VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí

Programování síťové služby LDAP server

20. listopadu 2017 Jakub Frýz

Obsah

1 Úvod

1.1 Zadání

Úkolem bylo vytvořit jednoduchý LDAP server, který bude načítat dotazy od LDAP klientů a vyhledávat odpovědi v lokální textové databázi.

Server nemusí podporovat češtinu či jiné národní znaky. Server nevyžaduje autentizaci, stačí podpora LDAPv2, který řeší pouze vyhledávání. Ostatní zprávy server ignoruje, popřípadě odpoví, že danému příkazu nerozumí.

Pracovní databáze serveru má formát textového souboru CSV (comma separated values se středníkem jako oddělovačem), který obsahuje tři položky - Jméno (`cn`, `CommonName`), Login (`uid`, `UserID`), Email (`mail`).

Co se týče TLV filter, tak nad pracovní databází připadá v úvahu podpora `and`, `or`, `not`, `equalityMatch` a `substrings`.[1]

[1] Pozn.: zadání bylo převzato a upraveno z WIS

2 Popis

V následujících podkapitolách se budu snažit přiblížit stručně protokol LDAP.

2.1 Protokol LDAP

LDAP (*Lightweight Directory Access Protocol*) je internetový protokol pro přístup k adresářovým službám. Je odlehčenou verzí předchozího protokolu `X.500`.

Systém LDAP je definován množinou čtyř modelů: Informační, jmenný, funkční a bezpečnostní model.

- **Informační model**
  - popisuje uložení informací v adresáři
    - záznamy = souhrn atributů (dvojice název-hodnota)
  - definuje typy dat a operace nad nimi (porovnání,. . .)
- **Jmenný model**
  - popisuje strukturu adresáře
  - tvoří stromovou strukturu DIT (Directory Information Tree)
- **Funkční model**
  - zabývá se přístupem k datům a operacemi nad nimi (vyhledávání,. . .)
- **Bezpečnostní model**
  - zabývá se zabezpečením dat
  - definuje práva pro vkládání a modifikaci dat adresáře
  - stará se o autentizaci uživatele (SASL)

Protokol LDAP se používá například pro vyhledávání e-mailové adresy u poštovního klienta nebo pro autentizaci uživatele.

2.2 Standard ASN.1

ASN.1 (*Abstract Syntax Notation One*) je prostředek pro popis datových struktur pro reprezentaci, kódování, přenos, ukládání a dekódování dat v telekomunikacích, počítačových sítích a informatice. Poskytuje soubor pravidel umožňujících popsat strukturu objektů způsobem nezávislým na konkrétním hardwarovém řešení. [7]

```
LDAPMessage ::= SEQUENCE {
    messageID MessageID,
    protocolOp CHOICE {

        bindRequest       BindRequest,
        bindResponse      BindResponse,
        unbindRequest     UnbindRequest,
        searchRequest     SearchRequest,
        searchResEntry    SearchResultEntry,
        searchResDone     SearchResultDone,
        searchResRef      SearchResultReference,
        modifyRequest     ModifyRequest,
        modifyResponse    ModifyResponse,
        addRequest        AddRequest,
        addResponse       AddResponse,
        delRequest        DelRequest,
        delResponse       DelResponse,
        modDNRequest      ModifyDNRequest,
        modDNResponse     ModifyDNResponse,
        compareRequest    CompareRequest,
        compareResponse   CompareResponse,
        abandonRequest    AbandonRequest,
        extendedReq       ExtendedRequest,
        extendedResp      ExtendedResponse },

    controls [0] Controls OPTIONAL }
```

Výše je ukázka z LDAP ASN.1 gramatiky [2].

2.3 Kódování BER

BER (*Basic Encoding Rules*) se používá pro kodóvání `ASN.1`. BER patří k TLV formátům (TLV = type-length-value = typ-délka-hodnota).

Pole `type` určuje význam hodnoty. Má následující strukturu:

8	7	6	5	4	3	2	1
Třída		P/C	Číslo tagu				

Třída určuje, kde je typ nadefinován (typy třídy `00` jsou nadefinované přímo standardem ASN.1). P/C rozlišuje, zda-li je hodnota jednoduchá nebo složená (obsahuje jiné položky). A zbytek je číslo tagu, který určuje konkrétní tag (2 = `integer`, 10 = `enumerated`, 16 = `sequence`).

Např.: Sekvence oktetů `02 01 01` znamená: integer délky 1 a hodnoty 1

3 Implementace

Aplikace byla napsána v jazyce C++. Část komunikace byla vytvořena pomocí projektu z minulého roku z IPK.

Při tvorbě mi hodně pomohla stránka cppreference.com.

3.1 Signál SIGINT

Pomocí `sigaction` odchytávám `SIGINT` signál. Ten teď namísto toho, aby natvrdo ukončí aplikaci, ukončí smyčku pro nová připojení, ukončí sokety a uzavře soubor.

3.2 Komunikace

K vytvoření komunikace používám funkce `socket()`, `bind()`, `listen()` a `accept()` v tomto pořadí. Funkce `accept()` slouží k připojení klienta, a proto je použita cyklu.

3.3 Neblokující soket

Tohoto jsem dosáhl pomocí funkce `select()`, kterou jsem použil i minulý rok v projektu do IPK.

Jedinou nevýhodou mého řešení je zvýšená zátěž na procesor.

3.4 Procesy

Po připojení klienta k serveru se pomocí fce `fork()` vytvoří tzv. 'dětský proces', který zdědí informace z paměťového prostoru 'rodičovského procesu' a následně si vytvoří svůj vlastní paměťový prostor. Díky tomu neinterferuje s jinými procesy a přesto přístup k databázi.

3.5 LDAP zprávy

Následně se musí dekódovat požadavky z klienta.

Moje funkce se pouze podívá na určitá místa požadavku a přečte si jejich hodnotu (přesněji řečeno dvě místa – identifikátor zprávy a typ zprávy). Následuje zakódování odpovědi. Celou zprávu mám natvrdo zadanou, jen do ní zapíšu identifikátor zprávy. Toto platí pro požadavky typu `BindRequest` a `UnbindRequest` a odpovědi typu `BindResponse` a `SearchResultDone`.

Pro požadavky typu `SearchRequest` mám vytvořen stavový automat, který vyčte všechny potřebné informace z různě dlouhých zpráv.

A pro odpovědi typu `SearchResultEntry` mám funkce pro vytvoření celé zprávy podle toho, co bylo nalezeno v databázi.

3.6 Vyhledávání v DB

Databáze ve formátu CSV se načítá do proměnné typu `ifstream`. Tento typ je pouze pro čtení.

Databázi procházím po jednom od začátku. Využívám k tomu funkci `getline()` pro načtění řádku databáze a funkci `strtok()` pro rozdělení atributů podle znaků ';'.

Pokud byl `SearchRequest` typu `equalityMatch`, používám fce `strcmp`, a pro `SearchRequest` typu `substrings` používám metodu `find()`, kterou má typ `string`.

4 Testování

Testování převážně probíhalo na virtuálním stroji na mém počítači. Na tomto virtuálním stroji jsem měl nainstalovaný OS Kubuntu 17.10.

Pro účely testování jsem si vytvořil vlastní databázi ve formátu CSV, která je stejná jako ukázková ve WIS, ale neobsahuje diakritická znaménka. Tato databáze je přibalena v odevzdaném archívu pod názvem `db.csv`. Symbol '@' jsem v e-mailových adresách ponechal, protože se mi na virtuálním stroji zobrazují správně, ale na serveru `eva` se mi e-mailové adresy zobrazují jako shluk symbolů. Za to může pravděpodobně jiné kódování řetězce ze strany klienta.

Značná část pomocných výstupu použitých k testování byla z odevzdané verze vymazána.

CPU	Intel i5-6600K 3.3GHz
GPU	MSI R9 390X GAMING 8G 8GB
RAM	Kingston HyperX Fury Black 16GB DDR4 2133MHz
DISK #01	Samsuing SD 850 EVO 250GB
DISK #02	WD Green EZRX 2TB (5400rpm)
DISK #03	Samsuing SD 850 EVO 500GB
OS	Windows 10 Education 1703

Tabulka 1: HW konfigurace testovacího PC

Virtuální stroj běžel na disku č. 3.

Testování proběhlo úspěšně i na serveru `merlin`, ke kterému jsem se připojoval ze serveru `eva`.

5 Omezení

- filtry `and` a `or` nejsou podporovány
- při `SearchResultEntry` nejsou ošetřeny některé chyby (aplikace v tom případě odešle klientu `SearchResultDone` s návratovou hodnotou `0` a žádný `SearchResultEntry`)
- ignoruje otázky na název objektu
- pokud je TLV filtr `SearchRequestu` typu `present`, server vypíše prvních `SIZE_LIMIT` záznamů (u mne to funguje, na `evě` klient vrátí chybu)

6 Literatura

[1] MATOUŠEK, P. *Síťové aplikace a jejich architektura*. 1. vyd. [b.m.]: Nakladatelství Vysokého učení technického v Brně VUTIUM, 2014. 396 s. ISBN 978-80-214-3766-1.

[2] *Lightweight Directory Access Protocol (v3)*. prosinec 1997. Dostupné na: <https://www.ietf.org/rfc/rfc2251.txt>.

[3] *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*. prosinec 1997. Dostupné na: <https://www.ietf.org/rfc/rfc2252.txt>.

[4] *The String Representation of LDAP Search Filters*. prosinec 1997. Dostupné na: <https://www.ietf.org/rfc/rfc2254.txt>.

[5] *Lightweight Directory Access Protocol (LDAP): The Protocol*. červen 2006. Dostupné na: <https://www.ietf.org/rfc/rfc4511.txt>.

[6] *Lightweight Directory Access Protocol*. listopad 2017. Dostupné na: <https://cs.wikipedia.org/wiki/LDAP>.

[7] *Abstract Syntax Notation One*. duben 2013. Dostupné na: <https://www.ietf.org/rfc/rfc4511.txt>.

[8] *Basic Encoding Rules*. březen 2015. Dostupné na: <https://www.ietf.org/rfc/rfc4511.txt>.