Adresářové struktury

(Directory structures, DAP, LDAP, LDIF, skripty)

Adresářové struktury

- Adresářové struktury (directory structures)
  - Seznam objektů - entit (reálného) světa
    - Souborové systémy s adresáři jsou podmnožina
  - Standardizováno – řada X.500 (ITU)
    - (X.509 jsme již probírali - certifikáty)
    - Standard obsahuje mj.
      - Formát
        - Včetně doporučení vnáření objektů
          - např. “[country] - organization - organizationalUnit”
      - DAP Directory Access Protocol
      - Replikační principy pro distribuovanou implementaci

Adresářové struktury

- Adresářová databáze
  - Schéma
    - DIT - directory information tree
    - Objekty obsahují (jsou “adresovatelné”)
      - DN - distinguished name, jednoznačné jméno v celém stromu
      - RDN - relative distinguished name, relativně jednoznačné jméno v kontejneru
    - Každý objekt má atributy (AttributeAssertion)
      - Obsahují “jméno”
        - Objektový identifikátor, kterému je přiřazeno jméno
      - Hodnotu
        - Obvykle je vázána nějakou syntaxí (obor hodnot), max. délkou
        - Definují se
          - vyhledávací operace (equals, substring, ...)
          - typ single nebo multivalue

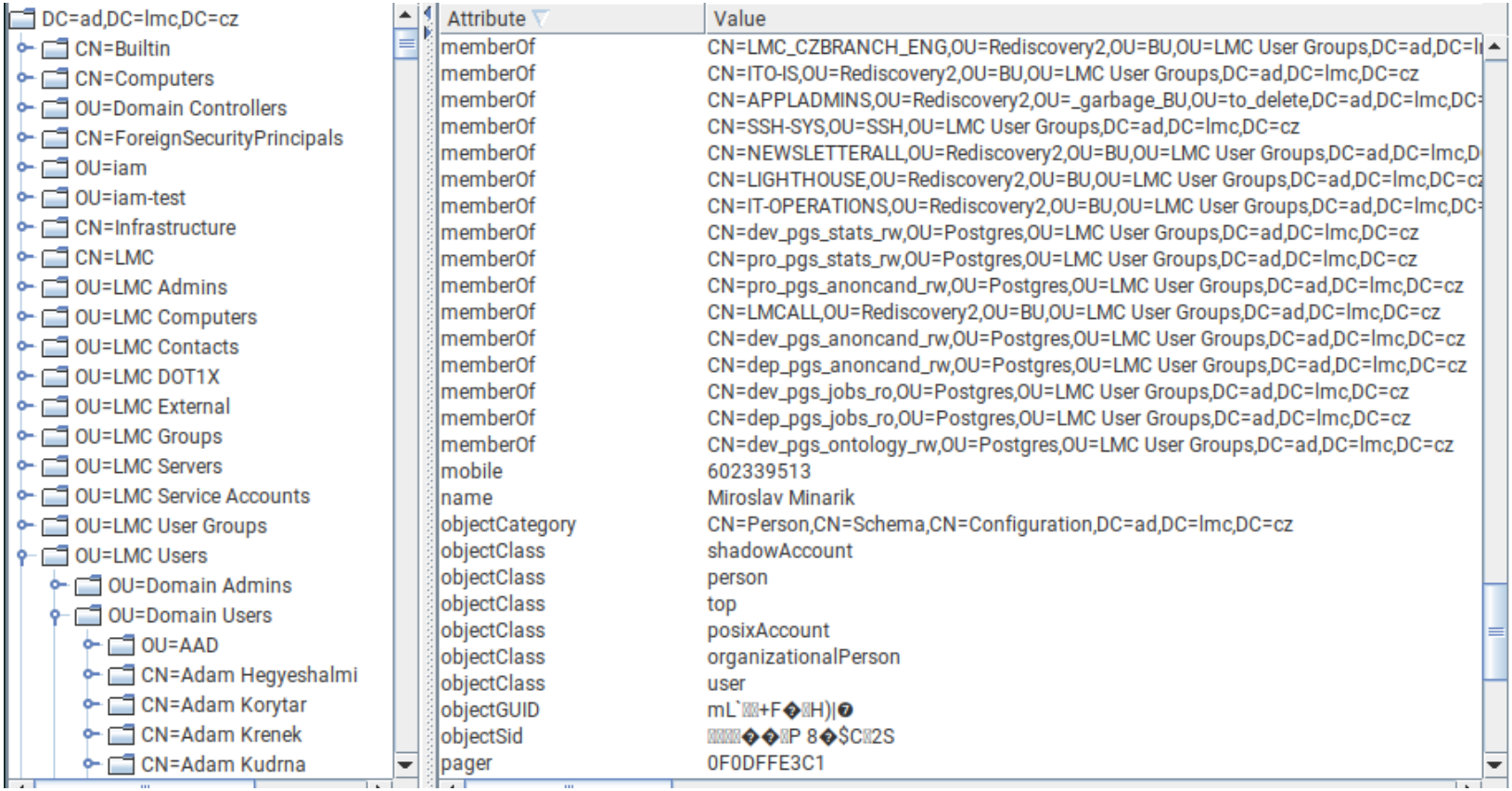
Adresářové struktury

- Adresářové servery – známé implementace
  - OpenLDAP
  - NDS, inetq
  - MS Active Directory
  - Apache DS (pure java)
  - … (většina CA má vlastní řešení, některé ho i prodávají...)
- Základní vlastnosti implementací
  - Distribuované databáze (režim multimaster s RW a R replikami)
    - Možnost replikace jen části stromu
    - Nemusí existovat replika obsahující všechna data (reference/linky...)
  - Možnost úpravy schématu

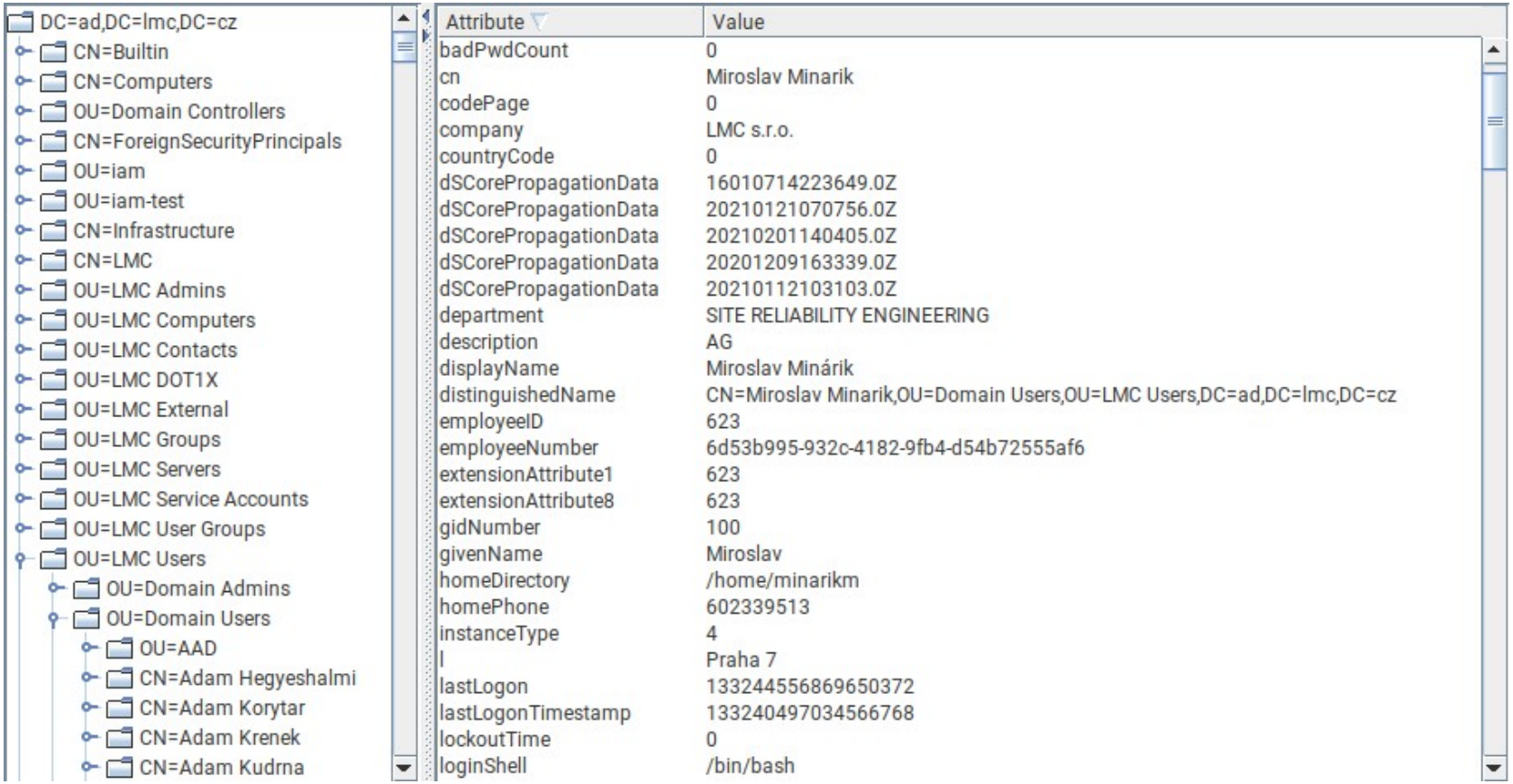
Adresářové struktury

- Prohlídka adresářové DB
  - GUI
    - LDAP browser 1998
    - Apache Directory Studio https://directory.apache.org/studio/
    - ...
  - Live demo
    - Stromové uspořádání
    - Self
    - cn, member, objectClass
    - ...

Adresářové struktury



Adresářové struktury



Adresářové struktury

- Schéma databáze
  - Princip - instance objektu je vytvořená na základě tříd, které předepisují, jaké atributy bude obsahovat
    - Ve speciálním atributu objectClass je seznam těchto tříd
      - třída “top” je univerzální, osahují ji všechny objekty
      - A další jako organization, person, ...
    - Další atributy – ID (“jméno”) + hodnota
  - Pro popis instance objektů (tříd, objectClass) a atributů (attributeType) se používá specifický jazyk
    - Je stadné přidávat další třídy a atributy (rozšiřovat schema), ale obvykle nejde v reálném provozu odebírat
  - Některé servery schema zveřejňují online (nutné, pokud chcete na klientech hlídat syntaxi)

Adresářové struktury

- Schéma databáze v praxi … OpenLDAP
  - Najděme schémata
    - https://openldap.org → “source” → github/gitlab/…
      - Mají vlastní doménu
      - Hledáme adresář .../servers/slapd/schema
  - Serveru v konfiguraci nastavujeme seznam, která se mají použít
    - core.schema – základ
    - nis.schema - unix profil uživatele…
    - ...

Adresářové struktury

- Schémata OpenLDAP
  - U certifikátů jsme se setkali s organization, organizationalUnit, common name…
- Otevřeme core.schema
  - Hledejme (včetně ')
    - (Třídy)
      - 'organization'
        - … name, OID, alternativní name(s), description, may, must (list)
      - 'organizationalUnit'
    - (Atributy)
      - 'o'
        - … předek, name, OID, alternativní name(s), description ...
      - 'ou'
        - 'name'
          - ... syntax (obor hodnot) a případně max délka, operatios

Adresářové struktury

- Schémata OpenLDAP - core.schema
  - 'organization', 'organizationalUnit'

```
objectclass ( 2.5.6.4 NAME 'organization'
        DESC 'RFC2256: an organization'
        SUP top STRUCTURAL
        MUST o
        MAY ( userPassword $ searchGuide $ seeAlso $ businessCategory $
                x121Address $ registeredAddress $ destinationIndicator $
                preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $
                telephoneNumber $ internationalISDNNumber $
                facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $
                postalAddress $ physicalDeliveryOfficeName $ st $ l $ description ) )

objectclass ( 2.5.6.5 NAME 'organizationalUnit'
        DESC 'RFC2256: an organizational unit'
        SUP top STRUCTURAL
        MUST ou
        MAY ( userPassword $ searchGuide $ seeAlso $ businessCategory $
                x121Address $ registeredAddress $ destinationIndicator $
                preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $
                telephoneNumber $ internationalISDNNumber $
                facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $
                postalAddress $ physicalDeliveryOfficeName $ st $ l $ description ) )
```

Adresářové struktury

- Schémata OpenLDAP – core.schema
  - 'o', 'ou', 'name'

```
attributetype ( 2.5.4.10 NAME ( 'o' 'organizationName' )
        DESC 'RFC2256: organization this object belongs to'
        SUP name )

attributetype ( 2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
        DESC 'RFC2256: organizational unit this object belongs to'
        SUP name )


# system schema
#attributetype ( 2.5.4.41 NAME 'name'
#       EQUALITY caseIgnoreMatch
#       SUBSTR caseIgnoreSubstringsMatch
#       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

Adresářové struktury

- Schémata OpenLDAP
  - Soubory *.schema
    - jazyk pro popis struktur (tříd a atributů)
      - ObjectClass
        - Name, OBJECT_IDENTIFIER, description
        - “dědí”
        - Must, may atribut nebo seznam
      - Atribut
        - Name, OBJECT_IDENTIFIER, description
        - Syntaxe, max délka
        - operace

Adresářové struktury

- OpenLDAP tools
  - Balíček pro Linux
    - Dle distribuce
      - ldap-utils
      - openldap2-client
  - Tools
    - ldapsearch
    - ldapadd
    - ldapmodify
    - ldapdelete

Adresářové struktury

- OpenLDAP tools - použití
  - man ldap...
    - -H URI (např. ldaps://ldap.fjfi.cvut.cz)
    - -D user_dn (uživatel nebo objekt, pod kterým se přihlašujeme)
    - heslo
      - -W (zadáme interaktivně)
      - -w password (na příkazovém řádku, nepoužívat pro produkční systémy)
      - -z file_with_password (heslo je uloženo v souboru, pozor na CR a/nebo LF)
    - ...search
      - -b base_dn (místo ve stromu, vůči němuž relativně pracujeme)
      - -s scope (jedno z { sub once base}, “hloubka” prohledávání)
      - filtr (např. "(cn=minar*)"
      - Volitelně seznam atributů
    - ...add, ...modify, ...delete
      - -f ldif_file (soubor s požadovanými změnami)

Adresářové struktury

- OpenLDAP tools – použití

```
ldapsearch -H ldaps://ldap.fjfi.cvut.cz  \
    -b DC=fjfi,DC=cvut,DC=cz \
    -s sub \
    -D "cn=minarmir, ou=KM, ou=People, DC=fjfi, DC=cvut, DC=cz" \
    -W \
    "(cn=minar*)" \
    cn objectClass description member
```

Adresářové struktury

- OpenLDAP tools – použití

```
ldapsearch -H ldaps://ad.service.consul  \
    -b "OU=domain users,OU=LMC Users,DC=ad,DC=lmc,DC=cz" \
    -s sub \
   -D "CN=Miroslav Minarik, OU=Domain Users,OU=LMC Users,DC=ad,DC=lmc,DC=cz" \
    -W \
    "(sAMAccountName=minar*)" \
    cn objectClass description member
```

Adresářové struktury

- OpenLDAP tools – použití
  - Výstup ldapsearch je ve formátu LDIF
    - LDAP Data Interchange Format
    - Parametrem -L lze upravit výstup (verze, komentáře, ...)
  - LDAP filtry
  - Pokud vynecháme seznam atributů, dostaneme všechny

Adresářové struktury

- LDIF
  - LDAP Data Interchange Format
  - Základní forma

```
dn: <distinguished name>
<attribute1_name>: <value1>
<attribute2_name>: <value2>

dn: <distinguished name2>
<attribute3_name>: <value3>
<attribute4_name>: <value4>
<attribute5_name>: <value5>
```

    - UTF-8, ::, šířka

Adresářové struktury

- LDIF
  - ::
    - Hodnota atributu je binární, je zde zakódován base64
  - https://en.wikipedia.org/wiki/LDAP_Data_Interchange_Format
  - Standard a formální zápis RFC
    - https://datatracker.ietf.org/doc/html/rfc2849
    - Použijeme později s ldapmodify

Adresářové struktury

- LDAP filtry
  - Základní
    - (<name><operation><value>)
      - <name> jméno atributu (např. cn, ou, o, member, memberOf, objectClass,…)
      - <operation> porovnání (např. =)
      - <value> dle definovaných operací na atributu může být číslo, řetězec nebo řetězec se žolíky (*) apod
      - Poznámky
        - Závorky “ohraničují” filtr
        - Pokud používáme shell musíme vložit do uvozovek " nebo apostrofů '
    - Příklad

      (uid=minar*)

Adresářové struktury

- LDAP filtry
  - S logickými operátory
    - AND

      (& (objectClass=person)(cn=ales) ...)
    - OR

      (| (objectClass=person)(cn=ales) ...)
    - NOT

      (! (cn=ales))
    - Příklady

      (& (objectClass=person)(memberOf=cn=kf_stu_…))
      - Poznámka – pro memberOf musíme použít úplné DN jméno skupiny, nefungují * ani části jako CN

      (&(objectClass=group)(cn=kdaiz))

      (objectClass=group)

Adresářové struktury

- LDAP filtry - příklady
  - (& (objectClass=person)(memberOf=cn=kf_stu_…))
    - Poznámka – pro memberOf musíme použít úplné DN jméno skupiny, nefungují * ani části jako CN
  - (&(objectClass=group)(cn=kdaiz))
  - (objectClass=group)
  - (objectClass=person)
  - (cn=ales)
  - (& (objectClass=person)(cn=ales) ...)
  - (| (objectClass=person)(cn=ales) ...)
  - (! (cn=ales))

Adresářové struktury

- Testovací server (po dobu výuky)
  - CLI

```
ldapsearch -H ldap://kmlinux.fjfi.cvut.cz:4001 -b "dc=vyuka" -s sub \
    -D "cn=admin,dc=vyuka" -w openldap  "(cn=*)"
```

  - GUI
    - LDAP browser (1998)
      - Live demo
    - Apache Directory Studio

Adresářové struktury

- Testovací server - ldapadd
  - mujsoubor1

```
dn: o=prvniskolni,dc=vyuka
o: prvniskolni
objectClass: top
objectClass: organization
```

```
ldapadd -H ldap://kmlinux.fjfi.cvut.cz:3333 -D  "cn=admin,dc=vyuka" -w openldap -f mujsoubor1
```

Adresářové struktury

- Testovací server - ldapadd
  - mujsoubor2

```
dn: o=2skolni,dc=vyuka
o: 2skolni
objectClass: top
objectClass: organization
description: Tohle je test

dn: ou=podskolni,o=2skolni,dc=vyuka
ou: podskolni
objectClass: top
objectClass: organizationalUnit
```

```
ldapadd -H ldap://kmlinux.fjfi.cvut.cz:3333 -D  "cn=admin,dc=vyuka" -w openldap -f mujsoubor2
```

Adresářové struktury

- Testovací server - ldapadd
  - mujsoubor3

```
dn: cn=ales,o=2skolni,dc=vyuka
cn: ales
objectClass: top
objectClass: person
sn: Novak
description: Ales
```

```
ldapadd -H ldap://kmlinux.fjfi.cvut.cz:3333 -D  "cn=admin,dc=vyuka" -w openldap -f mujsoubor3
```

Adresářové struktury

- Testovací server - ldapmodify
  - mujsoubor4

```
dn: cn=ales,o=2skolni,dc=vyuka
changetype: modify
replace: sn
sn:: xxxxxxxxnovak...base64
-
replace: description
description:: xxxxxxxxales...base64
-
```

```
ldapmodify -H ldap://kmlinux.fjfi.cvut.cz:3333 -D  "cn=admin,dc=vyuka" -w openldap -f mujsoubor4
```

Adresářové struktury

- Testovací server - ldapmodify
  - RFC

```
changerecord        = "changetype:" FILL
                      (change-add / change-delete /
                       change-modify / change-moddn)

change-add          = "add"               SEP 1*attrval-spec

change-delete       = "delete"            SEP

change-moddn        = ("modrdn" / "moddn") SEP
                       "newrdn:" (    FILL rdn /
                                  ":" FILL base64-rdn) SEP
                       "deleteoldrdn:" FILL ("0" / "1")  SEP
                       0*1("newsuperior:"
                       (    FILL distinguishedName /
                        ":" FILL base64-distinguishedName) SEP)

change-modify       = "modify"            SEP *mod-spec

mod-spec            = ("add:" / "delete:" / "replace:")
                      FILL AttributeDescription SEP
                      *attrval-spec
                      "-" SEP

SPACE               = %x20
                      ; ASCII SP, space

FILL                = *SPACE

SEP                 = (CR LF / LF)

CR                  = %x0D
                      ; ASCII CR, carriage return
```

Adresářové struktury

- Testovací server - ldapmodify
  - mujsoubor5

```
dn: cn=tomas,... dc=vyuka
changetype: add
cn: tomas
objectClass: top
objectClass: person
sn: Novak
description: Tomas
telephoneNumber: 987

dn: cn=ales,... dc=vyuka
changetype: modify
replace: sn
sn:: xxxxxxxxnovak...base64
-
replace: description
description:: xxxxxxxxales...base64
-
add: telephoneNumber
telephoneNumber: 12345
-
```

Adresářové struktury

- Testovací server - ldapmodify
  - mujsoubor6

```
dn: cn=ales,.... dc=vyuka
changetype: modify
delete: telephoneNumber
-

dn: cn=ales,.... dc=vyuka
changetype: delete

dn: cn=tomas,.... dc=vyuka
changetype: delete
```

```
ldapmodify -H ldap://kmlinux.fjfi.cvut.cz:3333 -D  "cn=admin,dc=vyuka" -w openldap -f mujsoubor6
```

Adresářové struktury

- Testovací server - ldapdelete
  - mujsoubor7

dn: cn=ales,.... dc=vyuka
dn: cn=tomas,.... dc=vyuka

ldapdelete -H ldap://kmlinux.fjfi.cvut.cz:3333 -D "cn=admin,dc=vyuka" -w openldap -f mujsoubor7

Adresářové struktury

- Testovací server - ldapmodify
  - mujsoubor8

```
dn: cn=tomas,.... dc=vyuka
changetype: modrdn
newrdn: cn=tom
deleteoldrdn: 0

dn: cn=ales,.... dc=vyuka
changetype: moddn
newrdn: cn=ali
deleteoldrdn: 1
newsuperior: o=jineou,dc=vyuka
```

```
ldapmodify -H ldap://kmlinux.fjfi.cvut.cz:3333 -D  "cn=admin,dc=vyuka" -w openldap -f mujsoubor8
```

Adresářové struktury

- Testovací server - ldapmodify
  - RFC

```
changerecord        = "changetype:" FILL
                      (change-add / change-delete /
                       change-modify / change-moddn)

change-add          = "add"               SEP 1*attrval-spec

change-delete       = "delete"            SEP

change-moddn        = ("modrdn" / "moddn") SEP
                       "newrdn:" (    FILL rdn /
                                  ":" FILL base64-rdn) SEP
                       "deleteoldrdn:" FILL ("0" / "1")  SEP
                       0*1("newsuperior:"
                       (    FILL distinguishedName /
                        ":" FILL base64-distinguishedName) SEP)

change-modify       = "modify"            SEP *mod-spec

mod-spec            = ("add:" / "delete:" / "replace:")
                      FILL AttributeDescription SEP
                      *attrval-spec
                      "-" SEP

SPACE               = %x20
                      ; ASCII SP, space

FILL                = *SPACE

SEP                 = (CR LF / LF)

CR                  = %x0D
```

Adresářové struktury

- Skripty
  - Scénář
    - Connect po zabezpečeném spojení
      - ldaps
    - Bind
      - Typicky servisním účtem, lze použít i vlastní účet
    - Search
  - HA a související
    - Řešeno “za adresou” (CNAME na Lbs)
      - Může mít nastavení pro zavření při neaktivitě klienta I serveru (LMC 5 nebo 10s)
    - Knihovny obvykle podporují seznam serverů a nastavení connect timeout (než zkusí další). 1100ms pro 2 pokusy
    - (read timeout – na klientu pro neaktivitu ze strany serveru)
    - Query lze omezovat timeoutem na zpracování a limitem na počet nalezených

Adresářové struktury

- Skripty
  - To be continued...
    - Java (classic JNDI, ale v groovy)
    - Python
    - PHP
    - PowerShell

Adresářové struktury

- Skripty
  - Nejprve společný základ

```
export LDAP_USER="CN=Miroslav Minarik,OU=Domain Users,OU=LMC Users,DC=ad,DC=lmc,DC=cz"
#export LDAP_USER=minarikm@ad.lmc.cz
export LDAP_PASS="MOJE_HESLO"
export LDAP_SERVER=ldaps://ad.service.consul
export LDAP_CA_CERTS=/etc/ssl/ca-bundle.pem    # může se lišit podle distra
```

Adresářové struktury

- Demo – PHP connect and bind

```
<?php
$ldap_user = getenv( 'LDAP_USER');
$ldap_pass = getenv( 'LDAP_PASS');
$ldap_server = getenv( 'LDAP_SERVER');
$ldap_ca_certs = getenv( 'LDAP_CA_CERTS');
putenv( 'LDAPTLS_REQCERT=require');
putenv( "LDAPTLS_CACERT=$ldap_ca_certs");
//ldap_set_option(NULL, LDAP_OPT_DEBUG_LEVEL, 7);
$ldap_conn = ldap_connect( $ldap_server) or die ("Couldn't connect");
$ldap_bind = ldap_bind($ldap_conn, $ldap_user, $ldap_pass);
if ($ldap_bind) {
    print("\n logged in! \n\n");
    // SOMEWORK...
}
?>
```

Adresářové struktury

- Demo – PHP search

```
$sr=ldap_search( $ldap_conn, "OU=LMC Users,DC=ad,DC=lmc,DC=cz",
            "(uid=minarik*)", array("uidNumber", "gidNumber", "memberof", "xxx"));
  $result = ldap_get_entries( $ldap_conn, $sr);
  for( $i = 0; $i < $result[ "count"]; $i++) {
    $obj = $result[ $i];
    print( $obj[ "dn"]."\n  ");
    if( in_array( "uidnumber", $obj)) {
      // $obj[ "uidnumber"][ "count"]
      print( "u: ".$obj[ "uidnumber"][ 0]." ");
    }
    if( in_array( "xxx", $obj)) {
      // $obj[ "xxx"][ "count"]
      print( "x: ".$obj[ "xxx"][ 0]." ");
    }
    print( "\n");
    if( in_array( "memberof", $obj)) {
      $mo_cnt = $obj[ "memberof"][ "count"];
      for( $mo = 0; $mo < $mo_cnt ; $mo++) {
        print( "    ".$obj[ "memberof"][ $mo]."\n");
      }
    }
  }
```

Adresářové struktury

- Demo - Python connect and bind

```
#!/usr/bin/env python3
import ldap
import os
import sys
ldap_user = os.environ[ 'LDAP_USER'];
ldap_pass = os.environ[ 'LDAP_PASS'];
ldap_server = os.environ[ 'LDAP_SERVER'];
ldap_ca_certs = os.environ[ 'LDAP_CA_CERTS'];
#os.environ[ 'LDAPTLS_REQCERT'] = 'require'    TODO
#os.environ[ 'REQUESTS_CA_BUNDLE'] = ldap_ca_certs  TODO
    # ldap.OPT_X_TLS_CACERTDIR
ldap_conn = ldap.initialize( ldap_server )
ldap_bind = ldap_conn.simple_bind_s( ldap_user, ldap_pass)
if ldap_bind:
    print( "\n logged in! \n\n");
    # SOMEWORK
```

Adresářové struktury

- Demo - Python search

```
result = ldap_conn.search_s( "OU=LMC Users,DC=ad,DC=lmc,DC=cz",
    ldap.SCOPE_SUBTREE, "(uid=minarik*)",
    [ "uidNumber", "gidNumber", "memberOf", "xxx"]);
 for obj in result:
    print( '-----')
    print( 'dn: {}'.format( obj[ 0]))
    print( 'u: {}, g: {}, x: {}'.format(
        obj[ 1][ 'uidNumber'],
        obj[ 1][ 'gidNumber'],
        obj[ 1][ 'xxx'] if 'xxx' in obj[ 1] else ''
        ))
    print( 'memberOf: {}'.format( obj[ 1][ 'memberOf'] ))
```

Adresářové struktury

- Demo – Java connect and bind

```
import javax.naming.*
import javax.naming.directory.*

def sysenv = System.getenv()
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.sun.jndi.ldap.LdapCtxFactory")
env.put(Context.PROVIDER_URL, sysenv[ 'LDAP_SERVER'])
env.put(Context.SECURITY_AUTHENTICATION, "simple")
env.put(Context.SECURITY_PRINCIPAL, sysenv[ 'LDAP_USER'])
env.put(Context.SECURITY_CREDENTIALS, sysenv[ 'LDAP_PASS'])

Context ctx = new InitialDirContext(env);
// SOMEWORK
ctx.close();
```

Adresářové struktury

- Demo – Java search

```
SearchControls searchControls = new SearchControls();
searchControls.setSearchScope(SearchControls.SUBTREE_SCOPE);
//searchControls.setTimeLimit(30000);
//List attrs = [ 'uidNumber', 'gidNumber', 'memberOf', 'xxx']
//searchControls.setReturningAttributes( attrs)

def namingEnum = ctx.search( "OU=LMC Users,DC=ad,DC=lmc,DC=cz",
        "(uid=minarik*)", searchControls)
for( SearchResult result: namingEnum) {
  Attributes attrs = result.getAttributes ()
  println(attrs["distinguishedName"])
  println( "  u: ${attrs[ 'uidNumber']}, g: ${attrs[ 'uidNumber']}, x: ${attrs[ 'xxx']}")
  println( "  m: ${attrs[ 'memberOf']}")
  println '-----'
}
```

Adresářové struktury

- Demo – Java trust store (s důvěryhodnými certifikáty)
  - Seznam důvěryhodných (default heslo changeit)
    - keytool -keystore ./cacerts -list
    - keytool -keystore ./cacerts -list -v
  - Přidání nového důvěruhodného CA

    keytool -keystore ./cacerts -importcert -file /etc/ssl/certs/lmc_ad_root.pem -alias lmcad

    keytool -keystore /usr/lib/jvm/adoptopenjdk-8-hotspot-amd64/jre/lib/security/cacerts \

    -importcert -file /etc/ssl/certs/lmc_ad_root.pem -alias lmcad
  - Použití vlastního keystoru (s důvěryhodnými certifikáty)

    java \

    -Djavax.net.ssl.trustStore=/secrets/cacerts.jks \

    -Djavax.net.ssl.trustStorePassword=changeit \

    ...

Adresářové struktury

- Demo - simple PowerShell

  TODO

```
$user = Get-ADUser minarikm -Properties @( 'uid', 'uidNumber', 'gidNumber') \
    -Filter { SamAccountName -gt 'x' } \
    -LDAPFilter "(uid=minar*)"
# -SearchScope Subtree -SearchBase …
```

  - Get-ADGroup
  - PS s profilem (online exchange, … Get-Credentials)
  - Output

```
$user
echo $user
write-host -foreground red $user
```