INFORMATION FOR NETWORK OPERATION

CONTENT

- Directory service
- Standard X.500
- LDAP

DIRECTORY SERVICE

- Directory service
- in the folder are grouped individual attributes
  - folders contain attributes of different types – special type is folder again; directory structure is hierarchical
  - Some attributes are required, some allowed
  - directory structure and attributes in them are defining scheme

ATRIBUTES

- Each atribute has a name
- in the same folder we can have multiple attributes with the same name, but with different values - cf. with data structure dictionary
- The same name in diferent directories represent diferent atribute
  - Cf. In Java a.b.c doesn't equal a.c.c
  - Chalange: Where have we already met this?

OBJECTS AND NAMESPACES

- Objects (or entries) are the actual values held by directory structure according to the defined scheme
- objects that have been included in the directory are in container
- All objects in container are in the same namespace
  - Container is similiar structure as dictionary

NAMESPACE AND THE DISTINCTION

- objects in the name space are re-developed hierarchically
- objects must differ from each other
  - way of distinguishing is the part of the designing directory
  - Rules must be used to distinguish to provide a unique and unambiguous name
- objects „lives" in the namespace and not in container

DISTINGUISHING OBJECTS

- name to distinguish objects is called a distinguished name
- distinguished name can be absolute or relative - depending on the hierarchy of directories
- distinguished name is (usually) not stored in directory structure, but is defined by the rules

DISTINGUISHING OBJECTS

- example – EDUROAM:

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

NAMESPACE AND MANAGEMENT

- Cotent of namespace can:
  - Be divided between different servers (distribution) - distributed directory service
  - be rewritten in a different server (replication) – namespace content is still managed by original server

DATABASES AND DIRECTORY STRUCTURES

- traditional, relational, database is organized in tables
- In directory structure we also have attributes, which are:
  - Required –similiar to databases
  - Optional – in some way null values in databases
  - Repeated
  - attributes and their structure are standardized (IANA)
  - objects are grouped in namespaces, and each object inherits all the properties of parents

DNS

- DNS is actually directory service
  - Required: find the RFC and read it – literature
- namespace provides FQN (Fully Qualified Name)
- attributes provide services in the namespace
- concept of inheritance is not utilized

- TYPE meaning
- ----------
- A 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 RR (EXPERIMENTAL)
- WKS a well known service description
- PTR a domain name pointer
- HINFO host information
- MINFO mailbox or mail list information
- MX mail exchange
- TXT text strings

SOFTWARE

- On FreeBSD named
- Konfiguration in /etc/named/*
  - Chalange: install DNS server for your own domain and configure it

```
$ORIGIN     brodnik.name.
@           SOA     Svarun  (
                            2007012002
                            10800
                            3060
                            1814400
                            86400 )
@           IN      NS      Svarun
;-------------------------------------------------------
;-------------------------------------------------------
Svarun      IN      A       193.77.156.167
Svarun      IN      HINFO   i586 FreeBSD
;-------------------------------------------------------
;-------------------------------------------[ strezniski aliasi ]
;-------------------------------------------------------
Posta       IN      CNAME   Svarun
@           IN      MX 50   Posta
WWW         IN      CNAME   Svarun
```

```
hostmaster (
; Serial == YYMMDD
; Refresh of cache (in seconds)
; Retry interval for refresh
; Expire of secondary copy
; Default minimum expiration time
Svarun
```

STANDARD X.500

- For detailed description look:
  - http://www.x500standard.com/
- actually a family of standards
  - example: X.509 was the basis for SPKI
  - Chalange: find RFC for SPKI and find connection between SPKI and X.509
  - Required: find on the internet hod X.509 certificate is defined and compare it to SPKI certificate
- for the operation of the postal system in X standard (X.400) was necessary directory structure

STANDARD X.500

- consisting of 4 protocols
- protocol for accessing directory structure - operations on structure: Bind, Read, List, Search, Compare, Modify, Add, Delete and ModifyRDN
- standard defines the namespace, and in it are located objects
- each object is identified by its distinguishing name
- object can have one or more (also repeated) attributes
- directory structure consists of a single directory
  - individual parts of directory directory are used by various servers

LDAP – LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

- Described in RFCs 4510 – 4519
  - RFC4510: directory and check for other RFCs
  - RFC4511, Lightweight Directory Access Protocol (LDAP): The Protocol: communication protocol
  - RFC 4512, Lightweight Directory Access Protocol (LDAP): Directory Information Models: description of directory structures, schemas, attributes, classes
    - challenge: find RFC4511 and RFC4512, and read them. How they relate to each other?
- 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

LDAP – LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

- Ther are two versions: v2 and v3
- V2 is defined in RFC1777-1779
  - v2 is withdraw n from service (RFC 3494 – Lightweight Directory Access Protocol version 2 (LDAPv2) to Historic Status)
- additions to v3 are defined in a variety of RFCs
  - Required: what is the diference between v2 and v3?

LDAP

- LDAP is a protocol primarily for communication but also takes into account the metascheme stored data
- Protocol doesn‘t provide how data is stored on server
- different implementations: OpenLDAP, ActiveDirectory, ...

LDAP - PROTOCOL

- client begins to communicate with the server on well-known port
- it has a few commands available (RFC 4511):
  - start TLS - switch to SSL mode of communication (The alternative is to install a server on other port and implement a comprehensive communications via SSL Protocol - ldaps)
    - challenge: which port is used for ldap protocol and which for ldaps?

LDAP - PROTOCOL

- commands, continued:
  - bind - the desire for authentication and other possible communications parameters (version, ...). The sessioncan be also unauthorized.
  - unbind – the end of communication (session).

LDAP - PROTOCOL

- commands, continued:
  - search - search for individual objects in the database. The result depends on whether the client is authenticated or not.
    - ldapsearch-L-D 'cn=foo, dc=bar, dc=com' 'objectclass=posixAccount,
  - compare - the ability to compare values of object. It is not necessary to reveal the true value, it only check equality. Suitable for passwords and things like that.

LDAP - PROTOCOL

- commands, continued:
  - add - add an object in the database
  - delete - delete the object from the database
  - modify - change the value of object attributes
  - modify DN - change the object name (rename)
    - ldapmodify -r -D 'cn=foo, dc=bar, dc=com' -W < / tmp/user.ldif

LDAP - PROTOCOL

- commands, continued:
  - abandon - terminate processing requests, which we sended (it can be cancel search, comparison and corrections to the database)
  - extended - generic option for any additional command

LDAP SCHEME, CLASS and ATRIBUTES

- scheme combines various objects and attributes
  - We can also use inclusive commands (include) to simplify the modularisation
- classes (objectClass) combine the attributes
  - described by ASN.1 record
  - They are part of hierarchy and they inherit properties of parent
  - specify mandatory and optional attributes

LDAP SCHEME, CLASS and ATRIBUTES

- attributes describes the properties
  - described by ASN.1 record
  - in a way, the definition of type
  - their realization (instance) will actually nutrient values
  - They describe the syntax, comparisons method, etc..

CLASSES

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

- case of class definitions

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

LDAP and DATA

- To transfer data between LDAP servers we have defined LDIF format:

```
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
```

SOFTWARE

- On FreeBSD / Linux OpenLDAP
- server and application programs:
  - slapd, slurpd
  - ldapcomapre, ldapdelete ...
- configuration files in / usr / local / etc
- More on exercises
  - challenge: install OpenLDAP on your server and configure it

SOFTWARE

- user programs may include the possibility of fetching data from the LDAP server
  - freeradius, authentication on unix-s ...