



1

---

---

---

---

---

---

---

---



2

---

---

---

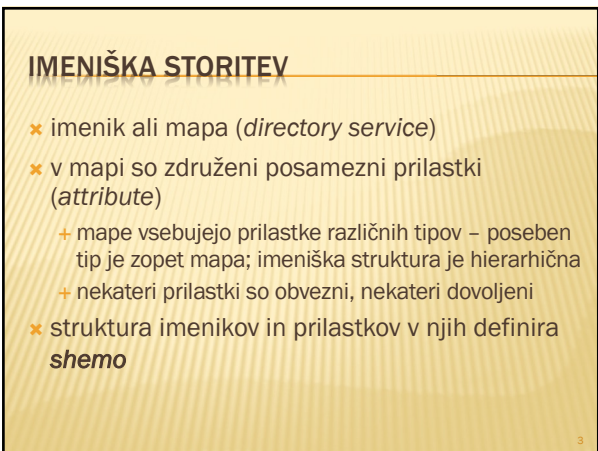
---

---

---

---

---



3

---

---

---

---

---

---

---

---

## PRILASTKI

- ✗ vsak prilastek ima svoje ime
- ✗ v isti mapi imamo lahko več prilastkov z istim imenom, a z različnimi vrednostmi – prim. s podatkovno strukturo slovar
- ✗ ista imena v različnih mapah predstavljajo različne prilastke
  - + primer: v Javi a.b.c ni enako a.c.c
  - + **izziv: Kje smo to že srečali?**

4

---

---

---

---

---

---

---

---

## PREDMETI IN IMENSKI PROSTOR

- ✗ predmeti ali objekti (*objects* tudi včasih *entries*) so dejanske vrednosti, ki jih hrani imeniška struktura glede na definirano shemo
- ✗ predmeti, ki so vstavljeni v imenik, so v vsebniku (vsebovalniku, *container*)
- ✗ vsi predmeti v vsebniku so v istem imenskem prostoru (*namespace*)
  - + vsebnik je podobna struktura kot slovar

5

---

---

---

---

---

---

---

---

## IMENSKI PROSTOR IN RAZLIKOVANJE

- ✗ predmeti v imenskem prostoru so ponovno hierarhično oblikovani
- ✗ predmete moramo med seboj razlikovati
  - + način razlikovanja je del **načrtovanja** imenika
  - + za razlikovanje moramo uporabiti pravila, ki določajo enolično in nedvoumno ime
- ✗ **predmeti »živijo« v imenskem prostoru in ne v vsebovalniku**

6

---

---

---

---

---

---

---

---

## RAZLIKOVANJE PREDMETOV

- ✘ ime, po katerem razlikujemo predmete, imenujemo **razločevalno ime** (*distinguished name*)
- ✘ razločevalno ime je lahko absolutno ali relativno – glede na hierarhijo imenikov
- ✘ razločevalno ime (običajno) **ni shranjeno** v imeniški strukturi, ampak je definirano s pravili

7

---

---

---

---

---

---

---

---

## RAZLIKOVANJE PREDMETOV

- ✘ primer – EDUROAM:

```
dn: dc=fakulteta,dc=univerza,dc=si
objectClass: top
objectclass: dcObject
objectClass: organization
dc: es-kranj
o: Fakulteta in Univerza
```

8

---

---

---

---

---

---

---

---

## IMENSKI PROSTOR IN UPRAVLJANJE

- ✘ vsebino imenskega prostora lahko:
  - + porazdelimo med različne strežnike (*distribution*) – porazdeljena imeniška storitev
  - + prepisemo še na drug strežnik (*replication*) – z vsebino imenskega prostora še vedno upravlja izvorni strežnik

9

---

---

---

---

---

---

---

---

## PODATKOVNE BAZE IN IMENIŠKE STORITVE

- ✘ običajna, relacijska, podatkovna baza je organizirana v tabelah
- ✘ v imeniški strukturi imamo tudi prilastke, ki pa so:
  - + obvezni – podobno podatkovnim bazam
  - + neobvezni – na nek način *null* vrednosti v bazah
  - + se lahko ponovijo
- + prilastki in njihova struktura so standardizirani (IANA)
- + predmeti so razvrščeni v imenske prostore, pri čemer posamezen predmet podeduje vse lastnosti starša

10

10

## DNS STORITEV

- ✘ dejansko je DNS imeniška storitev
  - + **obvezno: poiščite RFC ter ga preberite – literatura**
- ✘ imenski prostor določa FQN (*fully qualified name*)
- ✘ prilastki določajo storitve v imenskem prostoru
- ✘ pojem dedovanja ni izkoriščen

TYPE	meaning
A	a host address
AAAA	a host address
NS	an authoritative name server
MD	a mail destination (Obsolete - use MX)
MF	a mail forwarder (Obsolete - use MX)
CNAME	the canonical name for an alias
SOA	marks the start of a zone of authority
MB	a mailbox domain name (EXPERIMENTAL)
MG	a mail group member (EXPERIMENTAL)
MR	a mail rename domain name (EXPERIMENTAL)
NULL	a null record (EXPERIMENTAL)
WKS	a well known service description
PTR	a domain name pointer
RINFO	host information
MINFO	mailbox or mail list information
MX	mail exchange
TXT	text strings

11

11

## DNS STORITEV IN POŠILJANJE POŠTE

```
[AndyBook:~] andy% nslookup
> acm.si
Server:      192.168.126.1
Address:    192.168.126.1#53
```

```
Non-authoritative answer:
Name:   acm.si
Address: 193.2.2.76.34
> set query=txt
> acm.si
Server:      192.168.126.1
Address:    192.168.126.1#53
```

```
Non-authoritative answer:
acm.si text = "v=spf1 ip4:88.200.63.134 ip4:212.235.226.73
a:ns.fri.uni-lj.si a:mail-fri.friil.uni-lj.si -all"
```

12

12





### LDAP – LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

- ✦ Opisan v RFC-jih 4510 – 4519
  - + RFC 4510: imenik in pregled za ostale RFCje
  - + RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*: protokol komunikacije
  - + RFC 4512, *Lightweight Directory Access Protocol (LDAP): Directory Information Models*: opis imeniške strukture, sheme, prilastki, razredi
  - + **izziv: poiščite RFC4511 in RFC4512 in ju preberite. Kako se povezujeta med seboj?**
- ✦ [RFC 4513 - LDAP: Authentication Methods and Security Mechanisms](#)
- ✦ [RFC 4514 - LDAP: String Representation of Distinguished Names](#)
- ✦ [RFC 4515 - LDAP: String Representation of Search Filters](#)
- ✦ [RFC 4516 - LDAP: Uniform Resource Locator](#)
- ✦ [RFC 4517 - LDAP: Syntaxes and Matching Rules](#)
- ✦ [RFC 4518 - LDAP: Internationalized String Preparation](#)
- ✦ [RFC 4519 - LDAP: Schema for User Applications](#)

16

---

---

---

---

---

---

---

---

16

### LDAP – LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

- ✦ obstaja dve inačici: v2 in v3
- ✦ v2 je definirana v RFC1777-1779
  - + v2 je umaknjena iz uporabe ([RFC 3494 - Lightweight Directory Access Protocol version 2 \(LDAPv2\) to Historic Status](#))
- ✦ dopolnitve za v3 so definirane v kopici RFCjev
  - + **obvezno: v čem se inačica tri razlikuje od inačice dve?**

17

---

---

---

---

---

---

---

---

17

### LDAP

- ✦ LDAP je predvsem protokol za komunikacijo, ki pa upošteva metashemo shranjenih podatkov
- ✦ kako se podatki shranjujejo pri strežniku, protokol ne določa
- ✦ različne implementacije: *OpenLDAP*, *ActiveDirectory*, ...

18

---

---

---

---

---

---

---

---

18

## LDAP – PROTOKOL

- ✘ odjemalec prične komunikacijo s strežnikom na dobro poznanih vratih
- ✘ na voljo ima nekaj ukazov (RFC 4511):
  - + *start TLS* – preklop na TSL/SSL način komunikacije (druga možnost je namestitev strežnika na drugih vratih in izvajanje celotne komunikacije prek SSL protokola – *ldaps*)
  - + *izziv: katera so vrata za ldap protokol in katera za ldaps?*

19

19

---

---

---

---

---

---

---

---

## LDAP – PROTOKOL

- ✘ ukazi, nadaljevanje:
  - + *bind* – želja po overitvi ter ostalih možnih parametrih komunikacije (inačica, ...). Seja je lahko tudi neoverovljena.
  - + *unbind* – zaključek komunikacije (seje).

20

20

---

---

---

---

---

---

---

---

## LDAP – PROTOKOL

- ✘ ukazi, nadaljevanje:
  - + *search* – iskanje posameznih predmetov v bazi. Rezultat odvisen lahko odvisen od tega, ali je odjemalec overovljen ali ne.
    - ✘ `ldapsearch -L -D 'cn=foo,dc=bar,dc=com' 'objectclass=posixAccount'`
  - + *compare* – možnost primerjave vrednosti predmeta. Ni potrebno razkriti prave vrednosti predmeta, samo preverjamo enakost. Primerno za gesla in podobno.

21

21

---

---

---

---

---

---

---

---

## LDAP - PROTOKOL

- ✘ ukazi, nadaljevanje:
  - + *add* – dodamo predmet v bazo
  - + *delete* – pobrišemo predmet iz baze
  - + *modify* – spremenimo vrednosti prilastkov predmeta
  - + *modify DN* – spremenimo ime predmeta (*rename*)
    - ✘ `ldapmodify -r -D 'cn=foo,dc=bar,dc=com' -W </code>`
    - `/tmp/user.ldif`

22

22

---

---

---

---

---

---

---

---

## LDAP - PROTOKOL

- ✘ ukazi, nadaljevanje:
  - + *abandon* – prekinemo izvajanje zahteve, ki smo jo poslali (lahko prekinemo iskanje in primerjanje ter popravke baze)
  - + *extended* – generična možnost poljubnega dodatnega ukaza

23

23

---

---

---

---

---

---

---

---

## LDAP SCHEMA, RAZREDI IN PRILASTKI

- ✘ shema združuje različne predmete in prilastke
  - + uporabljamo lahko tudi vključevalne ukaze (*include*) za poenostavitev modularizacije
- ✘ razredi (*objectClass*) združujejo prilastke
  - + opisani z zapisom ASN.1
  - + so del hierarhije in dedujejo lastnosti starša
  - + določajo obvezne in neobvezne prilastke

24

24

---

---

---

---

---

---

---

---



## LDAP SCHEME, RAZREDI IN PRILASTKI

- ✗ prilastki (*attribute*) opisuje lastnosti
  - + opisani z zapisom ASN.1
  - + na nek način definicija tipa
  - + njihovo udejanjenje (instanciacija) bo dejansko hranila vrednosti
  - + opisujejo tudi sintakso, način primerjave ipd.

25

25

---

---

---

---

---

---

---

---

## RAZREDI

```
ObjectClassDescription =
"(" whsp
  numericoid whsp
  ; ObjectClass identifier
  [ "NAME" qdstring ]
  [ "DESC" qdstring ]
  [ "OBSOLETE" whsp ]
  [ "SUP" oids ]
  ; Superior ObjectClasses
  [ ( "ABSTRACT" /
    "AUXILIARY" ) whsp ]
  ; default structural
  [ "MUST" oids ]
  ; AttributeTypes
  [ "MAY" oids ]
  ; AttributeTypes
  whsp ")"
```

✗ primer definicije razreda:

```
objectclass (
  2.5.6.2
  NAME 'country'
  SUP top
  STRUCTURAL
  MUST c
  MAY ( searchGuide $
  description )
)
```

26

26

---

---

---

---

---

---

---

---

## LDAP IN PODATKI

- ✗ pri prenašanju podatkov med LDAP strežniki uporabljamo format LDIF:

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

27

27

---

---

---

---

---

---

---

---

**PROGRAMSKA OPREMA**

- ✘ na FreeBSD/Linux OpenLDAP
- ✘ strežniški in uporabniški programi:
  - + slapd, slurpd
  - + ldapcomapre, ldapdelete, ...
- ✘ konfiguracijske datoteke v /usr/local/etc
- ✘ več na vajah
  - ✘ *izziv: namestite OpenLDAP na vaš strežnik in ga skofigurirajte*

28

---

---

---

---

---

---

---

---

**PROGRAMSKA OPREMA**

- ✘ uporabniški programi lahko vsebujejo možnost zajema podatkov iz LDAP strežnika
  - + freeradius, overovljenje na unix-ih, ...

29

---

---

---

---

---

---

---

---