

Gymnázium, Brno, tř. Kpt. Jaroše 14

2005/2006

4.C

Tomáš Janoušek

závěrečná maturitní práce

Tomáš Janoušek

xchat-bttrw

příručka programátora

Prohlašuji, že jsem předloženou práci zpracoval samostatně a použil jen uvedené prameny a literaturu.

Vedoucí projektu: Mgr. Marek Blaha

V Brně dne 19. ledna 2006

Tomáš Janoušek

Obsah

1	Úvod	1
2	Trocha historie	1
3	Vývoj a správa verzí	2
4	Komunikační protokoly	2
4.1	IRC protokol	2
4.2	HTTP protokol	3
5	Komunikace s portálem xchat.cz	3
5.1	HTTP požadavky	3
5.2	Funkce pro přímé požadavky – místnost	4
5.3	Parser zpráv	4
5.4	Příchozí fronta	5
5.5	Odchozí fronta	5
5.6	Funkce pro přímé požadavky – obecné	6
6	Události	6
7	Převod znakových sad	6
8	Idler – udržovač	7
9	Smajlíky	7
10	IRC brána	7

1 Úvod

xchat-bttrw (celým názvem xchat-back to the real world) je brána pro přístup na chatový portál xchat.cz přes protokol IRC. Jejím srdcem je knihovna libxchat-bttrw, která realizuje veškerou komunikaci s portálem a kterou je možno použít nezávisle na bráně. Projekt je napsán v jazyce C++ a je již od počátku uvolněn pod licenci GNU GPL v2. Webová stránka projektu sídlí na adrese <http://nomi.cz/projects.shtml?id=xchat-bttrw>.

Tato práce má za cíl vytvořit srozumitelnou a kompletní programátorskou dokumentaci k projektu. Je rozdělena na dvě části, programátorskou příručku a referenční dokumentaci zdrojových kódů. Příručka programátora je psaná česky a je určena programátorům, kteří chtějí hlouběji pochopit principy fungování brány a především knihovny a případně do projektu přispívat. Naopak referenční dokumentace je generovaná automaticky z dokumentačních komentářů, podrobně popisuje jednotlivé třídy, funkce a proměnné, jejich použití a vztahy. Je tedy nezbytnou pomůckou při psaní nového kódu a úpravách původního. Její význam tkví také v tom, že veškerý její obsah pochází přímo ze zdrojových kódů, takže při jeho úpravách je člověk nucen upravovat i dokumentaci.

Uživatelská příručka není ani cílem, ani součástí této práce. Ta je umístěna na naší dokumentační wiki na adrese <http://wiki.nomi.cz/xchat-bttrw:start>. Na jejím vytvoření a doplňování se může podílet kdokoliv, protože wiki je otevřená a každý může přispívat.

2 Trocha historie

Projekt vznikl 18. dubna 2004, avšak vývoj zase utichl a na funkční podobě se začalo pracovat až 30. července 2004. V průběhu vývoje došlo k několika významným vylepšením, proto uvedu jejich výčet:

- **11. září 2004** – první velké pročištění kódu – brána byla pevně oddělena od knihovny
- **12. září 2004** – podpora zobrazování (nejen) českých znaků a jejich převodu do jiných znakových sad
- **19. září 2004** – kompatibilita se systémem Windows XP – to se týkalo především knihovny TomiTCP
- **16. ledna 2005** – výrazné vylepšení rozkládání zátěže na jednotlivé servery xchat.cz clusteru, díky kterému je brána mnohem tolerantnější vůči chybám xchatu
- **14. února 2005** – vydání verze 0.1
- **22. srpna 2005** – kompatibilita se systémy Windows 98, NT4 a 2000

3 Vývoj a správa verzí

xchat-bttrw je již od začátku jedním z projektů NOMI teamu¹. Trvale se na něm podílí několik lidí:

- Tomáš Janoušek – maintainer, programátor
- Ondřej Nový – programátor, vydavatel balíčků s instalátorem pro Windows
- Jan Kaláb – grafik

Kromě nás také pomáhá celá řada lidí, kteří hlásí chyby, pomáhají s testováním v různých prostředích, přispívají svými záplatami či nápady.

Ke správě verzí se používá Subversion². Projekt sídlí v mém osobním svn repozitáři – <http://svn.nomi.cz/svn/tomi/xchat-bttrw/>.

Projekt se stále vyvíjí, aktuální cíle vývoje jsou k nahlédnutí na adrese <http://wiki.nomi.cz/xchat-bttrw:roadmap>. Také probíhá diskuze nad novými návrhy v sekci Koncepty na adrese <http://wiki.nomi.cz/xchat-bttrw:koncepty>.

4 Komunikační protokoly

V programu se používá několik komunikačních protokolů. Veškerá síťová komunikace je realizována na vrstvě TCP/IP s použitím několika aplikačních protokolů, které dále popíšeme.

Komunikaci protokolem TCP zajišťuje knihovna TomiTCP, jíž jsem autorem. Je taktéž uvolněna pod licencí GNU GPLv2 a je dostupná na adrese <http://svn.nomi.cz/svn/TomiTCP/>.

4.1 IRC protokol

IRC – Internet Relay Chat – protokol vznikl v srpnu 1988 na univerzitě v Oulu ve Finsku. Od té doby se vyvíjel, především jeho část určená pro komunikaci serverů mezi sebou, a i dnes se stále hojně používá. Je to protokol určený pro internetové chatové servery, ba dokonce sítě – servery jsou mezi sebou propojené a uživatel se vůbec nemusí zajímat, ke kterému se připojí. Jeho síla tkví v jeho jednoduchosti a právě to bude zřejmě důvodem jeho „nesmrtelnosti“. Důkladně je popsán v několika RFC³ – [RFC1459], [RFC2812] a pár dalších, které pro nás nejsou podstatné. Pro IRC protokol existuje řada klientů, jako např. xchat (shoda jmen je opravdu čistě náhodná) či irssi, jejichž použití je mnohem příjemnější a praktičtější než webové rozhraní portálu xchat.cz.

¹<http://nomi.cz/>

²<http://subversion.tigris.org/>

³„RFC je zkratka anglického výrazu Request For Comments (žádost o komentáře), která se používá pro označení řady standardů a dalších dokumentů popisujících Internetové protokoly, systémy apod. Jak už název napovídá, RFC jsou oficiálně považovány spíše za doporučení než normy v tradičním smyslu, přesto se podle nich řídí drtivá většina Internetu.“ – [Wikipedia, <http://cs.wikipedia.org/wiki/RFC>]

Pro implementaci IRC protokolu stačily dvě funkce – *parsein* a *parsemode* – a sada IRC příkazů a odpovědí přímo v kódu brány (jelikož právě IRC komunikace je její hlavní náplní).

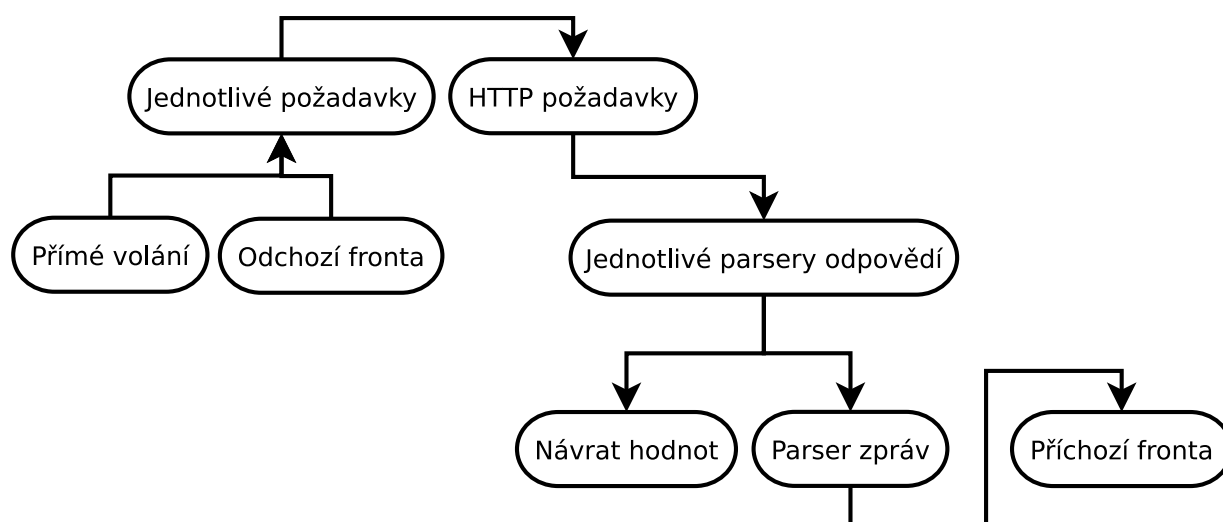
4.2 HTTP protokol

Protokol HTTP je základní webový protokol. Popsán je, mimo jiné, v [RFC2616]. Zde figuruje jako prostředek pro komunikaci s portálem.

Přestože jsou dostupné různé jiné knihovny s podporou protokolu HTTP, zde se používá jednoduchá implementace HTTP/1.0 v knihovně TomiTCP.

5 Komunikace s portálem xchat.cz

Komunikaci můžeme rozdělit na několik vrstev, které na sebe navazují podle následujícího schématu:



5.1 HTTP požadavky

Nejnižší úroveň, která nás zajímá, jsou samotné HTTP požadavky. Zde si popíšeme, jak vypadají, jak se řeší vyrovnávání zátěže, vyřazování nefunkčních serverů atd.

Pro HTTP komunikaci nám knihovna TomiTCP nabízí třídu *TomiHTTP*, která definuje následující typy, proměnné a funkce:

```

int GET(string address, TomiCookies* cookies);
int POST(string address, const string& data, TomiCookies* cookies);
typedef map<string,string> headers_t;
headers_t headers;

```

```
static string URLEncode(const string& s);  
static string URLDecode(const string& s);
```

Po zavolání požadavku je možno číst odpověď po řádcích funkcí *getline* zděděnou ze třídy *TomTCP*. HTTP hlavičky jsou v případě potřeby k dispozici v proměnné *headers*.

Xchat.cz má několik serverů, mezi které rozděluje zátěž. Občas nejsou některé z nich dostupné a je nutno se jejich použití dočasně vyvarovat. Jejich seznam se získává z DNS funkcí *makeservers* před přihlášením. Každý server je reprezentován třídou *server*, kde se uchovávají kromě jeho adresy také údaje o jeho funkčnosti. Pro výběr vhodného serveru pak slouží funkce *makesrv*, která podle informací o spolehlivosti vybere jeden z funkčních serverů. Při hledání serveru také maže staré informace o serverech, aby po nějakém čase nefunkčnosti dostaly další šanci.

Adresy pro HTTP požadavky vytváříme pomocí funkcí *makeurl* a *makeurl2*, které vyberou a dosadí vhodný server a funkce *makeurl2* navíc přidá údaje o sezení, kterými se identifikujeme po přihlášení.

V případě, že nějaký server v průběhu požadavku selže, voláme funkci *lastsrv_broke*. Ta aktualizuje informace o spolehlivosti serveru a případně označí daný server za nefunkční. Poté se pokusíme tentýž požadavek zopakovat ještě několikrát na jiné servery – to však neplatí pro vyřizování požadavků z front a získávání nových zpráv, tam se další pokus provede sám v příštím průchodu.

5.2 Funkce pro přímé požadavky – místnost

Místnost je reprezentována strukturou *room*, která udržuje všechny potřebné informace o místnosti. To zahrnuje seznam přítomných nicků, správce a další. Pro komunikaci s místností má knihovna *xchat-bttrw* následující funkce:

Pro vstup do místnosti slouží funkce *join*, která kromě samotného vstupu do místnosti ještě získá všechny potřebné informace a vyplní je do struktury *room*. Opuštění místnosti provádí funkce *leave*, která zároveň smaže záznam z pole místností.

Funkce *getroominfo* získává název a popis místnosti a seznam stálých správců. Volá se ve funkci *join*, ale také ve funkci *fill_recvq*, aby se zjistily případné změny popisu anebo stálých správců. Chceme-li naopak popis nastavit, voláme funkci *setdesc*.

K odesílání zpráv slouží funkce *putmsg*. K jejich příjmu pak *getmsg* – ta je nejsložitější ze všech, neboť musí zpracovat JavaScriptový kód z xchatu. Jednotlivé kusy pak přímo předává ke zpracování parseru. Kromě získávání zpráv také rozpoznává případné přesměrování na stránku o vyhození z místnosti.

5.3 Parser zpráv

Vstupním rozhraním parseru je funkce *recvq_parse_push*. Ta se postará o správné zpracování a rozeznání zprávy a její následné zařazení do příchozí fronty jako událost.

Parser pro vnitřní účely definuje řadu funkcí, které upravují vstupní řetězec a detekují jednotlivé typy zpráv:

Nejprve se volá funkce *isadvert*, která zjistí, zda-li se jedná o reklamní zprávu a uloží její případný odkaz. Pak *stripjsescapes*, *striphtml* a *striphtmlent* převedou vstup do čistě textové podoby. Poté se pomocí funkcí *getdate* a *getnick* získají hlavičky zprávy. Nyní je už zpráva připravena a může se rozpoznávat. Pokud z hlaviček vyplývá, že jde o normální zprávu, převedou se smajlíky pomocí funkce *unsmilize* a podle daných pravidel se emituje událost. Pokud se jedná o zprávu systémovou, je třeba ji dále rozpoznávat. O to se starají funkce *isjoin*, *isleave* a *iskick*. Po rozpoznání se emituje odpovídající událost.

Parser nikdy neselže, v případě nerozpoznatelné zprávy emituje systémovou událost. Toho se dokonce v mnoha případech záměrně využívá, aby se uživateli zobrazily systémové zprávy, které brána nepotřebuje zpracovávat.

5.4 Příchozí fronta

Příchozí fronta je dynamické pole objektů typu *recv_item*, které zapouzdřují automatický ukazatel na typ *Event*. Do této fronty se vkládají veškeré události, včetně chybových hlášek, které není žádoucí vyhodit jako výjimku. Přidávání do příchozí fronty tedy není jen záležitostí parseru zpráv.

Pro vložení události do fronty slouží funkce *recvq_push*. Ukazatel, předávaný této funkci, není nutno a ani dovoleno uvolňovat, neboť automatický ukazatel v příchozí frontě přebírá jeho vlastnictví.

K výběru z fronty je k dispozici funkce *recvq_pop*. Vrací ukazatel, o který se musíme potom postarat. Je tedy vhodné ho vložit do vlastního automatického ukazatele.

5.5 Odchozí fronta

Odchozí fronta je dynamické pole objektů typu *send_item*. Je potřebná z několika důvodů. Hlavní příčinou je, že xchat má jistá omezení proti záplavám zpráv. Mezi jednotlivými zprávami je tedy potřeba dělat několikasekundové prodlevy. Čekání v hlavní smyčce by bylo samozřejmě nepřijatelné, takže se řadí do fronty. Z toho pak dále plyne požadavek na priority zpráv. Příkazy pro správce místnosti, jako je např. vykopnutí, se vkládají na začátek fronty, aby mohlo dojít k jejich okamžitému odbavení. Dalším omezením ze strany xchatu je maximální délka zprávy, která činí přibližně 200 znaků. Je tedy potřeba zprávu před odesláním rozdělit a zbylé části doručit později.

O vyřizování odchozí fronty se stará funkce *do_sendq*. Ta vždy jednou za čas vezme zprávu ze začátku fronty. Není-li u zprávy určena místnost, pokusí se příjemce najít, a pokud neuspěje, pošle zprávu jako globální příkazem „/m“. Potom v případě, že je zpráva moc dlouhá, oddělí první kus o dané maximální délce. Teprve poté se pokusí o odeslání. Když odeslání selže kvůli technické chybě, vyhodí se výjimka a zpráva setrvá ve frontě. Pokud xchat zprávu odmítne přijmout, sníží se její počet pokusů a po několika neúspěších se ztratí úplně.

Pro jednoduché vkládání zpráv definuje třída XChat tyto funkce: *msg*, *whisper*, *kick*, *kill*, *admin*, *lock* a *unlock*. Jejich detailní popis je k dispozici v referenční dokumentaci.

5.6 Funkce pro přímé požadavky – obecné

Kromě chatování v místnosti potřebujeme ještě přístup k několika obecným informacím, jako například seznam místností, detailní informace o uživateli apod.

Pro některé z těchto informací dává xchat k dispozici jednoúčelové skripty, které méně zatěžují server a jsou jednodušší na zpracování. Popis k nim je dostupný na [xchat - skripty]. Tyto skripty se snažíme používat všude, kde je to možné.

K získání seznamu místností slouží funkce *list*. Ta natáhne a zpracuje úvodní stránku xchatu a přečte z ní seznam stálých místností. Do pole předaného parametrem pak uloží seříděný seznam místností a jejich identifikačních čísel.

Jednoduché a rychlé zjištění, jestli je uživatel právě online, umožňuje funkce *ison*. Pokud potřebujeme více informací, jako jsou kupříkladu jeho jméno, věk, množství nachatovaného času apod., použijeme funkci *userinfo*. Obě tyto funkce získávají data z výše uvedených jednoúčelových skriptů.

6 Události

Události jsou reprezentovány třídou *Event* a třídami od ní odvozenými. Třída *Event* definuje virtuální funkce *str* a *date* a odpovídající datové položky *s* a *d*. Od ní je pak odvozeno pár tříd pro chyby, šeptání, systémové zprávy a především události v místnosti – třída *EvRoomOther*. Od té se pak odvozuje několik tříd pro různé konkrétní události.

Třída *Event* ani žádná z jejích potomků nedefinuje datovou položku, kterou by ji bylo možno identifikovat, jak bývá zvykem. Namísto toho se pro jejich identifikaci používá jedna z vlastností RTTI¹ – dynamické přetypování. Test, zda-li ukazatel na *Event*, pojmenovaný *e*, je objektem třídy *EvRoomMsg*, se provádí takto:

```
if (dynamic_cast<EvRoomMsg*>(e))  
    příkaz;
```

7 Převod znakových sad

Převod znakových sad je nutný dvou místech. Jedním je převod HTML entit na UTF-8, druhým je pak převod na rozhraní celého libxchat-bttrw podle aktuálně nastaveného kódování v proměnné *client_charset*.

O převod HTML entit se stará funkce *striphtmlent*. Činí tak podle převodní mapy z HTML entit na UCS-2, kterou inicializuje funkce *init_recode*.

Pro převod na rozhraní knihovny definuje třída XChat dvě funkce – *recode_to_client*, pro převod směrem ven z knihovny, a *recode_from_client*, pro převod směrem dovnitř. Obě

¹Identifikace typu za běhu, detailněji popsána např. v [MvC++, str. 722]

používají funkci *recode* z knihovny TomiTCP, která používá *iconv*. Na systémech, které nepoužívají *glibc*, je tak brána závislá na přítomnosti knihovny *libiconv*.

8 Idler – udržovač

Jelikož *xchat.cz* automaticky vyhazuje uživatele z místnosti po určité stanovené době nečinnosti, bylo pro zachování pohodlí IRC nutné implementovat udržovač, který čas od času pošle nějakou zprávu a tím vyhození předejde.

Systém *xchat.cz* umožňuje předejít vyhození několika způsoby. Jedním z nich je šeptání sám sobě, což má tu výhodu, že se dá poslat rozpoznatelný řetězec, a ten pak odfiltrovat. Náš udržovač posílá 30 znaků dlouhý řetězec složený ze dvou polovin, které mají mezi sebou přesně daný vztah. Vytváří jej funkce *genidle* a funkce *checkidle* rozpoznává, zda-li zadaný řetězec vyhovuje těmto pravidlům. To umožňuje pro udržovač emitovat zvláštní druh události.

Zprávy udržovače se posílají ve funkci *do_sendq*, pokud je odchozí fronta prázdná a od poslední odeslané zprávy v místnosti uplynula zadané doba.

9 Smajlíky

Na *xchatu* je uživatelům k dispozici široké spektrum různých smajlíků a jiných emotikonů. Ty jsou označeny čísly a zobrazují se jako malé obrázky. Jelikož je protokol IRC čistě textový, umožňuje brána jejich převod do textové podoby, aby byly čitelné pro každého.

Knihovna podporuje jak převod smajlíků v číselné, tak i v grafické podobě. Detekce smajlíků v grafické podobě probíhá ve funkci *striphtml* a převádí se tam do podoby textové.

Používá se k tomu převodní mapa definovaná v souboru *smiles.c*. Konkrétní překlad smajlíků se dělá ručně a podílelo se na něm několik lidí. Na překladu nově přibývajících smajlíků se také pracuje a je rozpracován koncept částečně automatizovaného systému usnadňujícího překlad nových smajlíků.

10 IRC brána

IRC brána simuluje IRC server a umožňuje tak přístup na *xchat.cz* pomocí IRC klienta. Po spuštění začne poslouchat na daném TCP portu a čeká na příchozí spojení. Při příchodu spojení se oddělí nový proces (pozn. to neplatí pro platformu Windows, protože na ní chybí funkce *fork* a podpora vláken zatím není hotová) a ten klienta převezme. Po úspěšné autentifikaci příkazy NICK, PASS a USER se pokusí o přihlášení (vytvoření instance třídy XChat).

Dále pak čeká na příkazy od klienta a v pravidelných intervalech vyřizuje odchozí frontu a získává nové zprávy.

Použitá literatura

- [RFC1459] Oikarinen, J., Reed, D.: RFC 1459 – Internet Relay Chat Protocol, 1993
<http://www.faqs.org/rfcs/rfc1459.html>
- [RFC2812] Kalt, C.: RFC 2812 – Internet Relay Chat: Client Protocol, 2000
<http://www.faqs.org/rfcs/rfc2812.html>
- [RFC2616] RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1
<http://www.faqs.org/rfcs/rfc2616.html>
- [Wikipedia] <http://www.wikipedia.org/>, otevřená internetová encyklopedie
- [xchat - skripty] <http://xchat.centrum.cz/~guest~/extra/?tab=3>
- [MvC++] Prata, S.: Mistrovství v C++, 2. vydání, Computer Press, 2004

Resumé

xchat-bttrw je brána pro přístup na chatový portál xchat.cz přes protokol IRC. Její součástí je samostatně použitelná knihovna libxchat-bttrw, která realizuje komunikaci s portálem. Projekt je psán v jazyce C++ a je uvolněn pod licencí GNU GPL v2.

Cílem této práce bylo vytvořit srozumitelnou programátorskou dokumentaci. Ta se sestává ze dvou částí, anglicky psané referenční dokumentace a česky psané programátorské příručky. Referenční dokumentace je generovaná automaticky z dokumentačních komentářů ve zdrojových kódech programem doxygen.

Při vytváření této práce jsem se naučil psát dokumentační komentáře pro program doxygen, zvykl jsem si dokumentovat kód a také jsem ho v průběhu zkoumání analyzoval na případné nedostatky.

Předpokládám, že tato práce najde další užitek u programátorů, kteří budou chtít přispět do projektu. Pomůže jim pochopit jak brána funguje a referenční příručka jim jasně ukáže, co která funkce dělá a jak se používá.

xchat-bttrw is a gateway allowing to access the xchat.cz chat portal using the IRC protocol. Its part is the libxchat-bttrw library, which can be used separately and which does the communication with the portal. The project is written in the C++ programming language and is released under the GNU GPL v2 license.

The goal of this paper was creating an understandable programmer documentation. It consists of two parts, the reference documentation in English and the programmer manual in Czech. The reference documentation was autogenerated from documentation comments in the source code using the doxygen program.

During the creation of this paper I learnt how to write documentation comments, I got used to document the code and I also analyzed it for possible flaws.

I suppose that this paper will be useful to programmers willing to contribute to the project. It will help them understand how the gateway works and the reference documentation will show them what do the functions do and how to use them.