeslání. Pro zajištění korektní práce s frontou zpráv existuje v Javě prostředek, který umožňuje synchronizaci a zamykání kritických kusů kódu a tím je blok s klíčovým slovem `synchronized`. Tím se zajistí zachování konzistence dat a nedojde k jejich poškození nebo ztrátě.

Ve frontě zpráv jsou tedy zprávy k odeslání, ty jsou postupně odesílány serveru. Vzhledem k tomu, že se jedná o bezdrátovou komunikaci, nelze spoléhat na to, že server je pořád schopen zprávy přijímat. Nejprve se zjišťuje, jestli je server přítomný v dosahu a schopný přijímat (to je zjištěno hledáním služby). Po nalezení služby klient ví, že může posílat data.

Nejvíce energie je spotřebováváno, když je navázáno spojení se serverem, proto se spojení udržuje jen na nezbytně dlouhou dobu a poté je vždy přerušeno. Před posláním další zprávy je nutné zjistit, zda je spojení momentálně otevřeno. Pokud ne, je automaticky znovu otevřeno. Teď už může být zpráva odeslána. S jejím odesláním je generována Bluetooth událost o úspěšném odeslání zprávy. V případě, že došlo při odeslání zprávy k chybě, pokusí se klient poslat zprávu znovu, to z toho důvodu, že jak už bylo řečeno, jde o bezdrátový přenos a může docházet k chybám. Pokusů pro odeslání je definovaný počet, informace o zbývajících pokusech je uložena u každé Bluetooth zprávy. Teprve až v případě, že se ani tak nepodařilo zprávu odeslat, je generována chyba a přechází se na další zprávu. Po odeslání zprávy se vyčkává na odpověď, při jejím přijetí je generována Bluetooth událost o přijetí zprávy. Celý mechanismus odesílání zpráv je znázorněn na obrázku 6.2.



Obrázek 6.2: Průběh odesílání zprávy

## Odpojení při nečinnosti

Během tohoto všeho ještě ale běží v dalším vlákne speciální počítadlo (třída `BTimeout`), které měří dobu připojení k serveru. Při každém odeslání zprávy je počítadlo vynulováno. V případě, že napočítá do nastaveného času, provede automatické zrušení spojení se serverem.

## Bluetooth zpráva

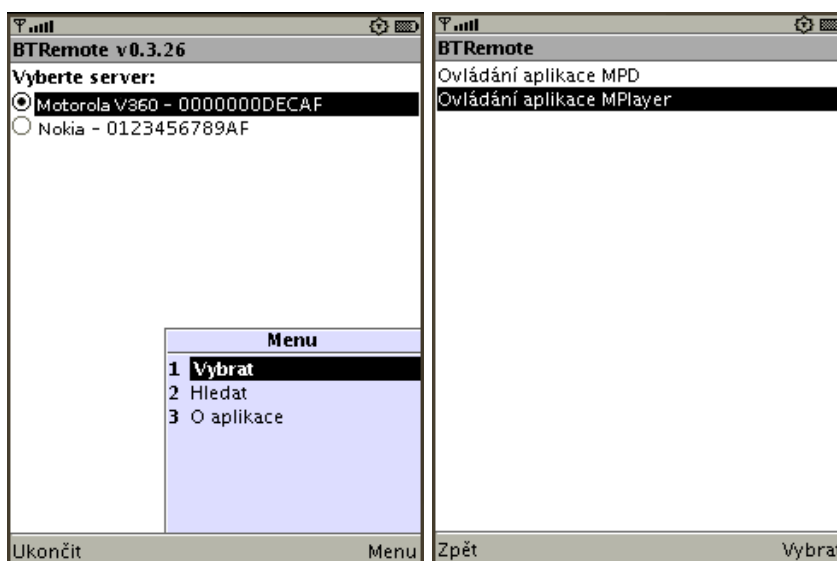
Do fronty zpráv se vkládají Bluetooth zprávy, třída `BTMessage`. Zpráva s sebou nese kromě samotné zprávy (data, která mají být přenesena) ještě informaci o tom, kterému serveru je zpráva určena a taky informaci o počtu zbývajících pokusů pro odeslání zprávy.

### 6.1.2 Klientská aplikace

Samotná aplikace pouze využívá funkcí jádra Bluetooth komunikace pro komunikaci ze serverem a vytváří uživatelské rozhraní, díky kterému je možné aplikaci používat. Uživatelské rozhraní se skládá z několika formulářů, následuje jejich přehled.

#### Úvodní formulář

Tento formulář slouží pro vyhledání a výběr serveru (obrázek 6.3 vlevo). Je to první formulář, který je zobrazen ihned po spuštění aplikace. Na tomto formuláři tedy lze ukončit aplikaci, přejít na formulář s informacemi o aplikaci, spustit hledání serveru a potom se k vybranému serveru připojit a tím přejít na formulář se seznamem ovládacích obrazovek. Při připojení k serveru je zároveň poslán serveru požadavek na získání seznamu ovládacích obrazovek (viz kapitola 5.3.1).



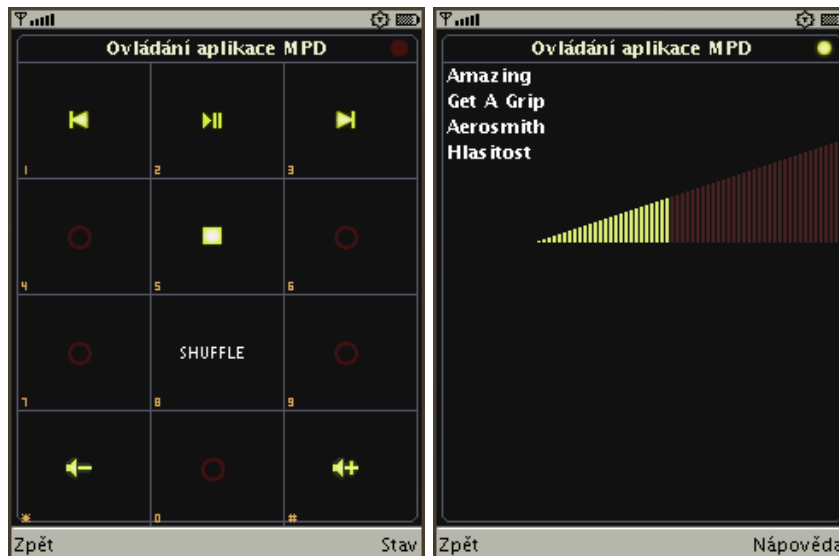
Obrázek 6.3: Ukázka formulářů

#### Seznam ovládacích obrazovek

Na tomto formuláři je zobrazen seznam ovládacích obrazovek získaných od serveru (obrázek 6.3 vpravo). Je možné přejít na vybranou ovládací obrazovku nebo se vrátit zpět na úvodní formulář a vybrat jiný server (eventuálně aplikaci ukončit). Při přechodu na ovládací obrazovku je od serveru vyžádán obsah ovládací obrazovky (viz kapitola 5.3.2).

## Ovládací obrazovka

Z informací od serveru je sestavena ovládací obrazovka, nejprve je nastaven titulek. Poté je zpracován kód přijatý od serveru. Podle těchto informací jsou definovány klávesy, kterými je možno odesílat příkazy. V případě požadavků na zpětné informace je ovládací obrazovka rozdělena na dvě části: zobrazení nápovědy k posílání příkazů pomocí kláves (obrázek 6.4 vlevo), to znamená zobrazení popisek nebo ikon k jednotlivým definovaným klávesám a na část zobrazení zpětných informací (obrázek 6.4 vpravo). Bez zpětných informací je zobrazena jen první část. Mezi oběma částmi se přepíná tlačítkem. Zpětné informace jsou zobrazeny pod sebou v části pro ně určené.



Obrázek 6.4: Ukázka uživatelského rozhraní ovládací obrazovky

### Přechody mezi formuláři

V některých případech je při přechodu mezi formuláři poslán požadavek serveru a podle odpovědi je upraven formulář. To může nějakou dobu trvat, z toho důvodu se v tomto čase zobrazí přechodný formulář, na kterém je zobrazen ukazatel průběhu. Tento přechodný formulář je implementován třídou `ProgressBarSwitch`.

Další akce, která trvá delší čas, je hledání okolních zařízení. V tuto dobu je zobrazen podobný přechodný formulář jako v předchozím případě, tento je ale implementován ve třídě `ProgressBarDiscoverer`.

### 6.1.3 Podpůrné třídy

K usnadnění implementace některých často opakovaných konstrukcí slouží podpůrné třídy v balíčku `support`.

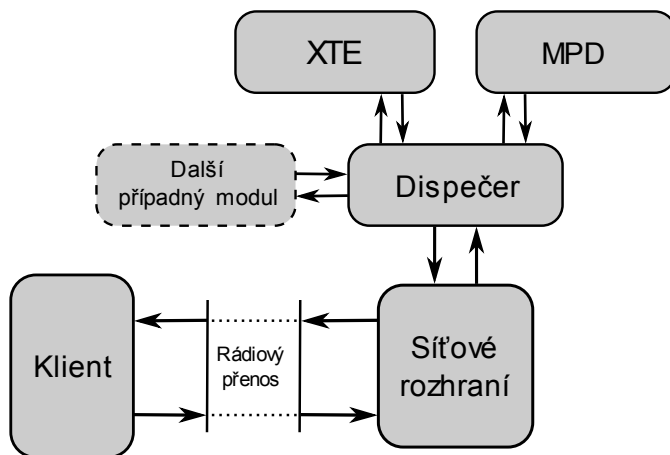
V J2ME například chybí implementace funkce pro rozdělení řetězce pomocí oddělovače. Vzhledem k častému použití při zpracování informací od serveru je tato funkce implementována ve třídě `SString`.

Další užitečnou třídou je třída `GraphicsItems`, ta má poskytovat funkce pro kreslení standardizovaných prvků na ovládací obrazovku. Prozatím je implementována jediná funkce a to `drawSlider` pro vykreslení ukazatele průběhu (obrázek 5.1).

Zpracované informace od serveru jsou ukládány do objektů, pro uložení zpětných informací je využívána třída `Item`. Třída `RefreshTime` implementuje počítání času pro opětovné vyžádání zpětných informací.

## 6.2 Server

Serverová část je implementována v jazyce C++. Je určena pro operační systém GNU/Linux. Využívá Bluetooth rozhraní BlueZ (viz kapitola 4.1). Slouží jako spojovací článek mezi počítačem a dálkovými ovladači v podobě mobilních telefonů, tzn. podle příkazů od klientů provádí požadované akce, k samotnému ovládní počítače jsou jednotlivé pluginy. Pluginy jsou zase spojovací články mezi serverem a konkrétní ovládanou aplikací na ovládaném počítači. Struktura serveru je znázorněna na obrázku 6.5.



Obrázek 6.5: Blokové schéma serveru

### 6.2.1 Spuštění/ukončení aplikace

Serverová část běží dlouhou dobu jen v pohotovostním stavu, kdy jen čeká na připojení a požadavky klientů, proto je vhodné ji spustit „na pozadí“. Pokud tedy chceme spustit server v tomto režimu, stačí server spustit s parametrem `-d`. To povede k systémovému volání `fork`, které vytvoří nový proces tak, že rozdvojí stávající, původní proces je ukončen a nový proces pak už běží jako přímý potomek procesu `init` (hlavní proces v operačním systému GNU/Linux), tím je docíleno běhu „na pozadí“.

Jelikož by nebyla žádná možnost, jak aplikaci ukončit, pokud běží „na pozadí“, je při jejím spuštění zaregistrována obsluha systémových signálů, konkrétně signálů `SIGTERM` a `SIGINT`. Po zaslání těchto signálů aplikaci je přerušena nekonečná smyčka, ve které dochází k obsluze klientů a aplikace je ukončena.

## 6.2.2 Bluetooth

Rozhraní pro práci s Bluetooth použité v serverové části je BlueZ (viz kapitola 4.1). Aby klienti mohli komunikovat se serverovou částí, musejí se k ní být schopni nejprve připojit. To je možné díky využití jednoho z Bluetooth protokolů – SDP (viz kapitola 2.2.5). Při inicializaci serverové části je zaregistrována Bluetooth služba s dohodnutými parametry, díky nimž je zmiňovaným protokolem služba dohledatelná pro klienty.

Pro samotnou síťovou komunikaci systém používá Bluetooth protokol RFCOMM (viz kapitola 2.2.4). Práce s tímto protokolem je téměř stejná jako s BSD sokety, tzn. komunikace probíhá v „rouře“ (socketu), jeden konec je v serveru a druhý ústí u klienta. Nejprve je vytvořen „konec roury“ v serveru, tzv. hlavní socket. Poté server začne poslouchat na tomto konci roury. K druhému konci se může připojit případný klient, z toho vyplývá, že pro každého klienta potřebujeme zvláštní rouru. Hlavní socket tedy slouží k připojování nových klientů, komunikace s každým klientem pak už probíhá vlastní vytvořenou rourou. Abychom se vyhnuli nutnosti použití implementace vícevláknové aplikace, využívá se speciální funkce `select`, která dokáže sledovat změny na všech socketech a v případě, že ke změně dojde, zavolat obslužné rutiny. V případě změny na hlavním socketu připojí nového klienta tím, že je pro něj vytvoří vlastní socket. V případě, že ke změně došlo na socketu určitého klienta, je zjištěno, o jakého klienta se jedná, a je přijat jeho požadavek. Tyto údaje jsou předány řídicímu objektu, tzv. dispečeru, ten je zpracuje a vrátí odpověď. Ta je posléze odeslána klientu.

## 6.2.3 Dispečer

Dispečer (třída `Dispatcher`) z údajů jemu předaných zpracuje požadavek a vygeneruje příslušnou odpověď. Základní příkazy dispečer zpracovává sám, ale v případě příkazů na ovládání nebo požadavku na zpětné informace musí volat příslušné moduly, tzn. dispečer zajišťuje i práci s moduly. Kromě toho ještě spravuje informace o ovládacích obrazovkách.

## 6.2.4 Konfigurační soubory

Ovládací obrazovky jsou definovány v konfiguračních souborech, ty je tedy nutno načíst a zpracovat (zajišťuje třída `Loader`). Načtená data jsou uložena v dispečeru, kde se s nimi dále pracuje.

## 6.2.5 Moduly

Moduly jsou objekty pro ovládání určité aplikace. Každý modul musí poskytovat možnost provést určitý příkaz a získat požadovanou informaci. Aby s nimi mohlo být pracováno obecně, je každý modul zděděn z třídy `Plugin`, ve které jsou deklarovány dvě virtuální metody, které to umožňují. Pro provedení příkazu je to metoda `process` a pro získání požadované zpětné informace metoda `get`. Každý konkrétní modul má tedy vlastní implementaci těchto metod.

## Xte

Modul `Xte` (třída `Xte`) slouží k simulování stisků kláves na klávesnici počítače. Využívá k tomu standardní rozšíření X serveru `XTest`. Ve svojí implementaci metody `process` tedy podle zadaného parametru „stiskne“ klávesu (možné parametry viz dodatek A.2). Metoda

`get` vrací vždy řetězec `FALSE` značící, že požadovanou informaci není možné získat. To proto, že tento modul neumožňuje získávat žádné zpětné informace.

## **Mpd**

Modul `Mpd` (třída `Mpd`) slouží k ovládní aplikace MPD (*Music Player Daemon*). Aplikace MPD je sama osobě serverová aplikace, její ovládní probíhá tak, že klient, v tomto případě modul `Mpd`, se připojí k aplikaci pomocí socketů. K tomu je využito knihovny `libmpdclient`, která je volně dostupná. Ta zajišťuje síťové připojení k aplikaci MPD. Modul `Mpd` pak jen na základě požadavků ve dvou metodách určených pro řízení zasílá požadavky aplikaci MPD a odpovědi zpracuje (seznam parametrů pro práci s modulem `Mpd` viz dodatek A.1).



# Kapitola 7

## Závěr

V této práci jsem se zabýval vytvořením systému pro dálkové ovládání počítače s operačním systémem GNU/Linux pomocí mobilního telefonu s podporou standardu JSR-82 (Bluetooth standard pro J2ME). Systém se zaměřuje na minimalizování spotřeby energie na mobilním telefonu tím, že Bluetooth spojení je udržováno jen po nezbytně nutnou dobu.

Při vývoji tohoto systému jsem se seznámil s mnoha moderními technologiemi jako jsou Bluetooth a J2ME. To mě přesvědčilo o tom, že je možné tyto technologie využít k mnoha různým účelům.

Systém je již v použitelném stavu, ale i tak bych ho považoval jen jako základ pro další možný vývoj. Mezi oblastí, které tato práce nepokrývá, patří například bezpečnost. V tomto ohledu by bylo možné implementovat systém právě tak, že by přístup k určitým ovládacím obrazovkám byl omezený. Možná důležitější by bylo implementovat systém autorizace, díky němuž by se předešlo jeho zneužití.

Z hlediska použitelnosti na jiných mobilních telefonech by jistě bylo třeba testovat klient-skou aplikaci na širokém spektru mobilních telefonů různých výrobců, jelikož každý mobilní telefon, i když by měl splňovat standard, má své odlišnosti. Během práce jsem měl přístup pouze k omezenému množství přístrojů. Také by mohl být vylepšen systém vykreslování informací na displeji mobilního telefonu (v případě textu širšího, než umožňuje zobrazit displej konkrétního zařízení, vytvořit například „běžící text“).

Systém je navržen tak, že umožňuje snadnou rozšiřitelnost implementací nových modulů, které slouží pro ovládání různých aplikací. Součástí této práce jsou dva moduly. Do budoucna by toto číslo mohlo značně stoupnout a tím vyhovět požadavkům každého uživatele. Například modul na spouštění jiných aplikací, díky němuž by se dal spustit i libovolný skript, vytvořený uživatelem. Rozšiřitelnost by šlo ještě usnadnit implementací načítání modulů jako dynamických knihoven, tím by odpadla nutnost rekompilovat aplikaci při přidání nového modulu.

# Literatura

- [1] Dee Dee M. Bakker and Diane McMichael Gilster. *Bluetooth End to End*. John Wiley & Sons, Inc., New York, NY, USA, 2002. ISBN 07-645-4887-5.
- [2] Qusay H. Mahmoud. *Naučte se Java 2 Micro Edition*. GRADA, 2002. ISBN 80-247-0444-7.
- [3] WWW stránky. Affix – Bluetooth rozhraní. <http://research.nokia.com/research/projects/affix/index.html>. [Online, přístupné 29.4.2008].
- [4] WWW stránky. Bluetooth. <http://en.wikipedia.org/wiki/Bluetooth>. [Online, přístupné 29.4.2008].
- [5] WWW stránky. Bluetooth profily. [http://en.wikipedia.org/wiki/Bluetooth\\_profile](http://en.wikipedia.org/wiki/Bluetooth_profile). [Online, přístupné 29.4.2008].
- [6] WWW stránky. Bluetooth protokoly. [http://en.wikipedia.org/wiki/Bluetooth\\_protocols](http://en.wikipedia.org/wiki/Bluetooth_protocols). [Online, přístupné 29.4.2008].
- [7] WWW stránky. BlueZ – Bluetooth rozhraní. <http://www.bluez.org>. [Online, přístupné 29.4.2008].
- [8] WWW stránky. Connected Device Configuration (JSR 36). <http://www.jcp.org/en/jsr/detail?id=36>. [Online, přístupné 29.4.2008].
- [9] WWW stránky. Connected Limited Device Configuration (JSR 30). <http://www.jcp.org/en/jsr/detail?id=30>. [Online, přístupné 29.4.2008].
- [10] WWW stránky. Java API pro Bluetooth. <http://www.jcp.org/en/jsr/detail?id=82>. [Online, přístupné 29.4.2008].
- [11] WWW stránky. Remote Method Invocation (JSR 66). <http://www.jcp.org/en/jsr/detail?id=66>. [Online, přístupné 29.4.2008].

# Dodatek A

## Specifikace funkcí modulů

Každý z modulů má své specifické funkce, proto zde následuje seznam funkcí pro získání zpětné informace i pro řídicí příkazy pro oba moduly již implementované v systému. Použití řídicí funkce v konfiguračním souboru je následující:

`key:modul:klávesa na mobilním zařízení:řídicí funkce:identifikující text`

### A.1 Modul MPD

Modul MPD slouží k ovládání aplikace Music Player Daemon (MPD).

#### A.1.1 Řídicí funkce

- `toggle` – pokud je přehrávání hudby zapnuto, provede pozastavení, jinak spuštění přehrávání.
- `pause` – pozastaví přehrávání hudby.
- `play` – spustí přehrávání hudby.
- `stop` – zastaví přehrávání hudby.
- `next` – přepne přehrávání na další písničku.
- `prev` – přepne přehrávání na předchozí písničku.
- `incVolume` – zvýší hlasitost.
- `decVolume` – sníží hlasitost.
- `random` – pokud je zapnuto přehrávání v náhodném pořadí, tak ho vypne, jinak zapne.

#### A.1.2 Funkce pro získání zpětné informace

- `song` – získá název aktuálně přehrávané písničky.
- `artist` – získá autora aktuálně přehrávané písničky.
- `album` – získá album aktuálně přehrávané písničky.

- `state` – získá aktuální stav přehrávání.
- `israndom` – získá informaci o tom, zda je zapnuto náhodné přehrávání.
- `volume` – získá aktuálně nastavenou hlasitost v procentech.
- `progress` – získá stav aktuálně přehrávané písničky v procentech.

## A.2 Modul XTE

Modul XTE slouží k simulování stisku kláves na klávesnici počítače.

### A.2.1 Řídící funkce

Jako řídicí funkce může být v tomto modulu použita většina znaků, které je možné napsat na klávesnici, tzn. například.

a b c A B C . ? ! 0 5 9 ( ) { } [ ]

Dále řídicí klávesy jako například.

space Return Escape Backspace  
 Left Right Up Down  
 F5 Tab  
 Delete Page\_Up Page\_Down Home End  
 XF86AudioPlay XF86AudioPause

### A.2.2 Funkce pro získání zpětné informace

Tento modul žádné takové funkce neposkytuje.

## Dodatek B

# Ikony podporované klientem

Ikony definované na klientu pro zobrazení jako nápověda místo textu. První sadou ikon jsou ikony pro ovládání přehrávače:



Další jsou ikony navigačních šipek:



Ikony týkající se hlasitosti:



A nakonec další ikony nezařazené do žádné skupiny:



# Dodatek C

## Obsah CD

Obsah přiloženého CD je následující.

- soubor `bp.pdf` – technická zpráva ve formátu PDF.
- adresář `paper` – zdrojové soubory technické zprávy (`LATEX`).
- adresář `client` – zdrojové texty klientské části v J2ME.
- adresář `server` – zdrojové texty serverové části v C++.
- adresář `doc/client` – programová dokumentace klientské části vygenerovaná pomocí aplikace Doxygen. Titulní strana je v souboru `index.html`.
- adresář `doc/server` – programová dokumentace serverové části vygenerovaná pomocí aplikace Doxygen. Titulní strana je v souboru `index.html`.