



Dokumentácia projektu
Čtečka novinek ve formátu Atom a RSS s podporou TLS
Sít'ové aplikace a správa sítí

1. Úvod

Cieľom projektu bolo vytvoriť čítačku feedov vo formáte Atom a RSS 2.0 v programovacom jazyku C/C++. Pre implementáciu som si zvolil jazyk C++ s objektovým prístupom. Program parsuje vstupné argumenty popísané v priloženom README.md a následne s využitím knihovne OpenSSL sťahuje feedy z internetu, z ktorých vypíše údaje požadované užívateľom.

2. Návrh a implementácia

Program je rozdelený na 3 podčasti, z ktorých každá implementuje inú časť projektu. V tejto kapitole ich stručne popíšem.

2. 1. Trieda Arguments

V tejto triede sa nachádza viacero parametrov, ich setterov a getterov pre načítavanie a uchovávanie argumentov. Okrem toho sa tu nechádza funkcia main(), ktorá načítava argumenty a predáva ich metóde parseArguments. Ak užívateľ zadá nevalidný argument program vypíše na štandardný výstup chybovú hlášku a ukončí program s chybovým kódom. Potom prebieha dodatočná kontrola argumentov. Ak nebola zadaná URL program hľadá URL z argumentu “feedfile” a parsuje ich do vektoru. Následne prebieha kontrola portu. Ak URL má špecifikovaný port, parsuje sa a uloží do objektu. Ak URL nemá špecifikovaný port zvolí sa predvolený port pre HTTP alebo HTTPS protokol. Podobne to funguje aj pre možnosť, kedy sa berú URL zo súboru. Jediný rozdiel je v tom, že porty sa ukladajú do pola a teda platí, že k URL na indexe ‘i’ patrí port na rovnakom indexe. Maximálna dĺžka URL adresy je zvolená na 500 znakov. Používanie argumentov ‘-f feedfile’ a ‘feedURL’ súčasne nie je podporované.

2. 2. Trieda OpenSSL

Tu prebieha celá komunikácia pomocou OpenSSL knihovne. Knochovňa sa najprv inicializuje a potom cyklí cez vektor URL adries. URL adresa sa oreže a pridá sa na koniec port. Následne sa vytvára spojenie. Pri protokole HTTPS sa ešte navyše overuje platnosť certifikátu. Skontroluje sa, či sa spojenie vytvorilo správne a pošle sa HTTP požiadavok (HTTP request). Potom sa načíta HTTP odpoveď (HTTP response) a tá sa posiela Parseru.

2. 3. Trieda Parser

Táto trieda slúži na parsovanie HTTP odpovede a potrebných informácií z feedu. Metóda “parseHttpResponse” odstraňuje hlavičku z HTTP odpovede a vráti pointer na prvý znak feedu. Ak HTTP odpoveď vráti kód 301 (redirect), funkcia tento fakt zaloguje a pracuje s ďalšími URL. Metóda “parseXML” zistí o aký typ feedu sa jedná a následne z neho vypisuje potrebné údaje. Zadanie nešpecifikovalo poradie parametrov, takže som si zvolil aby atribút “title” bol vždy prvý a ostatné podľa ich poradia vo feede.

3. Návod na použitie

Stručný návod na použitie:

- Extrahovanie: `tar -xf xvalen27.tar`
- Preklad: `make` alebo `make feedreader`
- Spustenie: `./feedreader <URL | -f <feedfile>> [-c <certfile>] [-C <certaddr>] [-T] [-a] [-u]`

Pre podrobnejší návod na použitie a popis argumentov viz priložené README.md.

4. Záver

Naimplemenoval som všetko, čo zadanie vyžaduje a testovanie aplikácie som robil ručne a neobjavil som nejaké chybné chovanie programu.

Zadanie vyžadovalo striktný formát výstupu v češtine, takže som to robil podľa zadania. Výpis error hlášok je ale v angličtine, pretože využívam funkciu “`ERR_reason_error_string(ERR_get_error())`”. Tá vypisuje erro hlášky v angličtine a chcel som udržať konzistenciu.

Pri práci na projekte som sa naučil používať OpenSSL knihovňu a zdokonalil som sa v C++.