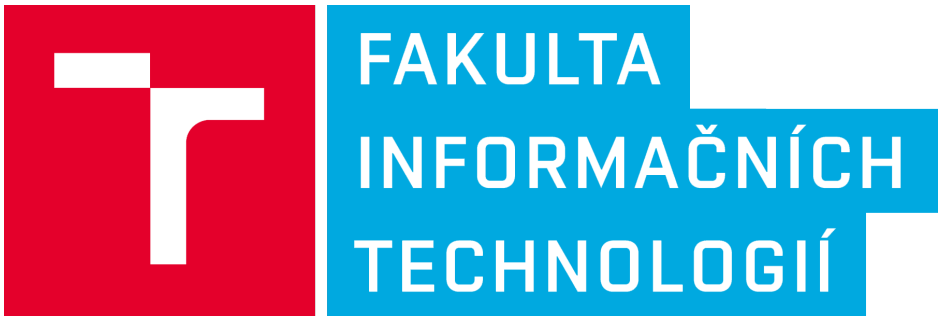
Vysoké učenie technické v Brne
Fakulta infromačních technologií




# Feedreader

Dokumentácia k projektu pre predmet ISA

17. listopadu 2018

**Ján Jakub Kubík, xkubik32**

# Obsah

# 1 Úvod

Tento dokument je dokumentácia k programu feedreader, ktorý bol zadaný ako projekt do predmetu ISA (sieťové aplikácie a správa sietí) na VUT FIT (Vyské učenie technické v Brne Fakulta informačných technológii) v akademicko roku 2018/19 v zimnom semestri.

Cieľom projektu je vytvoriť program feedreader, ktery bude vypisovat informácie uvedené v stiahnutých zdrojoch (feed) vo formáte Atom a RSS. Program po spustení stiahne zadané zdroje a na štandardný výstup vypíše informácie požadované uživateľom.

Dokument je členený na 7 kapitol. Základné pojmy a princípy, ktoré sú neskôr používané v texte a je na ne odkazované sú predmetom 2. kapitoly. Kapitola 3 obsahuje návrh programu feedreader a v kapitole 4 je tento návrh impelementovaný. Príklady spustenia programu spolu so screenshotmi sú obsiahnuté v kapitole 5. Priebeh testovania a automatizované testy sú opísané v kapitole 6. Dokumentácia je ukončená zhrnutím v 7. kapitole.

# 2 Dôležité pojmy

## 2.1 Webové feedy

Sú to dátové formáty sprostredkujúce uživateľom často sa meniaci obsah na webe. Najčastejšie používané formáty sú RSS1, RSS2 a Atom.

### RSS

RSS (Really Simple Syndication) je dialekt XML (eXtensible Markup Language). Všetky RSS súbory musia obsahovať XML 1.0 špecifikáciu. RSS má 2 hlavné verzie:

- RSS1 (viď. http://web.resource.org/rss/1.0/)
- RSS2 (viď. http://www.rssboard.org/rss-specification)

Oficiálne moduly pre RSS1 sú:

- Dublin Core
- Syndication
- Content

RSS2 a RSS1 je principálne to isté len každý z nich používa inú podmnožinu xml elementov. Jednotlivé elementy majú často aj rovnaký význam, len sú inak pomenované.

### Atom

Atom je taktiež založený na XML, ale na roziel od RSS je štandardizovaný v RFC4287.

## 2.2 SSL a TLS

SSL (Secure Socket Layers) a TLS (Transport Layer Security) sú kryptografické protokoly poskytujúce autentizáciu a šiforvanie dát medzi serverom, a klientskou aplikáciou komunikujúcou cez internet. SSL je predchodca TLS.

## 2.3 HTTP a HTTPS

HTTP (Hypertext Transfer Protocol) sedenie je sekvencia request-response transakcii. HTTP klient iniciuje request vytvorením TCP (Transmission Control Protocol) spojenia na špecifický port servera (typicky 80, zriedkavo 8080). HTTP server naslúcha na tomto porte a čaká na klientský request, ktorý následne spracujé a pošle klientovi response obsahujúci hlavičku s návratovým kódom (môže byť aj error) a telo odpovede.

Request metódy sú napr.: GET, POST atď.
Metóda GET sa používa v projekte.

HTTP je nezabezpečená komunikácia. HTTPS (Hypertext Transfer Protocol Secure) umožňuje zabezpečenú komunikáciu po sieti. Využívá protokol HTTP spolu s protokolom SSL alebo TLS. Pri HTTPS používa server typicky port číslo 443.

# 3 Návrh

Návrh je rozdelený do 4 spolu súvisiacich komplexnjekších častí pomenovaných: spracovanie argumentov, spracovanie feedfile, komunikácia so serverom, pársovanie feedov.

## 3.1 Spracovanie argumentov

Povolené argumenty programu a ich popis sú podrobne rozobrané v Readme.md. Argumenty sú parsované pomocou getopt. V prípade zadania nespravných argumentov je vypísaná nápoveda a program je ukončený. Ošetrené je taktiež duplicitné zadávanie argumentov -¿ program vypíše chybu, nápovedu k programu a ukončí sa. Nápoveda je vypísaná tiež po zadaní prepínača -h. Poradie agumentov je voliteľné a jednotlivé argumenty sa môžu spájať podľa unixových zvyklostí.
Napr.:

```
./feedreader http://www.fit.vutbr.cz/news/news-rss.php -T -u -a
./feedreader http://www.fit.vutbr.cz/news/news-rss.php -Tua
./feedreader -T -u -a http://www.fit.vutbr.cz/news/news-rss.php
./feedreader -Tu http://www.fit.vutbr.cz/news/news-rss.php -a
```

Všetky vyššie uvedené príklady spustenia programu s argumentami sú ekvivalentné.

## 3.2 Spracovanie feedfile

Feedfile je súbor z ktorého sa po zadaní príslušného prepínaču (-f feedfile) vyberú URL.
Obr. 1. zobrazuje všetky možné kombinácie slov vo feedfile. Uloženie alebo ignorovanie jednotlivých slov resp. URL popisujem pod obrázkom.

Obrázek 1: Vzorový príklad feedfile so všetkými možnými druhmi prvkov v súbore

```
1 # komentar
2             # komentar nie od zaciatku subora
3 http://www.fit.vutbr.cz/news/news-rss.php  # komentar za URL
4              http://www.fit.vutbr.cz/news/news-rss.php
5 http://www.fit.vutbr.cz/news/news-rss.php   NESPRAVNE_URL
6 http://www.fit.vutbr.cz/news/news-rss.php    http://www.fit.vutbr.cz/news/news-rss.php
7 NESPRAVNE_URL NESPRAVNE
```

Na riadku 1 je znázornemý komentár. Komentár začína znakom #. Komentár nemusí byť len ako prvý znak na riadku. Môže byť ako n-tý znak. Všetko za znakom komentáru je až do konca riadku považované za komentár (riadok 2). Komentár môže byť aj za URL na konci riadku (3. riadok).

Na 4. riadku je jedno URL, ktoré nemusí začinať ako prvý znak na riadku. V 5. riadku je príklad správného a nesprávneho URL. Ak sú za nesprávnym URL aj nejaké iné správne URL, tak tieto správne URL sa na aktuálnom riadku preskočia a pokračuje sa ďalším riadkom. Správne URL sa uloží a nesprávne sa ignoruje, vypíše sa chyba a spracovávanie feedfile pokračuje ďalej. V prípade, ak je na jednom riadku viac URL (riadok 6), tak sa spracujú všetky URL.

Ak je na riadku hocičo iné ako URL (riadok 7), tak sa to ignoruje a preskočí. Prázdne riadky sa taktiež preskakujú.

## 3.3 Komunikácia so serverom

Klient dostane ako parameter URL. URL si následne rozpársuje, pripojí sa na server a získa požadovaný obsah. Klient rozpársuje URL na server, hosta, port (ak bol zadaný) a požadovaný komunikačný protokol (HTTP alebo HTTPS). Pri HTTPS je použitá knihovna OpenSSL. Server musí byť zadaný ako ascii adresa servera. Host je dokument ktorý získava klient zo servera.

Pri komunikácii pomocou HTTP aj HTTPS sú využité BIO sockety. Explicitne sa klient pri použití HTTP pripája na port 80 a pri použití HTTPS na port 443. HTTP aj HTTPS používaju request metódu GET v tvare:

```
GET adressaServra HTTP/1.0
    Host: pozadovanyDokument
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
    Connection: Close
```

Klient nepožíva verziu HTTP 1.1 ale HTTP 1.0 v metóde GET. HTTP 1.0 používam preto, že pri používaní HTTP 1.1 requestov vracal občas server požadovaný obsah s veľkosťou chunkov a na tom následne padalo pársovanie feedov. Rersponse od servera je vždy v HTTP 1.1. Klient si overí návratový kód z hlavičky odpovede. Ak nie je návratový kod 200, tak vypíše hlavičku s chybovou hláškou. Pri správnom návratovom kóde uloží obsah tela odpovede tj. potenciálny feed.

## 3.4 Pársovanie feedov

Pársujú sa formáty RSS1, RSS2 a a Atom. Na pársovanie sa použiva knižnica libxml2. Jednotlivé formáty sa rozlišujú na základe prvého elementa. Podľa zadanie je treba získavať z feedov: názov zdroja. Ďalej sa získava titulok záznamu, a po zadaní voliteľných prepínačov sa získavajú pre každý jeden záznam autor, aktualizácia, URL.

V tabuľke sú uvedené formáty a k ním jednotlive elementy, z ktorých sa získava požadovaný obsah.

Obrázek 2: Prehľad, RSS1, RSS2 a Atom elementov, ktorých obsah sa priraďuje zdroju, titulku záznamu, aktualizácii, autorovi a URL

| | **RSS 1.0** | **RSS 2.0** | **ATOM 1.0** |
|---|---|---|---|
| **Názov zdroja (kanálu)** | rdf:RDF -> channel -> title | rss -> channel -> title | feed -> title |
| **Titulok záznamu** | rdf:RDF -> item -> title | rss -> channel -> item -> title | feed -> entry -> title |
| **Aktualizácia záznamu** | rdf:RDF -> item -> dc:date | rss -> channel -> item -> pubDate | feed -> entry -> published / updated |
| **Autor (meno a e-mail) záznamu** | rdf:RDF -> item -> dc:creator | rss -> channel -> item -> author | feed -> entry -> author -> name / email *(autorov môže byť viacej)* |
| **URL záznamu** | rdf:RDF -> item -> link | rss -> channel -> item -> link | feed -> entry -> link href="" |

Pri Atom formáte sa ukladajú všetci autori záznamu v tvare: autor (email), dalsi autor (email), ... Pri Atom formáte pre aktualizácia sa použije element published. Ak nie je element published ale je element updated tak ten sa použije pri aktualizácii.

Formát výpisu dát je pevne daný. Vypisuje sa po jednotlivých zdrojoch, ktoré su vždy oddelené 1 prázdnym riadkom. Ak sú zadané všetky voliteľne parametre tak sa vypisuje:

```
***názov zdroja***
titulok záznamu 1
Aktualizace: dátum
Autor: meno autora
URL: url

titulok záznamu 2
Aktualizace: dátum
Autor: meno autora
URL: url

....

titulok záznamu n
Aktualizace: dátum
Autor: meno autora
URL: url
```

Výpis aktualizácie, autora, URL je vždy v tomto poradí a nie je závislý na poradí zadávania argumentov programu. Ak nie je zadaný prepínač napr. pre autora tak sa ten riadok nevypíše. To isté platí aj pre URL a pre aktualiuáciu. Ak je zadaný prepínač napr. pre URL a obsah URL je prázndy, aj tak sa vypíše URL: . To isté platí aj pre autora a pre aktualzáciu.

Ak nie je zadaný ani jeden voliteľný parameter programu, tak sa vypisujú titulky v jednom zázname bez medzier:

```
***názov zdroja***
titulok zaznamu 1
titulok záznamu 2
....
titulok záznamu n
```

# 4 Implementácia

Ako implementačný jazyk som si zvolil C++ pre jeho jednoduchosť práce s reťazcami, pokročilú možnosť pársovania argumentov (getopt) a možnosť využitia OOP (Object Oriented Programming) princípov. Program vracia chybový návratový kód len pri zle zadaných argumentoch (99) a pri chybe pri otváraní feedfile (42). Vo všetkých zvyšných chybových stavoch program len vypíše chybu, ale neukončí sa a pokračuje ďalej.

## 4.1 Logické členienie programu

Každá sekcia z predchádzajúcej kapitoly predstavuje samostatnú triedu, ktorá vykonáva svoju špecifickú činnosť popísanú v predchádzajúcej kapitole.
Rozdelenie tried:

- trieda ArgumentParser slúži na spracovanie argumentov
- trieda Client slúži na komunikáciu so serverom
- trieda FeedFileParser slúži na spracovanie feedfile
- trieda FeedParser slúži na pársovanie feedov
- modul main je vstupný bod programu a obsahuje základnú logiku celého programu.

## 4.2 Popis činnosti programu v module main

Najskôr sa vytvorí objekt z triedy ArgumentParser, ktorý rozpársuje argumenty programu a uloží ich do svojej internej štruktúry.

Následne sa overí, či bol zadaný argument pre feedfile. Ak áno, tak sa vytvorí objekt z triedy FeedFileParser. Tento objekt ďalej spracuje feedfile a platné URL zapíše do do vektoru z modul main, ktorý je predaný konštruktoru daného objektu ako ukazateľ.

Ak nebol zadaný feedfile ako argument programu, tak sa do do vektoru v module main zapíše premenná z dátovej štruktúry v objekte vytvorného z triedy ArgumentParser. Táto premenná obsahuje URL zadané ako argument programu feedreader.

Vektor v main (všetky URL) sa precháza prvok po prvku pomocou cyklu. V každej iterácii cyklu sa vytvára nový objekt z triedy Client, ktorému sa pri konštrukcii predáva parameter aktuálne spracovávaného URL. Na základé URL sa pripojí k servru a získa požadovaný feed ktorý vracia do internej premennej vo tele cyklu v module main. Táto premenná je následné predaná ako parameter pri konštrukii objektu z triedy FeedParser. Tento objekt rozpársuje feed a vypíše na stdout.

# 5 Príklady spustenia

Screenshoty z jednotlivých spustení programu boli zachytené na mojom počítači s operačným systémom Ubuntu 18.04.

Obrázek 3: Spustenie programu bez volitelných argumentov
(tj. žiadna medzera medzi feedmi v kanáli)

```
jakub@Jakub13:~/Code/ISA/feedreader$ ./feedreader https://tools.ietf.org/dailydose/dailydose_atom.xml
***The Daily Dose of IETF***
The Daily Dose of IETF - Issue 3226 - 2018-11-05
The Daily Dose of IETF - Issue 3225 - 2018-11-02
The Daily Dose of IETF - Issue 3224 - 2018-11-01
The Daily Dose of IETF - Issue 3223 - 2018-10-31
The Daily Dose of IETF - Issue 3222 - 2018-10-30
The Daily Dose of IETF - Issue 3221 - 2018-10-29
The Daily Dose of IETF - Issue 3220 - 2018-10-26
The Daily Dose of IETF - Issue 3219 - 2018-10-25
The Daily Dose of IETF - Issue 3218 - 2018-10-24
The Daily Dose of IETF - Issue 3217 - 2018-10-23
The Daily Dose of IETF - Issue 3216 - 2018-10-22
The Daily Dose of IETF - Issue 3215 - 2018-10-19
The Daily Dose of IETF - Issue 3214 - 2018-10-18
The Daily Dose of IETF - Issue 3213 - 2018-10-17
The Daily Dose of IETF - Issue 3212 - 2018-10-16
The Daily Dose of IETF - Issue 3211 - 2018-10-15
The Daily Dose of IETF - Issue 3210 - 2018-10-12
The Daily Dose of IETF - Issue 3209 - 2018-10-11
The Daily Dose of IETF - Issue 3208 - 2018-10-10
The Daily Dose of IETF - Issue 3207 - 2018-10-09
```

Obrázek 4: Spustenie programu s volitelnými argumentami
(tj. jedna medzera medzi feedmi v kanáli)

```
jakub@Jakub13:~/Code/ISA/feedreader$ ./feedreader https://tools.ietf.org/dailydose/dailydose_atom.xml -Tua
***The Daily Dose of IETF***
The Daily Dose of IETF - Issue 3226 - 2018-11-05
Aktualizace: 2018-11-05T05:00:26Z
Autor:
URL: https://tools.ietf.org/dailydose/3226.html

The Daily Dose of IETF - Issue 3225 - 2018-11-02
Aktualizace: 2018-11-02T05:00:17Z
Autor:
URL: https://tools.ietf.org/dailydose/3225.html

The Daily Dose of IETF - Issue 3224 - 2018-11-01
Aktualizace: 2018-11-01T05:00:14Z
Autor:
URL: https://tools.ietf.org/dailydose/3224.html

The Daily Dose of IETF - Issue 3223 - 2018-10-31
Aktualizace: 2018-10-31T05:00:14Z
Autor:
URL: https://tools.ietf.org/dailydose/3223.html

The Daily Dose of IETF - Issue 3222 - 2018-10-30
Aktualizace: 2018-10-30T05:00:13Z
Autor:
URL: https://tools.ietf.org/dailydose/3222.html

The Daily Dose of IETF - Issue 3221 - 2018-10-29
Aktualizace: 2018-10-29T05:00:14Z
Autor:
URL: https://tools.ietf.org/dailydose/3221.html

The Daily Dose of IETF - Issue 3220 - 2018-10-26
Aktualizace: 2018-10-26T05:00:13Z
Autor:
URL: https://tools.ietf.org/dailydose/3220.html

The Daily Dose of IETF - Issue 3219 - 2018-10-25
Aktualizace: 2018-10-25T05:00:13Z
Autor:
URL: https://tools.ietf.org/dailydose/3219.html

The Daily Dose of IETF - Issue 3218 - 2018-10-24
Aktualizace: 2018-10-24T05:00:15Z
Autor:
URL: https://tools.ietf.org/dailydose/3218.html

The Daily Dose of IETF - Issue 3217 - 2018-10-23
Aktualizace: 2018-10-23T05:00:20Z
Autor:
URL: https://tools.ietf.org/dailydose/3217.html
```

# 6 Testovanie

Program som testoval u seba na Ubuntu 18.04 a na školských serveroch eva a merlin.

## 6.1 Manuálne testovanie

Prevažná časť tesovania bola vykonávaná manuálne.
Konkrétne som testoval:

- správnosť spracovania argumentov
- správnosť spracovanie feedfile
- používanie HTTPS s lokálnym certifikátom alebo adresárom
- sťahovanie jednotlivých feedov a ich výpis v požadovanom formáte

## 6.2 Poloutomatizované tesovanie

Vytvoril som menšiu sadu poloautomatizovaných testov zameraných na správnosť spracovania jednotlivých feed formátov. Pre spustenie testov je potrebné mať nainštalovaný interpret pythonu aspoň vo verzii 3.5. Automatizované testy sa spúšťajú pomocou make test.

Najskôr je treba spustiť v terminály jednoduchý http server pomocou pytho3 tests/server.py. Make test spustí program na testovanie (test.py). Jednoduchý HTTP server beži na localhoste na porte 5300. Testovací program (test.py) spúšťa opakovane program feedreader, ktorý posiela requesty na lokálny HTTP server. Jednoduchý HTTP server odpovedá na request metódu:

```
GET adressaServera HTTP/1.0
    Host: pozadovanyDokument
    Connection: Close
```

pričom adresaServra je localhost a pozadovanyDokument je požadovaný feed. Odpoveď servera obsahuje hlavičku (návratový kód) a telo. Telo obsahuje požadovaný feed.

Výstup z feedreaderu na stdout sa porovná v test.py s preddefinovanými správne naformátovanými feedmi a na stdout sa vypíše, či test prešiel alebo nie. Na záver sa vypíše celkové zhodnotenie úspešnosť testov. Na úplny záver make test pomocou pkill -9 python3 ukončí bežiaci http server.

# 7 Záver

Zo zadania sú splnené všetky podmienky, len getopt nefunguje správne pri spúštaní programu na referenčnom stroji eva.fit.vutbr.cz.
Ako rozšírenia by sa ešte mohlo dorobiť ("keby mám viac času"):

- timer pri prpájaní na server (pretože, ak sa pripája na zatvorený port tak tak aplikácia uviazne v deadlocku)
- štrukturálny, behaviorálny, tvorivý návrhový vzor alebo ich kombináciu pre kvalitnejší program
- viac a kvalitnejšie automatizované testy

# 8 Referencie


1. Zadanie, [cit. 10-11-2018], URL: https://wis.fit.vutbr.cz/FIT/st/course-sl.php?id=666821&item=70193&cpa=1

2. RFC 4287, [cit. 10-11-2018], URL: https://tools.ietf.org/html/rfc4287

3. RSS1, [cit. 10-11-2018], URL: http://web.resource.org/rss/1.0/

4. RSS2, [cit. 10-11-2018], URL: http://www.rssboard.org/rss-specification

5. Použitie openssl, [cit. 10-11-2018], URL: https://developer.ibm.com/tutorials/l-openssl/

6. C++, [cit. 10-11-2018], URL: https://en.cppreference.com/w/

7. SSL vs TLS, [cit. 10-11-2018], URL: https://www.globalsign.com/en/blog/ssl-vs-tls-difference/

8. python server na testovanie, [cit. 10-11-2018], URL: https://gist.github.com/bradmontgomery/2219997