ISA PROJECT 1(LDAP SERVER)

Author

- Miroslav Bálek (xbalek02)

Content table

1. Brief
2. TCP
3. TCP-Server
4. LDAP
5. LDAP-Server
6. Tests
7. Known Limitations
8. Refernces

1. Brief

The project implements a simplified server for the LDAP protocol. The program establishes the connection and communicates with the cient in the way specified for this protocol. Server is running at specified port listening to all ip addresses on both IPv4 and IPv6. Client then sends ldap search and the server responds with ldap response, containing requested information. Server searches simple semicolon separated csv for requested information.

2. TCP

TCP (Transmission Control Protocol) is a reliable, connection-oriented communication protocol used in computer networks. It provides ordered and authenticated data delivery between devices. TCP establishes a connection using a three-way handshake, where it sets parameters for communication. It provides flow control to prevent data congestion and uses delivery acknowledgement mechanisms to confirm the successful transmission of data. TCP is used for applications that require secure and accurate data transfer, such as browsers and file transfer.

3. TCP-Server

TCP communication first creates a socket. Then created socket is binded to specific ip address and port. In this case the socket is binded to in6addr_any (::) and port specified at launch of server (default port is 389). Server then listens on this socket for any incoming connections. When connection request is recieved, server accepts this connection at new socket and creates new process for this connection. After comunication is done the server closes this socket.

4. LDAP

LDAP is protocol that allows to store and access data about users, device, etc. Data are stored using tree like structure directory server. LDAP relies on TCP for efficient communication. LDAP uses Basic Encoding Rules (BER) specified in ASN.1.

5. LDAP-Server

This server acts as it is on domain fit.vutbr.cz.

If the server receives a non-LDAP message, it sends a notice of disconection to the client, otherwise the query is processed based on the tag value of the requested operation. If the operation is not recognized or supported, a notice of disconection message is sent to the client. If a bindrequest is received, the server responds with an appropriate bindresponse message. The searchRequest is responded to by the searchresentry server. If the server receives a searchrequest with an unsupported filter it responds with a searchresentry with error code 53 (Server is unwilling to perform). Upon receiving an unbindrequest message, the connection is terminated.

6. Tests

Test were performed on Ubuntu 22.04.3

Tests were performed by comparing the expected output with the actual one using ldapsearch.

Server was started using.

```
./isa-ldapserver -f lidi.csv -p 12345
```

Equality filter

This test performs search using equality filter. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(uid=xbalek02)"
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=xbalek02)
# requesting: ALL
#

#
dn: xbalek02,dc=fit,dc=vutbr,dc=cz
cn: Balek Miroslav
mail: xbalek02@stud.fit.vutbr.cz

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Substring filter (postfix)

This test performs search using postfix substring filter. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(uid=*ek02)"
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=*ek02)
# requesting: ALL
#

#
dn: xbabek02,dc=fit,dc=vutbr,dc=cz
cn: Babek Radomir
mail: xbabek02@stud.fit.vutbr.cz

#
dn: xbalek02,dc=fit,dc=vutbr,dc=cz
cn: Balek Miroslav
mail: xbalek02@stud.fit.vutbr.cz

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Substring filter (prefix)

This test performs search using prefix substring filter. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(uid=xbalu*)"
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=xbalu*)
# requesting: ALL
#

#
dn: xbaluc00,dc=fit,dc=vutbr,dc=cz
cn: Baluch Matus
mail: xbaluc00@stud.fit.vutbr.cz

#
dn: xbalus03,dc=fit,dc=vutbr,dc=cz
cn: Balusek Pavel
mail: xbalus03@stud.fit.vutbr.cz
```

```
# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Substring filter (infix)

This test performs search using infix substring filter. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(uid=*bal*)"
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=*bal*)
# requesting: ALL
#

#
dn: xbalar00,dc=fit,dc=vutbr,dc=cz
cn: Balarin Jakub
mail: xbalar00@stud.fit.vutbr.cz

#
dn: xbalek02,dc=fit,dc=vutbr,dc=cz
cn: Balek Miroslav
mail: xbalek02@stud.fit.vutbr.cz

#
dn: xbaluc00,dc=fit,dc=vutbr,dc=cz
cn: Baluch Matus
mail: xbaluc00@stud.fit.vutbr.cz

#
dn: xbalus03,dc=fit,dc=vutbr,dc=cz
cn: Balusek Pavel
mail: xbalus03@stud.fit.vutbr.cz

#
dn: xkubal12,dc=fit,dc=vutbr,dc=cz
cn: Kubala Jan
mail: xkubal12@stud.fit.vutbr.cz

#
dn: xkubal05,dc=fit,dc=vutbr,dc=cz
cn: Kubalik Jakub
mail: xkubal05@stud.fit.vutbr.cz
```

```
#
dn: xzobal01,dc=fit,dc=vutbr,dc=cz
cn: Zobal Ondrej
mail: xzobal01@stud.fit.vutbr.cz

# search result
search: 2
result: 0 Success

# numResponses: 8
# numEntries: 7
```

Substring filter (any substring in middle)

This test performs search using any substring filter in middle of searched value. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(uid=xba*k02)"
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=xba*k02)
# requesting: ALL
#

#
dn: xbabek02,dc=fit,dc=vutbr,dc=cz
cn: Babek Radomir
mail: xbabek02@stud.fit.vutbr.cz

#
dn: xbalek02,dc=fit,dc=vutbr,dc=cz
cn: Balek Miroslav
mail: xbalek02@stud.fit.vutbr.cz

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Size limit (Size limit not exceeded)

This test performs search using prefix substring filter with size limit of 4 that is exceeded. The result is the same as expected output.

```
 ldapsearch -x -H ldap://localhost:12345 "(uid=xbal*)" -z 4
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=xbal*)
# requesting: ALL
#

#
dn: xbalar00,dc=fit,dc=vutbr,dc=cz
cn: Balarin Jakub
mail: xbalar00@stud.fit.vutbr.cz

#
dn: xbalek02,dc=fit,dc=vutbr,dc=cz
cn: Balek Miroslav
mail: xbalek02@stud.fit.vutbr.cz

#
dn: xbaluc00,dc=fit,dc=vutbr,dc=cz
cn: Baluch Matus
mail: xbaluc00@stud.fit.vutbr.cz

#
dn: xbalus03,dc=fit,dc=vutbr,dc=cz
cn: Balusek Pavel
mail: xbalus03@stud.fit.vutbr.cz

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 4
```

Size limit (Size limit exceeded)

This test performs search using prefix substring filter with size limit of 1 that is exceeded. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(uid=xba*)" -z 1
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (uid=xba*)
# requesting: ALL
#
```

```
#
dn: xbabek02,dc=fit,dc=vutbr,dc=cz
cn: Babek Radomir
mail: xbabek02@stud.fit.vutbr.cz

# search result
search: 2
result: 4 Size limit exceeded
text: Size limit exceeded.

# numResponses: 2
# numEntries: 1
```

Size limit (No size limit)

This test performs search using prefix substring filter with size limit of 0 that is represented as no-limit. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "uid=xbal*" -z 0
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: uid=xbal*
# requesting: ALL
#

#
dn: xbalar00,dc=fit,dc=vut,dc=cz
cn: Balarin Jakub
mail: xbalar00@stud.fit.vutbr.cz

#
dn: xbalek02,dc=fit,dc=vut,dc=cz
cn: Balek Miroslav
mail: xbalek02@stud.fit.vutbr.cz

#
dn: xbaluc00,dc=fit,dc=vut,dc=cz
cn: Baluch Matus
mail: xbaluc00@stud.fit.vutbr.cz

#
dn: xbalus03,dc=fit,dc=vut,dc=cz
cn: Balusek Pavel
mail: xbalus03@stud.fit.vutbr.cz

# search result
search: 2
result: 0 Success
```

```
# numResponses: 5
# numEntries: 4
```

Usage of unsuported filter

This test performs search using unsuported filter. The result is the same as expected output.

```
ldapsearch -x -H ldap://localhost:12345 "(!(uid=xbalek))"
# extended LDIF
#
# LDAPv3
# base <> (default) with scope subtree
# filter: (!(uid=xbalek))
# requesting: ALL
#

# search result
search: 2
result: 53 Server is unwilling to perform
text: Usage of unsupported filter.

# numResponses: 1
```

7. Known Limitations

Server supports only equality match filters and substring filters. If a substring filter with multiple * is used the server behaves as if it received a prefix filter and ignores the rest of upcoming filters. Since correct functionality of this is not implemented it has not been tested.

8. References

RFC 4511 - Lightweight Directory Access Protocol (LDAP): The Protocol. Online. 2006. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4511. [cit. 2023-11-15].

LDAP.com – Lightweight Directory Access Protocol [online]. 2006 [cit. 2023-11-15]. Dostupné z: https://ldap.com/

ASN.1 Basic Encoding Rules. OssNokalva [online]. c2023 [cit. 2023-11-17]. Dostupné z: https://www.oss.com/asn1/resources/asn1-made-simple/asn1-quick-reference/basic-encoding-rules.html