CS 580 Client-Server Programming
Fall Semester, 2000
Doc 17 Types of Servers
Contents

Types Of Servers 2
Iterative verses Concurrent.....3
Stateless verses Stateful Servers 5
Modes of Operation 6
POP3 Protocol.....7
AUTHORIZATION.....9
TRANSACTION 10
UPDATE State 12
Optional POP3 Commands.....13

References

Internetworking with TCP/IP, BSD Socket Version Vol. 3, Comer, Stevens, Prentice-Hall, 1993

RFC 1939 Post Office Protocol - Version 3, Myers, Rose, 1996, http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1939.txt

Copyright ©, All rights reserved.
2000 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA.
OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.

Types of Servers

- Connectionless(UDP) verse Connection-Oriented (TCP)
- Iterative verses Concurrent
- Stateless verse stateful

Iterative verses Concurrent

Iterative

- Single process
- Handles requests one at a time
- Good for low volume & requests that are easy to satisfy

Concurrent

- Handle multiple requests concurrently
- Normally uses thread/processes
- Needed for high volume & complex requests
- Harder to implement than iterative
- Must deal with currency
  - Create threads/processes
  - Handle deadlock/livelock issues
  - How to detect a "bad" thread/process

Single Process/Thread Concurrent Server

One can implement a concurrent server using one thread/process

```
while (true) {
   check if any new connects (non-block accept)
   if new connection accept
   process a little on each current request
}
```

Stateless verses Stateful Servers

State information

Information maintained by server about ongoing interactions with clients

Stateless server

Server that does not maintain state information

Stateful server

Server that does maintain state information

State information cause problems

Consumes resources

How long does one maintain the state?

Modes of Operation

Stateful servers sometimes have different modes of operation

Each mode has a set of legal commands

In Login mode only the commands password & username are accepable

After successful login client-server connection in transaction mode

In transaction mode command X, Y Z are legal

These modes are also called server states or just states

POP3 Protocol

Purpose: Allow PC's, Macs, etc. to download mail from server

Port number 110

Protocol uses ASCII only

Command to server

keyword blank argument1 [blank argumentk] CRLF

| keyword | = 3, 4 characters

| argument | <= 40 characters

keyword and arguments are separated by single space character

Server Response

Status keyword additionalInfo

Status is either "+OK" or "-ERR"

A single line response ends in CRLF

If response requires more than one line:

Each line ends in a CRLF

The response ends in CRLF.CRLF

If a line starts with a "." prepend a "." to it

A POP3 server may have an autologout timer

A server must wait at least 10 minutes before timing out a client

The POP3 server on saturn times out in 2 minutes

States

AUTHORIZATION

Must log in with password before entering the transaction state

TRANSACTION

Client can request actions of server, get mail for example

UPDATE

Updates mail box to reflect actions taken in the transaction state

AUTHORIZATION

Server acknowledges connection from client with

+OK "message"

+OK UCB Pop server (version 2.1.2-R3) at saturn starting.

Commands: USER, PASS, APOP, QUIT

USER PASS

Combination of commands are used to progress to the transaction state

USER must come first
PASS or QUIT must come after USER

Example

rohan 14-> **telnet saturn 110**
Trying 130.191.229.1...
Connected to saturn.sdsu.edu.
Escape character is '^]'.
+OK UCB Pop server (version 2.1.2-R3) at saturn starting.
USER whitney
+OK Password required for whitney.
PASS don'tyouwish
+OK whitney has 21 message(s) (44285 octets).

TRANSACTION

Commands: STAT, LIST, RETR, RSET, QUIT

STAT

Arguments: none

Returns "+OK" numberOfMessages SizeOfMail

Example

STAT
+OK 22 45595

LIST

Arguments: a message-number (optional)

Returns: size of message in octets

Examples:

LIST 2
+OK 2 3064

LIST
+OK 22 messages (45595 octets)
1 2980
2 3064 (message 3 - 21 deleted to save space)
22 1290
.

RETR

Arguments: a message-number

Returns: the message

Example:

RETR 21
+OK 825 octets
Received: from [130.191.9.18] (ebb2p9.sdsu.edu [130.191.9.18]) by saturn.sdsu.ed
u (4.1/8.6.10) with SMTP id UAA29486 for <whitney@saturn.sdsu.edu>; Mon, 11 Mar
1996 20:16:07 -0800 (PST)
X-Sender: whitney@cs.sdsu.edu (Unverified)
Message-Id: <v02110100ad6aaaf097b6@[130.191.9.70]>
Mime-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Date: Mon, 11 Mar 1996 20:16:50 -0800
To: whitney@saturn.sdsu.edu
From: whitney@saturn.sdsu.edu (Roger Whitney)
Subject: Sample Mail
X-UIDL: 826604201.000

this is a test
..
the end

Roger Whitney Mathematical Science Department
whitney@cs.sdsu.edu San Diego State University
http://cs.sdsu.edu:8080/~whitney San Diego, CA 92182-
7720
(619) 594-3535
(619) 594-6746 (fax)

.

DELE

Arguments: a message-number to delete

Returns: a confirmation of deletion

Marks a message to be deleted

NOOP

Arguments: none

Returns: a positive response

Does nothing

QUIT

Arguments: none

Returns: a positive response

Send POP3 server to UPDATE state

UPDATE State

Updates mail box to reflect transactions taken during the transaction state, then logs user out

If session ends by any method except the QUIT command during the transaction state, the update state is not entered

Optional POP3 Commands
TOP

Arguments: a message-number

Returns: Top 10 lines of indicated message

State allowed in: transaction

UIDL

Arguments: a message-number (optional)

Returns: a unique-id listing for message

State allowed in: transaction

Examples:

UIDL 1
+OK 1 826312760.001

UIDL
+OK uidl command accepted.
1 826312760.001
2 826312760.006
3 826493796.004
4 826510843.001
5 826510843.002
6 826576073.000
7 826594928.004
8 826603475.000
9 826648617.001
10 826648617.002
.

APOP

Arguments: a mailbox and a MD5 digest string

State allowed in: authorization

Action: If MD5 string is correct move to transaction state