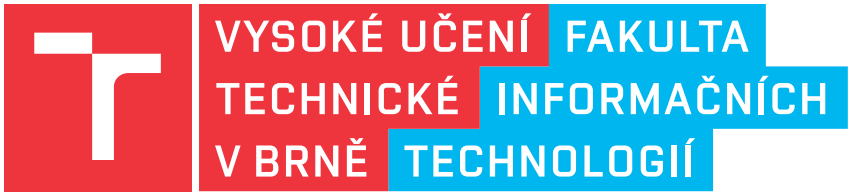

Projekt k předmětu Síťové aplikace a správa sítí

Čtečka novinek ve formátu Atom a RSS s podporou TLS

13. listopadu 2022 Kravchuk Marina (xkravc02)

Obsah

1 Úvod

Cílem projektu bylo vytvořit komunikující aplikaci v jazyce C/C++, která stahuje XML soubory (tzv. `feedy`) přes internetovou síť a tyto soubory zpracovává a vypisuje v nich uvedné informace podle uvedenych požádavku uživatele. Vstupní zdroje i výstupní informace je možné nastavovat vstupními argumenty programu, viz 3.

Program zpracovává XML soubory ve formátu `Atom 1.0` a `RSS 2.0`.

2 Návrh a implementace

Aplikace je napsána v programovacím jazyce C a rozdělena do několika zdrojových souborů. V této kapitole jsou stručně popsány jednotlivé části implementace.

2.1 Vstupní bod programu, `feedreader.c`

V dánem souboru se nachází funkce `main`, která příjimá vstupní argumenty programu a následně je předá další funkci `parser` 2.2, která vrátí strukturu obsahující všechny informace ze vstupních dat, které potřebujeme pro další postup. Pokud zpracování argumentů selže, je program ukončen s chybovým kódem.

Dalším krokem je kontrola přítomnosti souboru `feedfile`. Pokud uživatel zadal soubor feedfile, je tento argument předán funkci `parse_feedfile`2.3, která vrátí strukturu se všemi URL uvedenými uvnitř souboru.

Poté se uloží údaje o certifikátu uvedené v parametrech a začne zpracování adresy URL. Zdroje URL zpracováváné funkcí `secure_connect` 2.4. Pokud zpracování selže, program skončí s chybovým kódem.

Stažený dokument xml je předán funkci `parse_xml` 2.6, která analyzuje jednotlivé tagy. Pokud dojde k chybě při zpracování tagu, program vrátí chybové hlášení.

Konečným bodem programu je vypsání zpracováných dat na standardní výstup 2.7. Výstupní formát se bude lišit v závislosti na parametrech zadaných uživatelem dříve.

2.2 Zpracování argumentů, `args_parser.c`

Struktura `par_content` a funkce `parser` jsou definovaná v souboru `args_parser.h` a implementovaná v souboru `args_parser.c`.

Funkce zpracovává vstupní argumenty pomocí cyklu `while`. Všechny zpracovávané informace jsou pak uloženy v jednotlivých položkách struktury `par_content`.

Při chybně zadaných argumentech je výpsána chybová zpráva a program skončí.

2.3 Zpracování feedfilu, `feed_parser.c`

Struktura `feed_content` a funkce `parse_feedfile` jsou definovaná v souboru `feed_parser.h` a implementovaná v souboru `feed_parser.c`.

Tato funkce otevře soubor feedů a odešle přijatá data k analýze. Smyčka `while` čte url adresy znak po znaku a zapisuje je do výsledné struktury. Komentáře ve feedfile jsou označeny # a programový kód je ignoruje. Počítá se s tím, že každý řádek tohoto souboru bude zakončen znakem `LF`.

Pokud zpracování souboru `feedfile` selže, je vypsaná chybová zpráva.

2.4 Zpracování a stažení zdrojů, `downloader.c`

Struktura `down_content` a funkce `secure_connect` jsou definovaná v souboru `downloader.h` a implementovaná v souboru `downloader.c`.

Tato funkce stahuje jednotlivé zdrojové soubory a následně je dále zpracovává. Stahování probíhá pomocí knihovny `OpenSSL`.

Nejprve se zpracuje přijatá URL adresa pomocí funkce `parse_url` 2.5. Pokud toto selže, tak vypisují se chybová zpráva.

Dále provádí se připojení k danému zdroji. Pokud se používá připojení SSL, nastaví se umístění certifikátů. Pokud byly jako argumenty zadány `-c` nebo `-C`, pak jsou nastaveny pro informace o certifikátu. Pokud ne, tak používá se výchozí umístění, které definuje `OpenSSL` knihovna. Pokud ověření certifikátů nebo připojení k serveru selže, tak vypíši chybovou zprávu.

Pokud se program úspěšně připojí k serveru, odešle nejprve zprávu HTTP. Požadavek GET je odeslán na prostředek serveru. Poté přečte HTTP odpověď serveru. Pokud při zápisu nebo čtení požadavku nebo odpovědi HTTP dojde k chybě, vypíše se chybová zpráva.

Po úspěšném načtení odpovědi HTTP se provede kontrola vráceného kódu odpovědi. Za validní odpovědi jsou považovány všechny odpovědi s HTTP kódem s hodnotou 200 – 299. Pokud přijatá hodnota neodpovídá zadanému rozsahu, zobrazí se chybové hlášení a kód HTTP.

Nakonec funkce vrátí celou odpověď HTTP do `main`, kde dále se oddělí hlavička od obsahu zpravy.

2.5 Parsování URL, `url_parser.c`

Struktura `url_content` a funkce `parse_url` jsou definovaná v souboru `url_parser.h` a implementovaná v souboru `url_parser.c`.

Tato sekce slouží k validaci URL adresy a k jejímu rozložení na jednotlivé části, které se pak zapíší do příslušných položek struktury. Podporovány jsou pouze formáty `http` a `https`. V případě, že adresa nemá zadaný port, bude automaticky doplněn na 80 u `http` a 433 u `https`.

2.6 Parsování XML, `xml_parser.c`

Struktura `xml_content` a funkce `parse_xml` jsou definovaná v souboru `xml_parser.h` a implementovaná v souboru `xml_parser.c`.

Tato funkce slouží k parsování XML zdroje. Jsou podporovány pouze zdroje ve formátu `Atom 1.0`, a `RSS 2.0`.

Samotné parsování XML probíhá pomocí knihovny `<libxml/parser.h>`.

Pokud XML struktura parsovaného souboru není validní, v souboru se nenachází kořenový element nebo typ XML souboru není podporovaný, vypíše se chybová zpráva.

Struktura `xml_content` obsahuje buňky pro název feedu a pole struktur `xml_item` pro jednotlivé články. Později se z této struktury vypisují informace na konci programmu.

2.7 Vypisování inforamcí

Nejdříve je vždy vypsán titulek daného zdrojového souboru ve tvaru `*** TITLE ***`. Pokud titulek ve zdroji chybí, tak jako první řadek se vypíše `*** <Feed bez nazvu> ***`.

Poté je každý článek zpracován ve smyčce `for`. Nejprve je vypsán jeho název, ostatní informace jsou doplňující a jsou vypsány, pokud byly zadány příslušné parametry `-T, -a, -u`.

- Čas aktualizace je reprezentován jako `Aktualizace:`
- Jméno nebo e-mailová adresa autora je reprezentován jako `Autor:`
- Asociované URL je reprezentován jako `URL:`

Pokud některá z položek není ve zdroji uvedena, tak bude informace vypsana jako `<Datum|Autor|URL neurceno>`.

3 Návod na použití

Po překladu programu příkazem `make`, je vytvořen spustitelný program `feedreader`.

3.1 Syntaxe a sémantika spouštění programu

`./feedreader <url | -f <feedFile>> [-c <certFile>] [-C <certDir>] [-T] [-a] [-u]`

- `url` URL zdroje.
- `-f <feedFile>` Soubor `feedfile`. (Textový soubor, kde je na každém řádku uvedena URL zdroje. Prázdné řádky a řádky začínající znakem # jsou ignorovány. Poslední znak na každém řádku musí být `LF`.)
- `-c <certFile>` Soubor s certifikáty pro ověření platnosti certifikátu předloženého serverem při použití SSL/TLS.
- `-C <certDir>` Adresář, ve kterém se vyhledávají certifikáty, které se použijí pro ověření platnosti certifikátu předloženého serverem při použití SSL/TLS.
- `-T` Pro každý zdroj se navíc zobrazí informace o čase změny, či vytvoření zázamu.
- `-a` Pro každý zdroj se navíc zobrazí jméno, či e-mailová adresa autora záznamu.
- `-u` Pro každý zdroj se navíc zobrazí asociované URL záznamu.

Povinně je uvést buď URL zdroje nebo soubor `feedfile`. Podporovaná schémata zdrojů jsou `http` a `https`. Parametry je možné zadávat v libovolném pořadí.

Nápovědu je možné si vypsat příkazem `./feedreader -h`.

Reference

[1] https://www.openssl.org

[2] https://developer.ibm.com/tutorials/l-openssl/

[3] https://support.google.com/merchants/answer/160593?hl=en

[4] https://www.rssboard.org/rss-specification

[5] http://www.xmlsoft.org/examples/

[6] http://stackoverflow.com/