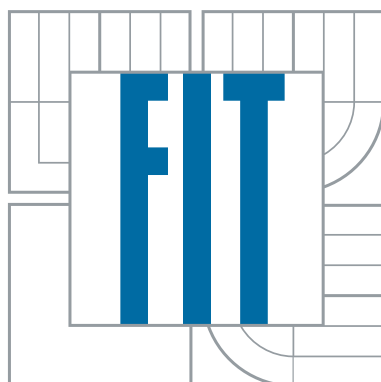


FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



Dokumentace k projektu předmětu IFJ – XMPP/Jabber klient

Obsah

1	Úvod	3
2	XMPP/Jabber protokol	3
3	Popis implementace	3
3.1	Zpracování parametrů	3
3.2	Připojení k serveru	3
3.3	Zpracování XML zpráv	3
3.4	Komunikace s XMPP serverem	4
3.5	Běh programu a interaktivní mód	4
4	Zajímavější pasáže implementace	4
5	Použité knihovny jazyka Python	5
6	Návod k použití	5
7	Závěr	6
8	Zdroje	6

1 Úvod

Cílem zadaného projektu bylo navrhnout a vytvořit klienta, umožňujícího komunikaci prostřednictvím XMPP protokolu. Klient by měl mít schopnosti přijímat a odesílat zprávy, přidávat nové a zobrazovat stávající kontakty v seznamu kontaktů, registrovat nového uživatele v síti a také pracovat v interaktivním režimu.

2 XMPP/Jabber protokol

Při použití XMPP protokolu je komunikace zapouzdřována do XML zpráv, podrobně popsanych v několika RFC, které jsou dále zpracovávány serverem či klientem. Pro možnost komunikace je potřeba, kromě klienta schopného zpracovat zprávy protokolu, mít zaregistrovaný účet na serveru, který se nazývá Jabber ID (často používaná zkratka je JID) a má tvar jako běžné emailové adresy. Standartně servery XMPP komunikují na portu 5222.

3 Popis implementace

Pro implementaci klienta byl použit skriptovací jazyk Python pro jeho jednoduchost a dostupnost knihoven potřebných pro vypracování zadaného projektu.

3.1 Zpracování parametrů

Počáteční fází implementace bylo zpracování parametrů příkazového řádku. K tomuto účelu byla použita standardní knihovna jazyka Python – `argparse`. Díky použití knihovny stačí nastavit ve skriptu názvy přepínačů a typ ukládané hodnoty. Knihovna také zajišťuje částečné ověření parametrů. Veškeré zpracování parametrů příkazového řádku probíhá ve třídě `Params`.

3.2 Připojení k serveru

Jednou z nejpodstatnějších částí projektu je komunikace se serverem, bez níž by nebylo možné v implementaci pokračovat. Komunikace klienta se serverem je vyřešena ve třídě `Connection` a byla pro ni použita knihovna `socket`. Pomocí metod knihovny se klient připojí na zadaný server, odesílá zprávy od klienta a přijímá zprávy přicházející od serveru.

3.3 Zpracování XML zpráv

Pro umožnění komunikace mezi klientem a serverem bylo potřeba zajistit zpracování přijímaných zpráv. Komunikace XMPP protokolu je implementována jako stream. Inicializace XMPP streamu začíná uvozující značkou `<stream:stream>`, a ukončující značka `</stream:stream>` je zaslána až při zakončení komunikace s XMPP serverem (viz [RFC 3920](#)). Použitím běžného parsování by došlo k ukončení chybou ve chvíli, kdy by parser nenalezl ukončující značku. Z toho důvodu bylo pro zpracování XML zpráv potřeba použít iterativní parsování. Parsování a vytváření XML zpráv je implementováno ve třídě `Communication`.

3.4 Komunikace s XMPP serverem

Vytváření a zpracování obsahu pro komunikaci s XMPP serverem se odehrává ve třídě `Xmpp`. První metoda této třídy, která se spustí při komunikaci s XMPP serverem je `init()`. Slouží k inicializaci XML streamu a načtení možností pro přihlášení se k účtu. Pro přihlášení slouží metoda `login()`. Ta prostřednictvím jednoho z přihlašovacích mechanismů ověří přihlašovací údaje uživatele, a buďto uživatele přihlásí, nebo aplikaci ukončí. Dále jsou implementovány metody pro načtení a vypsání seznamu kontaktů (`loadContacts()`, `showContacts()`), přidání uživatele do seznamu kontaktů (`addContact()`), registraci nového uživatele (`register()`), přijetí chat zprávy od serveru (`recvMsg()`) a pro odhlášení a ukončení streamu (`logout()`, `close()`)

3.5 Běh programu a interaktivní mód

Volání jednotlivých metod pro komunikaci s XMPP probíhá ve třídě `Application` v pořadí, jež je dáno zadáním, a lze je také kombinovat pomocí parametrů příkazového řádku. Pokud je program spuštěn v interaktivním módu, jsou tyto metody volány podle příkazu uživatele. V interaktivním módu má uživatel k dispozici příkazy:

login přihlášení, pokud byl interaktivní mód spuštěn bez poskytnutí přihlašovacích údajů

status vypsání zda je uživatel přihlášený nebo ne

whoami zobrazení JID uživatele

names zobrazení seznamu kontaktů uživatele

send odeslání chat zprávy jinému uživateli

exit ukončení interaktivního módu

help vypsání nápovědy

4 Zajímavější pasáže implementace

Zajímavá část implementace je u interaktivního režimu. Jelikož zde mohou zprávy přicházet kdykoliv, je potřeba zajistit neustálé načítání ze serveru. To je řešeno pomocí vytvoření dalšího vlákna, které má za úkol pouze načítat příchozí zprávy.

Protože však je zároveň potřeba mít možnost načíst zpětnou vazbu od serveru po odeslání požadavku, je nutno zajistit, aby tuto zpětnou vazbu nepřevzala právě metoda běžící ve druhém vlákne. Řešení je implementováno tak, že se odpovědi dočasně ukládají do seznamu spolu s transakčním identifikátorem který je u zpětné vazby serveru stejný, jako u odeslaného požadavku. Podle identifikátoru se poté vrátí odpověď tomu vláknu, kterému patří a odstraní se ze seznamu.

5 Použité knihovny jazyka Python

Při implementaci projektu byly použity tyto standardní knihovny jazyka Python:

argparse načtení parametrů příkazové řádky

sys kvůli zápisu na `STDERR` při chybě

logging k logování chyb a příchozích a odchozích zpráv

re ke kontrole zda zadané JID odpovídají správnému formátu

socket k zajištění spojení se serverem

errno pro odchycení `EINTR` – Interrupted system call

xml.etree.ElementTree pro parsování a vytváření XML zpráv

base64 pro zakódování přihlašovacích údajů

time pro metodu `sleep()` k uspaní programu na zadaný čas

threading pro použití vláken

6 Návod k použití

Pro spuštění je potřeba mít nainstalovaný interpret jazyka Python ve verzi 3.4. Práva na spouštění obstará Makefile přiložený ke skriptu.

Ukázka spuštění a přihlášení prostřednictvím interaktivního režimu:

```
./xmppclient -i
>>> help
login      [server (port) username password]
status     []
whoami     []
names      []
send       [receiver message]
exit       []
help       []
>>> login netfox.fit.vutbr.cz user password
>>> whoami
user@netfox.fit.vutbr.cz
>>> status
Logged in
>>> exit
```

Ukázka spuštění a zobrazení kontaktů přihlášeného uživatele:

```
./xmppclient -s netfox.fit.vutbr.cz -l user:password -c
- None <test@netfox.fit.vutbr.cz>
- New Buddy <anyone@netfox.fit.vutbr.cz>
- None <zkouska@netfox.fit.vutbr.cz>
```

7 Závěr

Vzhledem k tomu, že jazyk Python je multiplatformní a k implementaci nebyly použity knihovny třetích stran, měl by být klient spustitelný na jakékoliv platformě s touto verzí interpretu.

Výsledný program lze použít k jednoduché komunikaci, nicméně díky nízké uživatelské přístupnosti by bylo vhodnější řešení využít program ke komunikaci mezi několika programy.

Díky rozdělení kódu do logických struktur by mělo být možné kód jednoduše editovat a přizpůsobit požadovaným účelům.

8 Zdroje

- [1] RFC 3920 tools.ietf.org/html/rfc3920
- [2] RFC 3921 tools.ietf.org/html/rfc3921
- [3] RFC 6120 tools.ietf.org/html/rfc6120
- [4] RFC 6121 tools.ietf.org/html/rfc6121
- [5] Dokumentace jazyka Python 3.4 docs.python.org/3.4