ISA – protokol k projektu

Juraj Novosád

9. novembra 2022

Obsah

1 Zadanie

Cieľom projektu je vytvorenie nástroja na čítanie správ zo servera poskytujúceho RSS alebo atom správy a vypísať ich obsah na konzolu. Server musí komunikovať cez http alebo https protokol, pričom nástroj je schopný overiť platnosť certifikátu servera komunikujúcom cez https a rozšifrovať správu.

2 Teória

Rss a atom sú súbory, zapísané vo formáte XML. Primárne sú používané na štruktúrované zapisovanie krátkych správ o udalostiach.

2.1 Atom

Tento formát popisuje list informácií, ktoré nazývame `feed`. `Feed` je koreňová entita, obsahujúca jednotky, ktoré nazývame `entry`.
`Entry` zapúzdruje konkrétny záznam. Pre tento projekt je dôležité, že obsahuje `title`(nadpis), `link` a `author`, `link` a `author` nie je povinný. Voliteľne môže obsahovať `id`, `summary`, `updated`, ... Samotný feed môže obsahovať autora feedu a nadpis celého feedu. Tieto informácie sa môžu vyskytnúť hocikde v najvyššej vrstve xml dokumentu. Ak nie je v `entry` definovaný autor, ako autor `entry` je považovaný autor celého feedu. Informácie o atom formáte som čerpal z rfc4287

2.2 RSS

RSS formát zapúzdruje informácie o krátkych správach, je to akronym pre Really Simple Syndication. Koreň formátu je tag `rss`, v ktorom sa špecifikuje verzia. `Rss` tag obsahuje práve jeden tag `channel`.
`Channel` zapúzdruje informácie pre čitateľa. Musí obsahovať elementy `title, link, description`. Voliteľne môže obsahovať tagy `language, copyright, pubDate, ....` `Channel` voliteľne zapúzdruje samotné správy v tagoch item.
Item obsahuje `title, link a description`. Voliteľne môže obsahovať `author`, `category`, `comments`, `enclosure`, `guid`, `pubDate`, `source`. Informácie som čerpal z rss špecifikácie.

3 Priebeh implementácie

Projekt je písaný v c++ a primárne zostavovaný pomocou `CMakeLists`, ktorý vytvorí `Makefile`. Makefile je potom možné používať na zostavovanie a spúšťanie testov.
Na uľahčenie kontroly kompatibility s referenčným prostredím na serveri merlin som vytvoril pipeline, ktorá sa pri každom push na github spustila a pokúsila sa zostaviť program a spustiť testy. Referenčné prostredie podobné merlinovi som vytvoril ako docker image, ktorý podľa možností kopíruje verzie potrebných programov. Dockerfile je v zložke `scripts/build.Dockerfile`.

4 Použité knižnice

Na komunikáciu so serverom je použitá knižnica `openssl`. Na spracovanie xml súborov je použitá knižnica `libxml2`. Na spracovanie url adries je použitá štandardná knižnica `regex`.

5 Priebeh spracovania

Argumenty sa spracujú v objekte parameterParser. Tento objekt zapúzdruje argumenty, ich hodnoty vie poskytnúť ako asociatívne pole, alebo sa na argumenty dá dotazovať po jednom.
Získané URL alebo meno súboru sa vloží do objektu URLParser. URLParser načíta URL zo súboru, alebo len ako URL, regexom skontroluje správnosť. Poskytuje metódu next, vďaka ktorej môžeme cez feedfile alebo cez jednu adresu jednoducho iterovať. Každú URL taktiež rozparsuje do štruktúry na jednotlivé časti. Zároveň ak adresa nezačína http alebo https, automaticky doplní https ako protokol URL.
Spracované URL adresy po jednom spracováva feedDownloader. Jeho zodpovednosť je nadviazať spojenie so serverom, overiť jeho certifikát, prípadne sa nechať presmerovať, stiahnuť feed a spracovať. Spracovať je myslené odstrániť zarážky ak je príjem chunked, spracovať hlavičku http response.
XML feed je následne spracovaný objektom xmlParser::processor. Ten spracuje xml do stromu, to urobí knižnica a vypíše obsah položiek. Spracovanie prebieha tak, že sa dostane do úrovne, kde sú záznamy. Tam prechádza do šírky a hľadá tag označujúci správu. Ak nájde správu, tak v nej potom prehľadáva do šírky a hľadá tagy podobným spôsobom. Pričom vypisuje relevantné informácie.

6 Testovanie

Testy sa spúšťajú v rámci frameworku CTest. CTest postupne spustí testovacie scenáre najprv pre unit testy a potom spustí skript pre integračný test. Unit testy sú implementované pomocou GTestu. Framework GTest sa stiahne z oficiálneho github repozitára počas kompilovania testov.
Samotné testovacie scenáre testujú funkciu jednotlivých modulov, pričom každý modul má skupinu testov v podobe jednej binárky. Čiže je jedna testovacia binárka na jeden modul, v respektíve knižnicu.
Integračné testy sa spúšťajú nad samotným programom feedreader. Feedreader má pridanú možnosť spracovania feedu zo súboru. Na testovacie účely je definovaný prepínač -k <meno súboru>. Testovanie potom prebieha tak, že sa po jednom spracovávajú súbory zo zložky tests/test_files a výstup z programu sa porovnáva s referenčným výstupom z tests/reference.

7 Popis použitia

Aplikácia je dodávaná v forme zdrojových súborov. Skompilovať je možné pomocou príkazu:

```
make
```

Makefile vytvorí zložku build, do nej nechá Cmake nagenerovať Makefile a ten potom spustí. Akonáhle sa program preložil môže sa otestovať spustením príkazu:

```
make test
```

Použitie:

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

Program podporuje možnosti:

-f	Špecifikuje súbor s URL adresami, z ktoré sa spracujú
-c	Špecifikuje certifikát, na overenie servera
-C	Špecifikuje zložku s certifikátmi na overovanie
-T	Na výstup vypíše dátum poslednej aktualizácie
-a	Na výstup vypíše meno autora záznamu
-u	Na výstup vypíše url z každého záznamu
-k	Špecifikuje offline súbor na spracovanie

Príklad použitia:

`./feedreader https://www.fit.vut.cz/fit/news-rss/`
Spracuje jednu stránku

`./feedreader -f urls.txt -aT`
Spracuje všetky url z daného súboru