Vysoké učení technické v Brně

Fakulta informačních technologií

ISA - Síťové aplikace a správa sítí

2017/2018

Programování síťové služby

Čtečka novinek ve formátu Atom s podporou TLS (Dr. Polčák)

Brno Rudolf Kučera

2018 xkucer91

Obsah

1. Úvod

Dokumentace k projektu z předmětu Síťové aplikace a správa sítí (ISA). Dokument popisuje návrh, implementaci a použití aplikaci *feedreader* a s tím spojenými pojmy. Aplikace *feedreader* se může použít na získávání informací o novinkách na dané webové stránce.

2. Důležité pojmy

Aplikace přímo pracuje s níže uvedenými technologiemi.

2.1.Atom

Atom je typ dokumentu založený na XML, který opisuje seznam informací známých jako "feed". Feedy se skládají z několika prvků, které obsahují další metadata (každý prvek má svůj název).

2.2.RSS

RDF Site Summary (RSS) je víceúčelový popis metadat psán v XML. Dokument RSS popisuje seznam prvků adresovatelných pomocí URL. Každý prvek má vlastní název, odkaz, jednoduchý popis a někdy i autora. Pomocí RSS souboru lze zjistit co je na webu nebo blogu nového, bez toho aby jsme ho navštívili. Aplikace *feedreader* pracuje s RSS verzemi 1.0 a 2.0, které se taky liší jenom názvy XML elementů.

RSS je hodně podobné Atomu a v rámci tohoto projektu se liší pouze jinými názvy XML elementů. Ukázka jednoduchého Atom a RSS dokumentu:

Atom:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

  <title>Example Feed</title>
  <link href="http://example.org/"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
  </entry>

</feed>
```

Obrázok 2 https://tools.ietf.org/html/rfc4287

RSS 1.0:

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/"
>

  <channel rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>
      XML.com features a rich mix of information and services
      for the XML community.
    </description>

    <image rdf:resource="http://xml.com/universal/images/xml_ti

    <items>
      <rdf:Seq>
        <rdf:li resource="http://xml.com/pub/2000/08/09/xslt/xs
        <rdf:li resource="http://xml.com/pub/2000/08/09/rdfdb/i
      </rdf:Seq>
    </items>

  </channel>
```

Obrázok 1 http://web.resource.org/rss/1.0/spec

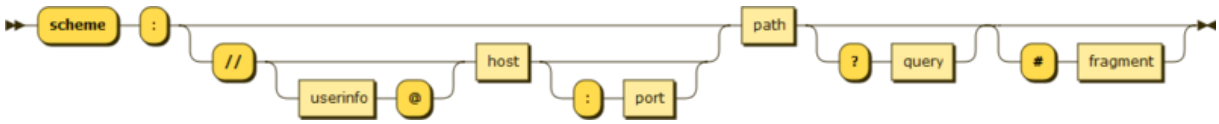
2.3.OpenSSL

Secure Sockets Layer (SSL) je standard bezpečný komunikace na internetu. Data jsou zašifrována předtím než opustí počítač a jsou dešifrována až když dosáhnou svého cíle. Používají se na to certifikáty a šifrovací algoritmy. OpenSSL je knihovna, která nám umožňuje s tím pracovat.

2.4.URL

Uniform Resource Locator (URL) nebo taky web adresa je odkaz na webový zdroj, který specifikuje jeho lokaci v síti a mechanismus jak ho získat. URL je určitý typ Uniform Resource Identifier (URI). URL se používá při odkazování se na webové stránky (http), ale taky na přenos souborů (ftp), nebo email (mailto).

Syntax URL:



Obrázok 3 https://en.wikipedia.org/wiki/URL

3. Návrh aplikace

Aplikace dostane od uživatele jednu, nebo více, URL adres. Podle toho, jestli je zadáno http, nebo https se připojujeme na port 80 (http) nebo 443 (https). Pokud se připojujeme přes zabezpečené připojení (https) musí se také ověřit certifikáty. Jestli ověrování proběhlo úspešně, pošle se HTTP GET request a přijme se odpověď. Pokud je odpověď v pořádku, oddělí se HTTP hlavička od těla a z toho se pomocí funkcí knihovny *libxml2* vytáhnou potřebné údaje, které se pak vypíšou na výstup.

4. Implementace

Program je implementován v jazyce C/C++. Není navržen objektově. Vyvíjeno a testováno na merlinovi (CentOS Linux release 7.5.1804) ve vimu.

4.1.Zpracování argumentů

Volá se funkce *args*(). Jestli je zadáno jedna nebo více URL adres, všechny se uloží do vectoru. Jestli byl zadán feedfile, načte z něho všechny URL adresy pomocí funkce *feedfile*() a uloží je do *vectoru* spolu s ostatními URL. Přes tenhle vector se pak prochází a volá funkce *ssl_connect*().

4.2.Připojení pomocí OpenSSL

Z každé URL se vytáhne host, cesta k rss souboru a způsob přenosu (http nebo https). Pak se připojuje pomocí funkce *BIO_new_connect()*. Pokud se jedná o http, připojujeme se na port 80, při https na port 443.

4.3.Ověřování certifikátů

Po připojení na port 443 (https) proběhne ověřování certifikátů. Uživatel mohl zadat konkrétní složku nebo soubor s certifikáty, v tomto případě se použije funkce *SSL_CTX_load_verify_locations()* . Když uživatel nezadal -c ani -C použije se funkce *SSL_CTX_set_default_verify_paths()*. Ověření, zda je certifikát platný pak provedeme pomocí *SSL_get_verify_result()*.

```
if (SSL_get_verify_result(ssl) != X509_V_OK)
{
    cerr << "Error certificate is not valid" << endl;
} else {
    response = get_response(bio, file, host);
}
```

4.4.HTTP request

Po správném připojení, případně ověření certifikátů, se volá funkce *get_response()*, která má na starosti odeslání HTTP požadavku na server a převzetí odpovědi.

Funkce odešle řetězec s GET HTTP požadavkem pomocí *BIO _puts()* a pak pomocí *BIO_read()* uvnitř while cyklu přijímá odpověď. Odpověď ukládá do řetězce, který pak vrací.

```
string response = "";
string req = "GET " + file + " HTTP/1.0\r\n";
req += "Host: " + host + "\r\n";
req += "Connection: close\r\n";
req += "Accept-Encoding: UTF-8\r\n";
req += "User-agent: Feedreader-xkucer91\r\n\r\n";

// odeslani requestu
BIO_puts(bio, req.c_str());
```

4.5.Zpracování XML souboru a výpis

Po získání odpovědi se pak zavolá funkce *parse_xml()*. Odstraní se nežádoucí znaky, hlavička HTTP a pomocí funkci z knižnice libxml2 *xmlDocGetRootElement()* se najde xml root element. Jestli má root element jméno "feed", jedná se o xml dokument typu ATOM. Jestli je jméno "RDF" nebo "rss" jedná se o RSS feed. Pak se pomocí struktury *xmlNodePtr* a ukazateli na potomky a další elementy prochází celý dokument a hledají se prvky, které chce uživatel vypsat. Na porovnávání jmen elementů používám funkci *xmlStrcmp()* a na získání hodnot *xmlNodeListGetString().*

Při práci s xml sem postupoval podle tutoriálu http://xmlsoft.org/tutorial/xmltutorial.pdf a využíval sem taky stránku http://www.xmlsoft.org/html/libxml-tree.html.

4.6.Použité knihovny

```
#include <iostream>
#include <vector>
#include <sstream>
#include <fstream>
#include <string>
#include <unistd.h>
#include "openssl/bio.h"
#include "openssl/ssl.h"
#include "openssl/err.h"
#include <libxml/tree.h>
```

5. Návod na použití

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

Pořadí parametrů je libovolné. Popis parametrů:

- Povinně je uveden buď URL požadovaného zdroje (přičemž podporovaná schémata jsou http a https), nebo parametr -f s dodatečným parametrem určujícího umístění souboru feedfile.
- Volitelný parametr **-c** definuje soubor <certfile> s certifikáty, který se použije pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- Volitelný parametr **-C** určuje adresář <certaddr>, ve kterém se mají vyhledávat certifikáty, které se použijí pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- Při spuštění s parametrem **-T** se pro každý záznam zobrazí navíc informace o čase změny záznamu, či vytvoření záznamu (je-li ve staženém souboru obsaženo).
- Při spuštění s parametrem **-a** se pro každý záznam zobrazí jméno autora, či jeho e-mailová adresa (je-li ve staženém souboru obsaženo).
- Při spuštění s parametrem **-u** se pro každý záznam zobrazí asociované URL (je-li ve staženém souboru obsaženo).