

Projekt do předmětu (ISA) Síťové aplikace a správa sítí



Zpracoval: Ondřej Pavlacký

Login: xpavla15

3.BIT 2020/21

Discord bot

Zadání

Vytvořte komunikující aplikaci podle konkrétní vybrané specifikace pomocí síťové knihovny BSD sockets (pokud není ve variantě zadání uvedeno jinak). . Pokud individuální zadání nespecifikuje vlastní referenční systém, musí být projekt přeložitelný a spustitelný na serveru merlin.fit.vutbr.cz pod operačním systémem GNU/Linux. Program by však měl být přenositelný.

Implementační jazyk: C++

Přeložení a spuštění programu

Přeložení

V řešení je přilože makefile, pro překlad tedy stačí použít `make` přepínače: `-lssl -lcrypto -w -lssl -lcrypto` : přepínače kvůli SSL připojení `-w` : Přepínač kvůli warningům spojeným s SSL knihovnou

Spuštění

- Spuštění programu bez parametrů zobrazí nápovědu.
- `-h|--help` : Vypíše nápovědu na standardní výstup.
- `-v|--verbose` : Bude zobrazovat zprávy, na které bot reaguje na standardní výstup ve formátu "`\ - \`".
- `-t \` : Zde je nutno zadat autentizační token pro přístup bota na Discord.

spuštění přes příkazovou řádku: `./isabot -v -t ExampleToken`

Popis implementace řešení

Slovní popis

Jedním z největších problémů byl **JSON parser**. Na ten jsme nemohli použít externí knihovny, proto moje řešení parsování odpovědí serveru není velmi elegantní, ale účel splní. Princip je velmi jednoduchý, procházím odpověď bit po bitu a hledám atributy, které následně použiji. Někdy mi stačí jedno a cyklus ukončím, nebo jako například u zpráv vrácím list atributů, které jsem našel.

Pro formátování API volání jsem použil **stringstream**, dokáže dobře vizualizovat jak bude volání vypadat.

pro volání jsme použil knihovnu **OpenSSL** pro HTTPS komunikaci.

Popis funkcí

void closeSSL() - Ukončení SSL spojení a uvolnění socketu

SSL* initSSL() - Tato funkce se využívá k vytvoření a inicializaci socketu a SSL komunikace

vector<string> ParseMessages() - Slouží k naparsování zpráv, které mi přijdou ze serveru a vrátí vector naparsovaných zpráv

string GetLastMessageId() - Funkce která získá z odpovědi id poslední zprávy

void BotTalk() - Nejdůležitější část programu. Probíhá zde nekonečný cyklus, kde se pořád dokola získávají zprávy a následně ze zpracovávají a posílají se odpovědi zpět na server s rozparsovanými daty. V této funkci funkci se i vypisuje verbose na standardní výstup.

tuple<string, string> FindIsaBotChannel() -Hledá v #isa-bot kanále všechny nové zprávy z odpovědi serveru

vector GetBotDiscordServers() -Funkce získá všechny servery na kterých se bot vyskytuje, primárně testováno na jedno serveru

void BotControl() - Touto funkcí se inicializují proměnné, které se budou v další fázi chodu programu využívat a volají se z ní funkce, které řídí bota

tuple<bool, const char*> ArgumentParser() - Slouží k pomoci s parsováním argumentů, které mohou přijít vrací jestli byly použity nějaké přepínače

****int main() **** - Inicializace programu, zavolá zpracování argumentů a následně předá chod programu funkci **BotControl()**

Použité knihovny a technologie

Seznam použitých knihoven

```
\<openssl/ssl.h> \<stdio.h> \<memory.h> \<errno.h> \ \<sys/types.h> \<sys/socket.h> \<netinet/in.h> \<arpa/inet.h> \<netdb.h> \ \<string.h> \ \ \ \<stdlib.h> \<unistd.h>
```

Technologie

socket a **SSL** : používáme k zašifrování komunikace a přes sockety je komunikujeme s REST API Discord serveru a posíláme GET a POST, k použití této technologie bylo použito **OpenSSL** (opensource knihovna pro implementaci SSL a TLS protokolů)

Chybové výstupy programu

Chyby založené na špatné SSL/socket komunikaci mají návratovou hodnotu **101** Chyba založená na špatném tokenu, nebo neexistující místnosti **404**

Odevzdané soubory

Odevzdává se **xlogin00.tar** s loginem studenta (v mém případě **xpav1a15.tar**) archiv obsahuje tuto strukturu souborů:

```
| README makefile src/isabot.cpp
```

Poznámky

Automatické testy nejsou součástí řešení! Každopádně bylo manuálně testováno na referenčním serveru **merlin**

Použitá literatura

<https://www.feistyduck.com/library/openssl-cookbook/online/ch-openssl.html>
<https://github.com/openssl/openssl/tree/691064c47fd6a7d11189df00a0d1b94d8051cbe0/demos> <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FISAIT%2Flectures%2Fisa-sockets.pdf&cid=14020> <http://www.cplusplus.com/>