VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV -
DEPARTMENT OF -

LDAP SERVER
LDAP SERVER

PROJEKOTVÁ DOKUMENTACE
PROJECT DOCUMENTATION

AUTOR PRÁCE
AUTHOR

ONDREJ KURÁK

BRNO 2017

Obsah

1 Úvod **2**
1.1 Cieľ projektu . . . 2
1.2 Popis . . . 2

2 Úvod do problematiky **3**
2.1 LDAP správy . . . 3
2.1.1 Bind Request . . . 3
2.1.2 Bind Response . . . 3
2.1.3 Search Request . . . 3
2.1.4 Search Result Entry . . . 3
2.1.5 Search Result Done . . . 4
2.1.6 Unbind Request . . . 4
2.2 LDAP filtre . . . 4
2.3 Rozšírenia . . . 4

3 Implementácia **5**
3.1 Server . . . 5
3.2 LDAP parser . . . 5
3.2.1 Čítanie/overovanie správ . . . 5
3.2.2 Vytváranie/odosielanie správ . . . 5
3.2.3 Spracovanie filtrov . . . 6
3.2.4 Spracovanie ASN.1 . . . 6
3.2.5 Pomocné triedy . . . 6

4 Testovanie **7**

5 Použitie **8**
5.1 Kompilácia . . . 8
5.2 Spustenie programu . . . 8

Literatúra **9**

Kapitola 1

Úvod

1.1 Cieľ projektu

Cieľom tohto projektu je implementácia jednoduchého konkurentného LDAP serveru (LDAPv2), ktorý má zvládať spracovávať správy typu Bind Request, Bind Response, Search Request, Search Response Entry, Search Response Done a Unbind Request. Server má podporovať filtre typu And, Or, Not, Equality Match a Substring.

1.2 Popis

Server je implementovaný v C++ ako konzolová aplikácia, ktorá dostáva ako argumenty port na ktorom načúva a databázu s ktorou pracuje.

Kapitola 2

Úvod do problematiky

Lightweight Directory Access Protocol (LDAP) je internetový protokol pre prístup k adresárovým službám. Podľa tohto protokolu sú jednotlivé položky ukladané formou záznamov ukladané formou stromovej štruktúry. Protokol bol navrhnutý ako jednoduchšia alternatíva k X.500 [6].

2.1 LDAP správy

Pre správy je používané kódovanie ASN.1 [3]. V správach pri menách sa nerozlišuje medzi veľkými a malými písmenami (case insensitive) [1]. Poradie prímania správ nie je fixné (nie je potrebné aby prvá správa bola Bind Request) [2]. Pri našej implementácii podporujeme správy typu Bind Request, Bind Response, Search Request, Search Result Entry, Search Result Done a Unbind Request.

2.1.1 Bind Request

Správa zahajujúca kominukáciu klienta so serverom. V implentácii LDAPv2 neobsahuje žiaden druh overenia. Na správu server odpovedá správou Bind Response

2.1.2 Bind Response

Správa od servera potvrdzuje klientovy, že je server pripravený s ním komunikovať.

2.1.3 Search Request

Správa od klienta s požiadavkou na vyhľadávanie v databáze servera. Na správu server odpovedá podľa výsledku správami Search Result Entry a Search Result Done.

2.1.4 Search Result Entry

Správa od servera s čiastkovím výsledkom vyhľadávania v databáze podľa požiadavky od klienta. Správa sa posiela pre každý nájdený záznam. Pokiaľ žiaden záznam neodpovedá požiadavke klienta správa sa neodosiela. Pri našej implementácii sa odošlú údaje cn, uid a mail.

2.1.5 Search Result Done

Správa od servera ukončujúca vyhľadávanie. Odosiela sa aj pokiaľ nebol žiaden výsledok.

2.1.6 Unbind Request

Správa od klienta ukončujúca spojenie so serverom. Server neodpovedá a ukončí spojenie s klientom.

2.2 LDAP filtre

Správa Search Request obsahuje filtre, podľa ktorých sa majú filtrovať výsledky pre vyhľadávanie. Filtre je možné ľubovolne kombinovať. V našej implenetácii podporujeme filtre typu And, Or, Not, Substring a Equality Match.

2.3 Rozšírenia

Program má podporu utf-8 národných znakov.

Kapitola 3

Implementácia

Server je implementovaný v jazyku C++ ako konzolová aplikácia s parametramy portu a súboru s databázou. Ako vzor pre implentáciu boli použité RFC4511[3], Wireshark a stránka Apache[5].

3.1 Server

Server podporuje iba protokol IPv4. Požiadavky sú spracovávané paralélne za pomoci vlákien (threads). Zdrojové súbory: `server.h server.cc`

3.2 LDAP parser

Čítanie správ LDAP je implentované ako konečný automat. Celý automat je implementovaný v triede `LDAP_parser()`. Trieda obsahuje niekoľko kategórii metód:

3.2.1 Čítanie/overovanie správ

- `start()` - metoda prejde spoločný základ pre všetky LDAP správy a následne rozhodne o ktorú správu sa jedná
- `bind_req()` - spracovanie Bind Request a odoslanie Bind Response
- `search_req()` - spracovanie Search Request, spracovanie a aplikácia filtrov, odoslanie Search Result Entry a Search Result Done
- `unbind_req()` - spracovanie Unbind Request a ukončenie spojenia

zdrojové súbory: `ldap_fsm.h ldap_fsm.cc`

3.2.2 Vytváranie/odosielanie správ

- `bind_response()` - vytvorenie a odoslanie správy Bind Response
- `search_res_entry()` - vytvorenie a odoslanie správy Search Result Entry pre každý výsledok aplikácie filtrov
- `search_res_done()` - vytvorenie a odoslanie správy Result Done

zdrojové súbory: `ldap_fsm.h ldap_fsm.cc`

3.2.3 Spracovanie filtrov

- `get_filter()` - rekurzívne spracovanie filtrov a ich uloženie do stromovej štruktúry
- `print_filters()` - vypísanie celej stromovej štruktúry filtrov
- `resolve_filters()` - rekurzívne aplikovanie všetkých filtrov na data z databázového súboru

zdrojové súbory: `ldap_fsm.h filters.cc`

3.2.4 Spracovanie ASN.1

- `get_ll()` - získanie dĺžky následujúceho reťazca na základe aktuálneho znaku
- `get_int()` - získanie `int` podľa aktuálneho znaku
- `get_string()` - získanie reťazca podľa aktuálneho znaku
- `cn()` - pomocná metóda na premenu čísla na reťazec o jednom znaku, využívaná pri spájaní reťazcov
- `make_ll()` - zo zadaného reťazca vytvorí retažec s predponou jeho dĺžky
- `make_int()` - zo zadaného `int` vytvorí reťac, ktorý ho obsahuje v reťazcovej forme

Všetky metódy sú pripravené na spracovanie ASN.1 [4].
zdrojové súbory: `ldap_fsm.h ber_functions.cc`

3.2.5 Pomocné triedy

- `Filter()` - trieda pre ukladanie LDAP filtrov do stromovej štruktúry
- `Message()` - trieda pre ukladanie podstatných informácii o LDAP správach

Kapitola 4

Testovanie

Projekt bol testovaný za pomoci nástrojov `wireshark` (kontrola či správy zodpovedajú LDAP protokolu) a `ldapsearch` (kotrola výsledku filtrov). Ako testovacia databáza bol použitý súbor v kódovaní `utf-8` s národnými znakmy (CZ, SK). Program bol testovaný na systémoch CentOS Linux release 7.4.1708 (server merlin) a Ubuntu 17.07.

Kapitola 5

Použitie

5.1 Kompilácia

Kompilácia prebieha za pomoci príkazu make.

```
~> make
```

Makelife má pripravené pramatre:

- myldap - kompilácia programu, predvolené nastavenie
- debug - kompilácia programu s debugovacím prepínáčom
- clean - odstránenie binárneho súboru myldap
- tar - zbalenie všetkých potrebných súborv pre odoslanie

5.2 Spustenie programu

```
~> ./myldap {-p <port>} -f <name>
```

Parametre:

- -p <port> - port na ktorom má server počúvať, predvolený je 389. Voliteľný argument
- -f <name> - cesta k súboru s databázou

Príklad spustenia:

```
~> ./myldap -p 1234 -f mydb.csv
```

Server sa ukončuje príkazom `Control + C`.

Literatúra

[1] Boswell, W.: *Understanding Active Directory Services*. Říjen 2003, [Online; navštívené 12.11.2017].
URL http://www.informit.com/articles/article.aspx?p=101405&seqNum=7

[2] Carpenter, B.: *Architectural Principles of the Internet*. 1996, [Online; navštívené 12.11.2017].
URL https://www.ietf.org/rfc/rfc1958.txt

[3] J. Sermersheim, E. N.: *Lightweight Directory Access Protocol (LDAP): The Protocol*. 2006, [Online; navštívené 12.11.2017].
URL https://tools.ietf.org/html/rfc4511

[4] Jr., B. S. K.: *A Layman's Guide to a Subset of ASN.1, BER, and DER*. Listopad 1993, [Online; navštívené 12.11.2017].
URL http://luca.ntop.org/Teaching/Appunti/asn1.html

[5] Lécharny, E.: *Ldap ASN.1 Codec*. Říjen 2006, [Online; navštívené 12.11.2017].
URL
https://cwiki.apache.org/confluence/display/DIRxSRVx10/Ldap+ASN.1+Codec

[6] Matoušek, P.: *Síťové aplikace a jejich architektura*. Akademické nakladatelství, VUTIUM, 2014, ISBN 9788021437661.