



Projektová dokumentace

Čtečka novinek ve formátu Atom a RSS s podporou TLS

Síťové aplikace a správa sítí

Obsah

1	Úvod	1
2	Návod na použití	1
2.1	Spuštění programu	1
3	Návrh a implementace	1
3.1	feedreader.cpp	1
3.2	argsParser.cpp	2
3.3	urlsProcessor.cpp	2
3.4	feedParser.cpp	2
4	Závěr	2

1 Úvod

Cílem projektu bylo pomocí jazyka C/C++ vytvořit program, který stahuje novinky ve formě souboru XML přes síť a tyto soubory v závislosti na formátu správně zpracovává a vypisuje informace požadované uživatelem. Program zpracovává XML soubory ve formátu Atom[7] a RSS 2.0[5].

2 Návod na použití

Program je určen pro systémy Linux (byl testován na Ubuntu 22.04 LTS). Pro překlad aplikace CMake(zkoušeno na verzi 3.16). Překlad je možné spustit příkazem make. Po překladu se vytvoří spustitelný soubor feedreader.

2.1 Spuštění programu

```
./feedreader <URL | -f <feedfile>> [-c <certfile>] [-C <certaddr>] [-T] [-a] [-u]
```

- `url` URL požadovaného zdroje.
- `-f <feedFile>` Textový soubor, ve kterém na každém řádku uvedena URL zdroje. Prázdné a řádky začínající znakem # jsou ignorovány. Poslední znak na každém řádku a konci souboru musí být LF.)
- `-c <certFile>` Definuje soubor s certifikáty, který se použije pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- `-C <certDir>` Určuje adresář, ve kterém se mají vyhledávat certifikáty, které se použijí pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- `-T` Pro každý záznam zobrazí informace o čase změny záznamu, či vytvoření záznamu (je-li ve staženém souboru obsaženo).
- `-a` Pro každý záznam zobrazí jméno autora, či jeho e-mailová adresa (je-li ve staženém souboru obsaženo).
- `-u` Pro každý zdroj se navíc zobrazí asociované URL záznamu (je-li ve staženém souboru obsaženo).

Povinně je třeba uvést parametr `url` nebo `-f <feedFile>`. Ostatní parametry jsou volitelné.

K zobrazení nápovědy je možné použít parametr `-h` nebo `--help`.

3 Návrh a implementace

Program je napsán v jazyce C++, a rozdělen do několika tříd, které budou popsány v této kapitole.

3.1 `feedreader.cpp`

Vstupní bod programu. V tomto souboru se nachází funkce `main`, která přijme vstupní argumenty. Pak argumenty budou zpracovány třídou `argsParser`. Pokud zpracování argumentů nepodaří, bude program ukončen s chybovým kódem.

Dále pomocí třídy `urlsProcessor` bude vytvořen seznam URL zdrojů, které je nutno zpracovat.

Dále ve smyčce pro každou URL pokud používáme https ověříme certifikáty, zkusíme se připojit k server. Pokud připojení se podařilo, pomocí GET požadavku dostaneme novinky ve formátu XML, a po ověření správnosti odpovědi začneme její zpracování pomocí třídy `feedParser`.

Pokud se něco selže, bude program pokračovat prací s následujícím zdrojem.

Stahování probíhá pomocí knihovny OpenSSL[1], při práci nad projektem mi pomohl tutoriál[4].

3.2 argsParser.cpp

Tato třída je určena pro zpracování vstupních argumentů. Pro zpracování jsem použil knihovnu `getopt`[3]. Při chybně zadaných argumentech bude výpsána nápověda programu.

3.3 urlsProcessor.cpp

Tato třída vytváří seznam URL zdrojů zadaných uživatelem. Pomocí regulárního[2] výrazu bude ověřena správnost URL. Pokud zpracování URL selže, bude výpsána chybová zpráva.

3.4 feedParser.cpp

Tato třída slouží k parsování XML a zpracování. Jsou podporovány formáty Atom 1.0 a RSS 2.0.

Parsování XML probíhá pomocí knihovny `Libxml2`[8]

Nejdříve je vždy výpsán titulek daného zdrojového souboru uvozen, a zakončen znaky „****“. Pokud titulek chybí, tak se zpracovávání daného zdroje končí.

Dále budou výpsány jednotlivé položky zdrojových souborů. Nejdříve bude výpsán titulek položky a pak další doplňující informace, pokud při spuštění byly uvedené odpovídající argumenty. Pokud některá z těchto položek není uvedena, tak se nevypisuje.

S touto částí mi pomohl tutoriál[6]

4 Závěr

Podařilo se mi implementovat všechny části aplikace, které byly uvedené v zadání projektu. Projekt byl testován pomocí zdrojů daných v zadání a nebyly nalezeny žádné chyby

Při práci nad projektem jsem se seznámil s knihovnou `OpenSSL` a `Libxml2` zjistil, jak funguje komunikace přes HTTPS, a se naučil pracovat s knihovnou `regex`

Literatura

- [1] OpenSSL. [online], [vid. 2022-11-14]. Dostupné z: <https://www.openssl.org>
- [2] Regular Expressions. [online], [vid. 2022-11-14]. Dostupné z: <https://cplusplus.com/reference/regex/>
- [3] getopt manual. [online], listopad 2018, [vid. 2022-11-14]. Dostupné z: <https://man7.org/linux/man-pages/man3/getopt.3.html>
- [4] Ballard, K.: Secure programming with the OpenSSL API. [online], srpen 2018, [vid. 2022-11-14]. Dostupné z: <https://developer.ibm.com/tutorials/l-openssl>
- [5] Board, R. A.: RSS 2.0 Specification. [online], březn 2009, [vid. 2022-11-14]. Dostupné z: <http://www.rssboard.org/rss-specification>
- [6] Fleck, J.: Libxml Tutorial. [online], rev. 24. srpen 2004, [vid. 2022-11-14]. Dostupné z: <http://xmlsoft.org/tutorial>
- [7] M. Nottingham, E.; R. Sayre, E.: The Atom Syndication Format. [online], prosinec 2005, [vid. 2022-11-14]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc4287>
- [8] Veillard, D.: The XML C parser and toolkit of Gnome. [online], [vid. 20-11-14]. Dostupné z: <http://www.xmlsoft.org>