



# ISA 2020/21 - Projektová dokumentácia

## **Variant: Discord bot**

15. novembra 2020

Radoslav Grenčík,  
[xgrenc00@stud.fit.vutbr.cz](mailto:xgrenc00@stud.fit.vutbr.cz)

# **Obsah**

<b>1 Uvedenie do problematiky</b>	<b>3</b>
<b>2 Návrh aplikácie</b>	<b>4</b>
2.1 Použité technológie . . . . .	4
<b>3 Popis implementácie</b>	<b>5</b>
3.1 Zaujímavé pasáže kódu . . . . .	5
3.2 Chybové správy . . . . .	6
<b>4 Základné informácie o programe</b>	<b>7</b>
4.1 Nastavenie „bot“ účtu . . . . .	7
4.2 Rozšírenia . . . . .	7
<b>5 Návod na použitie</b>	<b>8</b>
5.1 Ukončenie programu . . . . .	8

## **1 Uvedenie do problematiky**

Cieľom tohto projektu je vytvoriť prenositeľnú konzolovú aplikáciu, ktorá pomocou existujúceho „bot“ účtu posiela echo odozvy na všetky správy, ktoré zaznamená v textovom kanáli `isa-bot`. Spomínaný „bot“ účet, musí byť vytvorený na platforme Discord [3] a musí byť autorizovaný v nejakom servere, ktorý je takisto vytvorený na platforme Discord [3]. Postup vytvorenia a autorizovania „bot“ účtu je popísaný na tejto adrese [2]. Detaily k nastaveniu „bot“ účtu sú popísané v kapitole 4.1.

Aplikácia posiela echo odozvy na všetky správy, ktoré nie sú poslané ňou samou, alebo nie sú posланé iným botom (botom sa rozumie užívateľ, ktorého užívateľské meno obsahuje reťazec „bot“). Echo odozvy sú posielané v tvare: `echo: <username> - <message>`.

## 2 Návrh aplikácie

Aplikácia je navrhnutá tak, že prostredníctvom protokolu HTTP komunikuje s Discord [3] API [4]. Pre komunikáciu s Discord [3] API [4] je nutné použiť šifrované SSL spojenie, ktoré je vytvorené pomocou knižnice **OpenSSL** [6]. Komunikácia je realizovaná pomocou knižnice **BSD sockets** [1]. Aplikácia beží až kým nezachytí signál `SIGINT` a potom je korektnie ukončená.

Aplikácia je navrhnutá tak, aby bola schopná bežať čo najdlhšiu možnú dobu a to vďaka tomu, že je schopná pri prerušení alebo inom násilnom ukončení SSL spojenia toto spojenie automaticky reštartovať a fungovať ďalej bez prerušenia (detailedy bližšie popísané v kapitole 3.1.) Aplikácia dokáže takisto fungovať ďalej ak dostane ako odpoveď na HTTP požiadavok chybový kód `HTTP/1.1 500 Internal Server Error` (detailedy v kapitole 3.1).

Podľa zadania projektu nie je špecifikované, v koľkých serveroch vytvorených na platforme Discord [3] môže byť „bot“ účet, na ktorý sa aplikácia pripája, autorizovaný. Moja aplikácia je teda navrhnutá tak, že akceptuje len také „bot“ účty, ktoré sú autorizované presne v jednom Discord [3] servere.

Podobne je to z textovým kanálom `isa-bot`. Takisto nie je špecifikované kolko kanálov s týmto menom môže byť vytvorených v jednom Discord [3] servere. Aplikácia akceptuje také „bot“ účty, ktoré sú autorizované v servere s presne jedným takto nazvaným textovým kanálom.

### 2.1 Použité technológie

Boli použité štandardné knižnice jazyka C a C++, knižnice pre prácu s funkciami bežne používanými v sieťovom prostredí (ako je `netinet/*`, `sys/*`, `arpa/*`, ...), knižnica **OpenSSL** [6], knižnica **BSD sockets** [1], GNU knižnica **getopt.h** [5] na spracovanie („dlhých“) argumentov, knižnica **poll.h** [9] (detailedy v kapitole 3.1) a knižnica **signal.h** [7] zachytávanie a spracovanie signálu `SIGINT`. Ako nástroj pre automatizáciu prekladu bol použitý **GNU Make** [8].

### 3 Popis implementácie

Kód som sa snažil napísať tak, aby bol zrozumiteľný a jednoducho čitateľný. Názvy premenných som zvolil tak, aby bolo z názvu jasné čo premenné obsahujú. Tam kde by mohol byť kód nezrozumiteľný alebo mätúci sú vložené objasňujúce komenty. Všetky mnou vytvorené funkcie, procedúry a makrá sú deklarované a okomentované v hlavičkovom súbore **isabot.hpp**.

#### 3.1 Zaujímavé pasáže kódu

Najzaujímavejšou pasážou môjho kódu je funkcia `SSLReadAnswer()`.

Táto funkcia sa snaží prečítať dátu z TLS/SSL spojenia. Podstatou tejto funkcie je cyklické volanie OpenSSL knižničnej funkcie `SSL_read()`, ktorá číta dátu do buffra a tieto sú potom postupne konatenované do reťazca kam ukazuje ukazateľ `received`. Funkcia `SSL_read()` je volaná v cykle, pokiaľ nenastane jeden z nasledujúcich stavov:

1. Spojenie bolo prerušené alebo nastal fatal error - Je potrebné „reštartovať“ SSL spojenie a to zavolaním funkcie `Restart()`.
2. Nič viac sa teraz nedá prečítať.

Po prerušení čítacieho cyklu je potrebné zavolať funkciu `poll()` [9] (základný časový limit je 500ms) a zistiť, či sa dá ešte niečo prečítať. Pokiaľ nebolo nič prečítané, tak sa funkcia `poll()` [9] volá znova, ale časový limit sa vždy zdvojnásobí až na maximálne 16 sekúnd. Ak sa stále nepodarilo nič prečítať, tak program končí chybou.

Ako náhle je zaručené, že boli prečítané nejaké dátu z SSL spojenia, skontroluje sa, či server nevrátil chybový HTTP kód. Môže nastať jeden z nasledujúcich stavov:

1. HTTP/1.1 200 OK Boli prečítané valídne dátu.
2. HTTP/1.1 500 Internal Server Error Nastala chyba servera, neboli prečítané valídne dátu, program však nemusí byť prerušený.
3. INÉ KÓDY Neboli prečítané valídne dátu, program musí byť prerušený.

Podľa nasledovných stavov funkcia vracia:

1. EXIT\_SUCCESS
2. EXIT\_SERVER\_ERROR
3. EXIT\_FAILURE

## 3.2 Chybové správy

<b>Error</b>	<b>Chybový kód</b>	<b>Chybová správa</b>
argument '-v' deklarovaný viac ako raz	420	bad option - option '-v' declared more than once
argument '-t' deklarovaný viac ako raz	420	bad option - option '-t' declared more than once
neznámy argument	420	Non-option arguments detected: xyz, xyz, ...
zlá kombinácia argumentov	420	bad combination of options - option '-v' can be used only with option '-t'
nepodarilo sa vytvoriť socket	EXIT_FAILURE	socket() failed
nepodarilo sa nastaviť „receive timeout“ pre socket	EXIT_FAILURE	setsockopt() SO_RCVTIMEO failed
nepodarilo sa nastaviť „send timeout“ pre socket	EXIT_FAILURE	setsockopt() SO_SNDFTIMEO failed
nepodarilo sa preložiť doménové meno na IP	EXIT_FAILURE	server IP not found
nepodarilo sa pripojiť k serveru	EXIT_FAILURE	connect() failed
nepodarilo sa vytvoriť SSL_CTX objekt	EXIT_FAILURE	SSL_CTX_new() failed
nepodarilo sa vytvoriť SSL štruktúru	EXIT_FAILURE	SSL_new() failed
funkcia SSL_set_fd() zlyhala	EXIT_FAILURE	SSL_set_fd() failed
funkcia SSL_connect() zlyhala	EXIT_FAILURE	SSL_connect() failed
neznámy error	EXIT_FAILURE	UNKNOWN ERROR
nepodarila sa prečítať odpoveď na HTTP požiadavku	EXIT_FAILURE	nothing has been read from server
funkcia poll() zlyhala	EXIT_FAILURE	poll() failed
funkcia poll() vrátila nečakaný „return event“	EXIT_FAILURE	unexpected poll() return event
zlá odpoveď na HTTP požiadavku	EXIT_FAILURE	bad answer from server
HTTP chybový kód	EXIT_FAILURE	HTTP/1.1 ...
BOT nie je členom žiadneho serveru	EXIT_FAILURE	BOT is not member of any Discord guild
BOT je členom viac ako jedného serveru	EXIT_FAILURE	BOT is member of more than one Discord guild
v servere neexistuje text kanál s názvom „isa-bot“	EXIT_FAILURE	there is no „isa-bot“ channel in this Discord guild
v servere existuje viac ako 1 text kanál s názvom „isa-bot“	EXIT_FAILURE	there is more than 1 „isa-bot“ channel in this Discord guild
funkcia SSL_write() zlyhala	EXIT_FAILURE	SSL_write() failed

## **4 Základné informácie o programe**

### **4.1 Nastavenie „bot“ účtu**

- Rozsahy (scopes): bot
- Práva bota (Bot permissions): View Channels, Embed Links, Read Message History, Send Messages

### **4.2 Rozšírenia**

Aplikácia dokáže posielat „správne“ echo odpovede na správy, ktoré obsahujú nejakú prílohu (obrázok, súbor, ...) a to tak, že zo zachytenej správy získa URL prílohy a tú potom konkatenuje ku „content“ zachytenej správy a odošle echo.

## 5 Návod na použitie

Preloženie programu sa vykoná príkazom `make`.

Spustenie programu: `./isabot [-h|--help] [-v|--verbose] -t <access_token>`

Popis argumentov:

<code>-h   --help</code>	Program vypíše nápovedu na STDOUT a skončí.
<code>-t &lt;access_token&gt;</code>	Token existujúceho „bot“ účtu, na ktorý sa program pripojí.
	Účet musí byť autorizovaný presne v jednom Discord [3] servere.
<code>-v   --verbose</code>	Program vypíše správy, ktoré zachytil v textovom kanále <code>isa-bot</code> (a sú od iných užívateľov, ktorí nemajú reťazec „bot“ vo svojom užívateľskom mene), na STDOUT vo formáte: <code>&lt;channel&gt; - &lt;username&gt;: &lt;message&gt;</code> .

Ak neboli použití žiadny argument, tak program vypíše nápovedu na STDOUT a skončí.

### 5.1 Ukončenie programu

Program ukončí svoju činnosť, keď zachytí signál `SIGINT` (napríklad po stlačení kláves `ctrl + c`).

## Literatúra

- [1] Chapter 7. Sockets. [online], [online 2020-11-12]. Dostupné z: [https://www.freebsd.org/doc/en\\_US.ISO8859-1/books/developers-handbook/sockets.html](https://www.freebsd.org/doc/en_US.ISO8859-1/books/developers-handbook/sockets.html)
- [2] Creating a Bot Account. [online], [online 2020-11-12]. Dostupné z: <https://discordpy.readthedocs.io/en/latest/discord.html>
- [3] Discord - Your place to talk. [online], [online 2020-11-12]. Dostupné z: <https://discord.com/>
- [4] Discord developer portal - Documentation. [online], [online 2020-11-12]. Dostupné z: <https://discord.com/developers/docs/intro>
- [5] getopt.h. [online], [online 2020-11-12]. Dostupné z: [https://www.gnu.org/software/gnulib/manual/html\\_node/getopt\\_002eh.html](https://www.gnu.org/software/gnulib/manual/html_node/getopt_002eh.html)
- [6] Open SSL documentation. [online], [online 2020-11-12]. Dostupné z: <https://www.openssl.org/docs/>
- [7] signal.h. [online], [online 2020-11-12]. Dostupné z: <https://pubs.opengroup.org/onlinepubs/009695399/basedefs/signal.h.html>
- [8] GNU Make. [online], 1 2020, [online 2020-11-12]. Dostupné z: <https://www.gnu.org/software/make/>
- [9] Kerrisk, M.: poll(2) — Linux manual page. [online], 11 2020, [online 2020-11-12]. Dostupné z: <https://man7.org/linux/man-pages/man2/poll.2.html>