

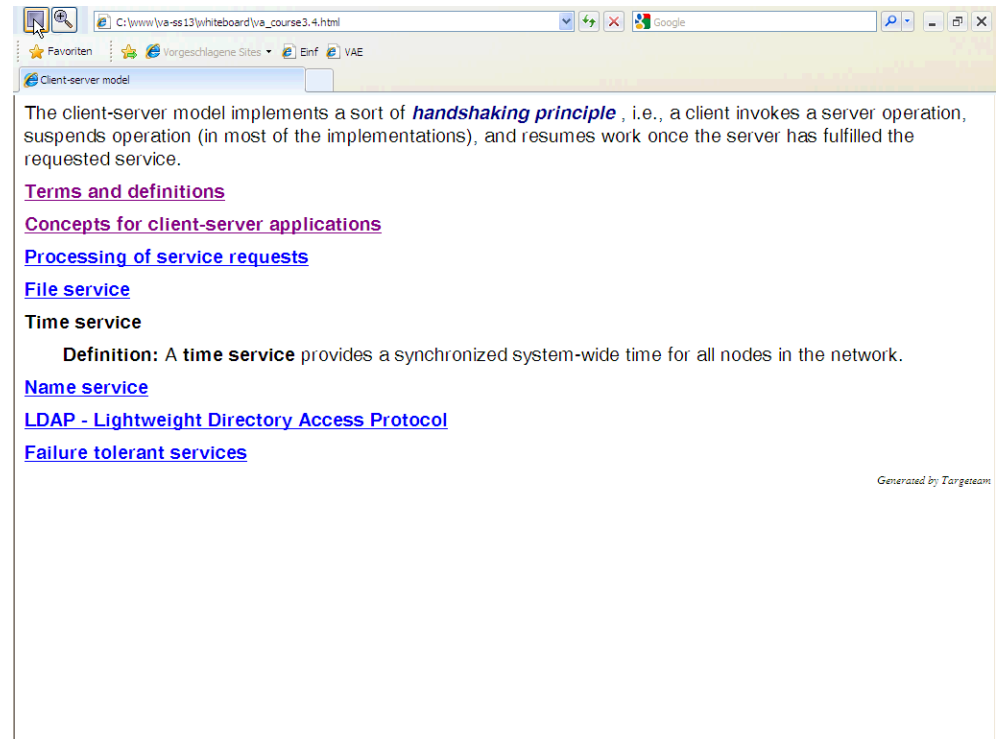
Script generated by TTT

Title: Distributed_Applications (07.05.2013)

Date: Tue May 07 14:30:16 CEST 2013

Duration: 89:05 min

Pages: 28



The client-server model implements a sort of *handshaking principle*, i.e., a client invokes a server operation, suspends operation (in most of the implementations), and resumes work once the server has fulfilled the requested service.

[Terms and definitions](#)

[Concepts for client-server applications](#)

[Processing of service requests](#)

[File service](#)

[Time service](#)

Definition: A **time service** provides a synchronized system-wide time for all nodes in the network.

[Name service](#)

[LDAP - Lightweight Directory Access Protocol](#)

[Failure tolerant services](#)

Generated by Targeteam



Processing of service requests



clients and servers have different life spans; servers manage these requests in a queue.

[Single dedicated server process](#)

[Cloning of new server processes](#)

Parallel request processing through threads

This is a variant of the second approach.

Shared address space, i.e. the approach allows shared utilization of variables;

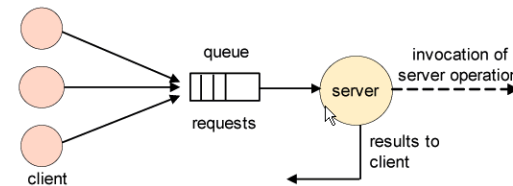
Generated by Targeteam



Single dedicated server process



A single dedicated server process is in charge of processing requests for service operations.



no parallel processing of requests, which results in the following disadvantage:

approach may be time consuming.

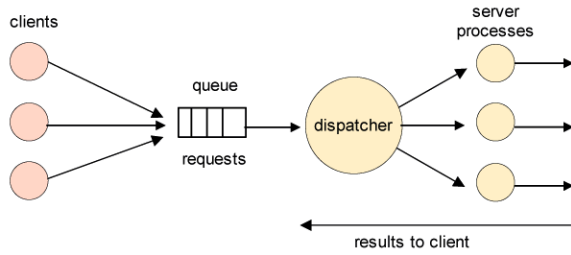
no interruption of the processing of the current request when a higher prioritized request appears in the queue.

server becomes bottleneck.

Generated by Targeteam



Every incoming request is handled by a new server process.



Cloning of new server processes is expensive;
Synchronization of access to shared persistent data;
Parallel processing of several applications is possible;



clients and servers have different life spans; servers manage these requests in a queue.

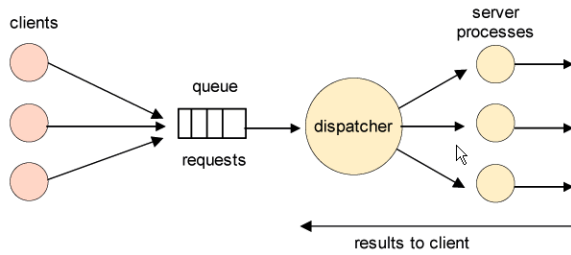
- [Single dedicated server process](#)
- [Cloning of new server processes](#)

Parallel request processing through threads

This is a variant of the second approach.
 Shared address space, i.e. the approach allows shared utilization of variables;



Every incoming request is handled by a new server process.



Cloning of new server processes is expensive;
Synchronization of access to shared persistent data;
Parallel processing of several applications is possible;



clients and servers have different life spans; servers manage these requests in a queue.

- [Single dedicated server process](#)
- [Cloning of new server processes](#)

Parallel request processing through threads

This is a variant of the second approach.
 Shared address space, i.e. the approach allows shared utilization of variables;



clients and servers have different life spans; servers manage these requests in a queue.

Single dedicated server process

Cloning of new server processes

Parallel request processing through threads

This is a variant of the second approach.

Shared address space, i.e. the approach allows shared utilization of variables;

Generated by Targeteam

Definition: A **file service** [Svobodova 1984] provides (remote) centralized data storage facilities to clients distributed among a network.

server deals with bulk data storage, high performance computation, collecting/managing large amounts of data.

client deals with "attractive" display, quick interaction times.

use of caching to speed up response time. Use of cache "hints" to facilitate cache management

speed up system when hint is correct.

mechanism to detect wrong hint and seek up-to-date information.

Distinction between stateless and stateful

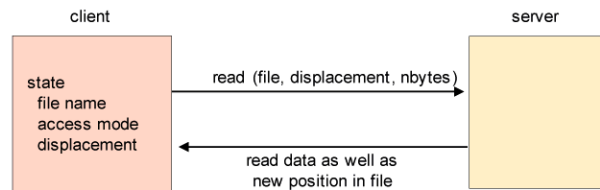
Stateless server

Stateful server

Generated by Targeteam



Stateless server do not manage any state information about their clients; the client must supply all necessary parameters to process the request.



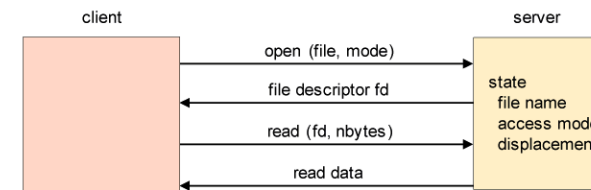
stateless server does not track clients or ensure that cached data stays up to date ⇒ cache refresh is responsibility of the client.

client uses often write-through caching policy.

A crashed server can be restarted without dealing with state reinstalls.

Generated by Targeteam

Stateful server subsystems manage state information about their clients.



server tracks its clients and takes actions to keep their cached states up-to-date. Client can trust its cached data ⇒ cache is owned by the server.

As a consequence, programming at the client site becomes less complex.

stateful transactional server architecture: after recovery of server crash an abort message is sent to client.

Generated by Targeteam



Client-server model



The client-server model implements a sort of *handshaking principle*, i.e., a client invokes a server operation, suspends operation (in most of the implementations), and resumes work once the server has fulfilled the requested service.

Terms and definitions

Concepts for client-server applications

Processing of service requests

File service

Time service

Definition: A **time service** provides a synchronized system-wide time for all nodes in the network.

Name service

LDAP - Lightweight Directory Access Protocol

Failure tolerant services

Generated by Targeteam



Name service



Definition: A **name service**, sometimes called a directory service, provides (remote) centralized name management facilities to clients distributed among a network; names refer to objects; examples are files, other servers, services, personal computers, printers, as well as users.

Name servers manage a list of names. Such a directory entry might be stored in a data structure

name	/* Name of the object as parameterized in a client request.*/
address	/* Address of the object within the network, e.g., host number concatenated with communication port number.*/
access information	/* This access information may limit access to the object for particular clients.*/
attributes	/* Additional attributes of the object.*/

Example for a Name Service

Generated by Targeteam



Example for a Name Service



Domain Name System (DNS):

hierarchical domain-based naming scheme for the Internet.

distributed database for implementing this naming scheme.

mapping of host names and email destinations (e.g. www11.in.tum) to their respective IP addresses.

top-level organizational domains:

edu: universities and other educational institutions

com: commercial organizations

de: organization in Germany

DNS database is distributed across a logical network of name servers.

Each server stores primarily data for the local domain.

Animation Domain Name Service

Generated by Targeteam



Animation Domain Name Service



Domain Name Service

Iterativ Rekursiv



Generated by Targeteam



Domain Name System (DNS) :

- hierarchical domain-based naming scheme for the Internet.
- distributed database for implementing this naming scheme.
- mapping of host names and email destinations (e.g. www11.in.tum) to their respective IP addresses.
- top-level organizational domains:
 - edu: universities and other educational institutions
 - com: commercial organizations
 - de: organization in Germany
- DNS database is distributed across a logical network of name servers.
- Each server stores primarily data for the local domain.

Animation Domain Name Service

Generated by Targeteam



LDAP is a **protocol** supporting the access to and update of directory information. It is an open industry standard.

LDAP is used by the **IntegraTUM** project to provide a university-wide directory service at TUM.

- [Basics](#)
- [LDAP architecture](#)
- [Information model](#)
- [Naming model](#)
- [Functional model](#)
- [Idif - exchange format](#)

Generated by Targeteam



Definition: A **directory** is a list of objects arranged in some order and with descriptive information (meta-data).

difference between directory and database

- directory has a high volume of read requests
- directories do not support transactions
- different query languages

A **directory service** is a name service containing object names and meta-data.

- Queries in directories: based on names **and** meta-data.
- White Pages: object access according to object name.
- Yellow Pages: object access according to object meta-data.
- LDAP is a communication protocol supporting access to / update of directory information.
 - it has been developed as simple alternative to X.500 standard.
 - it is based on TCP/IP rather than the ISO/OSI protocol stack.
 - modern web browsers (for example netscape) support LDAP.

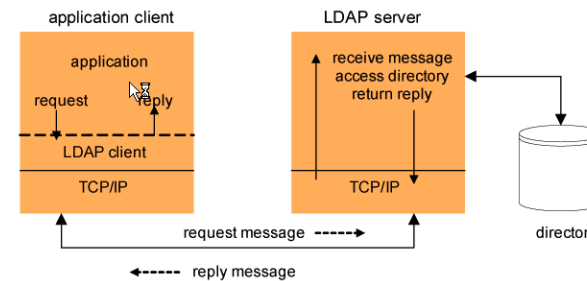
LDAP specifies several models

- information model: basic data structures
- naming model: referencing of objects (distinguished names)
- functional model: communication protocol and operations
- security model: control for directory access

Generated by Targeteam



The LDAP architecture is based on the client-server model and the TCP/IP protocols.



LDAP uses strings for data representation.

General interaction process

- Client initiates a session with the LDAP server (binding).
 - Client specifies a name or an IP address and port (e.g. port 389) of the LDAP server.
 - Client specifies user name and password.
- Client invokes LDAP operations (read, write, seek).
- Client terminates session (unbinding).

Generated by Targeteam



A directory entry describes an object, for example person, printer, server, organizations etc.
 each entry has a **distinguished name** (DN).
 each entry has a set of **attributes** with a type and one or several values.

[Attribute syntax](#)

[Examples](#)



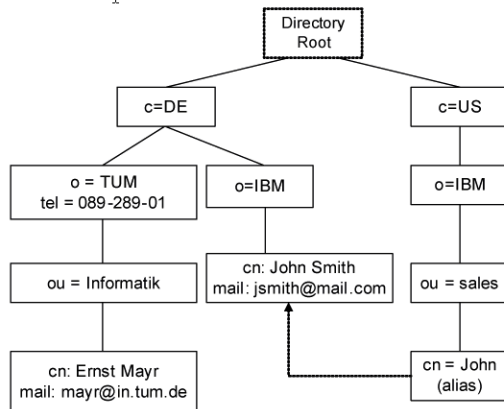
A directory entry describes an object, for example person, printer, server, organizations etc.
 each entry has a **distinguished name** (DN).
 each entry has a set of **attributes** with a type and one or several values.

[Attribute syntax](#)

[Examples](#)



The LDAP naming model defines how entries are identified and organized. Any **distinguished name** (DN) of an object consists of a sequence of parts, so-called **relative distinguished names** (RDN).
 The entries in an LDAP directory are hierarchically structured as tree (Directory Information Tree, DIT).
 Example of DN: cn=John Smith, o=IBM, c=DE.
 DIT also supports aliases.
 DIT can be distributed across several servers. Reference to entries of other LDAP servers via URLs.



LDAP is a **protocol** supporting the access to and update of directory information. It is an open industry standard.
 LDAP is used by the **IntegraTUM** project to provide a university-wide directory service at TUM.

[Basics](#)

[LDAP architecture](#)

[Information model](#)

[Naming model](#)

[Functional model](#)

[Ldif - exchange format](#)



The search operation allows a client to request that an LDAP server search through some portion of the DIT for information meeting user-specified criteria in order to read and list the result(s).

Examples

find the postal address for cn=John Smith,o=IBM,c=DE.

find all entries which are children of ou=Informatik,o=TUM,c=DE.

Search constraints.

base object: defines the starting point of the search. The base object is a node within the DIT.

scope: specifies how deep within the DIT to search from the base object, e.g.

baseObject: only the base object is examined.

singleLevel: only the immediate children of the base object are examined; the base object itself is not examined.

wholeSubtree: the base object and all of its descendants are examined.

filter: search filter on entry attributes; Boolean combination of attribute value assertions

example: (&(cn=schmi)*(!(c=de)))

Code example

Generated by Targeteam



```
#define SEARCHBASE "o=TUM,c=DE"
LDAP *ld;
char *User = NULL;
char *Passwd = NULL;
char searchfilter[] = "cn=Mayr";
/* open a connection */
if ((ld = ldap_open("ldapserver.in.tum.de", LDAP_PORT)) == NULL) exit(1);
/* authenticate as nobody */
if (ldap_simple_bind_s(ld, User, Passwd) != LDAP_SUCCESS) {
    ldap_perror(ld, "ldap_simple_bind_s");
    exit(1);
}
/* search the database */
if (ldap_search_s(ld, SEARCHBASE, LDAP_SCOPE_SUBTREE, searchfilter, NULL, 0) != LDAP_SUCCESS) {
    ldap_perror(ld, "ldap_search_s");
    exit(1);
}
.....
/* close and free connection resources */
ldap_unbind(ld);
```



ldif = LDAP Data Interchange Format; it is used to import and export directory information.

```
dn: cn=Informatik
cn: Informatik
objectclass: top
objectclass: groupOfNames
member: cn=Baumgarten,Uwe, mail=baumgaru@in.tum.de
member: cn=Schlichter,Johann, mail=schlicht@in.tum.de
....
dn: cn=Baumgarten,Uwe, mail=baumgaru@in.tum.de
cn: Baumgarten,Uwe
modifytimestamp: 20001213084405Z
mail: baumgaru@informatik.tu-muenchen.de
givenname: Uwe
sn: Baumgarten
objectclass: top
objectclass: person
....
dn: cn=Schlichter,Johann, mail=schlicht@in.tum.de
cn: Schlichter, Johann
modifytimestamp: 20001213084406Z
mail: schlicht@in.tum.de
```

The screenshot shows a Windows Internet Explorer browser window with the address bar set to "C:\www[va-ss13\whiteboard\va_course3.4.6.6.html". The page title is "ldif - exchange format". The content of the page is identical to the text shown in the left panel, including the definition of ldif and the example LDIF entries for 'Informatik' and 'Baumgarten,Uwe'.

ingTool - Version 25.10.2009

Student Teacher Post Processing Extras

Help

Start

Idf - exchange forma... TeleTeachingTool - V...

15:59