VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ČTEČKA NOVINEK VE FORMÁTU ATOM A RSS S PODPOROU TLS

NEWS READER USING ATOM AND RSS FEEDS WITH TLS SUPPORT

TECHNICKÁ ZPRÁVA
TECHNICAL REPORT

AUTOR PRÁCE **MARTIN ZMITKO**
AUTHOR

VEDOUCÍ PRÁCE **Ing. LIBOR POLČÁK, Ph.D.**
SUPERVISOR

BRNO 2022

Obsah

Kapitola 1

Úvod

V této zprávě se věnuji zpracování projektu do předmětu ISA – Čtečka novinek ve formátu Atom a RSS s podporou TLS. Cílem bylo navrhnout a implementovat program v jazyce C/C++ se vstupem jako seznam odkazů na vzdálené Atom a RSS zdroje a výstupem jako seznam zpráv, které vzdálené zdroje obsahují. Důležitou součástí programu je možnost zabezpečené komunikace pomocí SSL/TLS.

1.1 SSL/TLS

SSL (později nahrazeno TLS) jsou protokoly navržené k bezpečné a šifrované síťové komunikaci. Komunikace probíhá v několika krocích, které jdou shrnout ve třech nejdůležitějších bodech: [9]

1. navázání komunikace a dohoda účastníků na použitých algoritmech
2. odeslání serverového certifikátu klientovi, který ověří pomocí veřejného klíče certifikační autority jeho platnost
3. po úspěšném ověření probíhá zbytek symetricky šifrované komunikace

1.2 RSS a Atom feedy

RSS a Atom jsou standardizované formáty webových zdrojů umožňující snadno automatizovatelné sledování novinek z webových stránek. Tyto zdroje poskytují informace o samotném zdroji a seznam nových příspěvků a informací o nich (např. jméno autora, datum publikace, krátký popis a podobné). Zdroje jsou poskytovány ve formátu XML s jasnou specifikací [5] [7].

Kapitola 2

Návrh a implementace

Program se skládá z jednoho zdrojového souboru v jazyce C++, `feedreader.cpp`, který obsahuje veškerý zdrojový kód. Kód není rozdělený do modulů z důvodu jeho relativně krátké délky a jednoduchosti. Jednotlivé funkční celky jsou rozděleny do funkcí.

2.1 Zpracování argumentů

Zpracování argumentů probíhá ve funkci `parse_args()` s použitím funkce `getopt()` ze systémové stejnojmenné knihovny. Funkce `parse_args()` vrací strukturu `args_t`, která obsahuje veškeré informace získané z argumentů, také je do vektoru `urls` uložen seznam odkazů, které se v argumentech vyskytly.

Kvůli chování funkce `getopt()`, kdy při nastavené systémové proměnné `POSIXLY_CORRECT` ukončuje zpracování argumentů při prvním výskytu pozičního argumentu [6], což odporuje specifikaci v zadání (URL by mělo být na prvním místě), je první poziční argument považován za URL a ukazatel `argv` posunut na další argument.

Program umožňuje kombinaci přímo zadaných URL argumentem (může být i více naráz) a feedfile. Zdroje se zpracují všechny.

2.2 Zpracování odkazů

Po případném přečtení zadaného feedfile (s využitím `std::ifstream`) jsou jednotlivá URL parsována ve funkci `parse_urls()` pomocí regulárního výrazu [3] s matching skupinami, do kterých se uloží jednotlivé součásti URL. V URL musí být povinně specifikovaný protokol (podporovány jsou HTTP a HTTPS). Volitelně lze určit port, jinak se doplní podle protokolu (80 pro HTTP, 443 pro HTTPS).

2.3 TLS spojení

K TLS spojení využívám knihovnu OpenSSL [2]. Navázaní spojení probíhá pro každý zdroj ve funkci `do_ssl()`. Samotná HTTP komunikace proběhne ve funkci `do_request()` s pomocí BIO (abstrakce pro vstupně-výstupní datový proud socketu). Spojení probíhá na základě postupu naznačeného v IBM manuálu pro vývojáře [4].

Formát přijatého HTTP požadavku a kód odpovědi jsou zkontrolovány pomocí regulárního výrazu ve funkci `parse_http()`.

2.4 Zpracování XML a výpis

Zpracování přijatého XML zajišťuje knihovna libxml2 [1] a probíhá ve funkci `parse_xml()`. Rozpoznání, jestli se jedná o RSS nebo Atom zdroj je zajištěno porovnáním názvu a atributů kořenového uzlu. Na základě typu zdroje se určí názvy uzlů a struktura, podle kterých se ve zdroji dále vyhledává. Při výpisu více zdrojů nebo položek se správné vypsání prázdných řádků řeší pomocí nastavení vlajky po zpracování první položky.

Ve zdroji se vyhledává pomocí pomocných funkcí `find_node()` a `node_content()`. Vstupem funkcí je první uzel v seznamu a název vyhledávaného uzlu, výstupem je ukazatel na nalezený uzel, resp. ukazatel na textový obsah nalezeného uzlu.

Při nastaveném výpisu dodatečných informací je pro každou položku prohledán seznam pod-uzlů a při nalezení prvního požadovaného je jeho obsah vypsán. Pokud položka obsahuje více požadovaných záznamů (např. více autorů), je vypsán vždy pouze první nalezený.

Kapitola 3

Použití

Po překladu (3.1) se vytvoří spustitelný soubor `feedreader`, který se spouští s následujícími parametry:

```
./feedreader URL | -f <feedfile>> [-c <certfile>] [-C <certdir>]
[-T] [-a] [-u] [-h]
-f <feedfile> - nastavení souboru se seznamem odkazů na zdroje s jedním URL
na každém řádku
-c <certfile> - nastavení souboru s certifikátem použitým k šifrované
komunikaci se zdrojem
-C <certdir> - nastavení složky s certifikáty použitými k šifrované
komunikaci se zdrojem
-T - vypsat čas poslední změny u záznamu
-a - vypsat autora u záznamu
-u - vypsat URL u záznamu
-h - vypíše informace o použití programu a skončí
```

Pořadí parametrů je libovolné a je možné specifikovat více URL, stejně tak kombinovat feedfile a URL v parametrech, URL ale nemůže být mezi flagy a před flagy může být maximálně jedno URL. Program vypíše požadované informace na standardní výstup.

3.1 Překlad

Překlad je zajištěn překladačem c++ a systémem GNU make **(BSD make není podporován)**. Prerekvizitami jsou knihovny libssl (ověřeno s verzí 10 na serverech merlin a eva) a libxml (ověřeno s verzí 10 na serverech merlin a eva). Překlad se spouští příkazem `make`, smazání souborů vytvořených při překladu se provede příkazem `make clean`.

Kapitola 4

Testování

K programu jsou přiloženy testy k ověření funkčnosti spojení, zpracování argumentů a zdrojů a správného formátu výstupu. Testy jsou napsány v jazyce Python 3.8 s použitím frameworku pytest [8] a volají se příkazem `make test`. Testy jsem doplňoval v průběhu implementace pro ověření funkčnosti nového kódu.

Literatura

[1] *Libxml2 Reference Manual* [online]. [cit. 2022-11-13]. Dostupné z: https://gnome.pages.gitlab.gnome.org/libxml2/devhelp/index.html.

[2] *OpenSSL Documentation* [online]. [cit. 2022-11-13]. Dostupné z: https://www.openssl.org/docs/.

[3] *Regular expressions library* [online]. [cit. 2022-11-13]. Dostupné z: https://en.cppreference.com/w/cpp/regex.

[4] Ballard, K. *Secure programming with the OpenSSL API* [online]. [cit. 2022-11-13]. Dostupné z: https://developer.ibm.com/tutorials/l-openssl/.

[5] Board, R. A. *RSS 2.0 Specification* [online]. [cit. 2022-11-13]. Dostupné z: http://www.rssboard.org/rss-specification.

[6] Koenig, T. a Kerrisk, M. *Getopt(3) – Linux manual page* [online]. [cit. 2022-11-13]. Dostupné z: https://man7.org/linux/man-pages/man3/getopt.3.html.

[7] M. Nottingham, E. a R. Sayre, E. *The Atom Syndication Format* [online]. [cit. 2022-11-13]. Dostupné z: https://www.ietf.org/rfc/rfc4287.txt.

[8] Okken, B. *Python Testing with pytest: Simple, Rapid, Effective, and Scalable*. 2. vyd. Pragmatic Bookshelf, 2022. ISBN 978-1680508604.

[9] Ristic, I. *Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications*. Feisty Duck, 2014. ISBN 978-1907117046.