POP3 (Post Office Protocol version 3) Implementation

SCENIX

Application Note 28
Abraham Si
December 1999
Patents Pending

1.0 Introduction

This application note describes an implementation of the POP3 (Post Office Protocol version 3) per RFC1725 (Meyes & Rose, 1994).

POP3 is the most commonly used protocol for retrieving email across the Internet. It is, like SMTP, a client server protocol, the client usually being an email program residing on a PC or in an embedded system, and the server is the POP3 server in an intranet or a POP3 server in the ISP premise. The SX implementation is a POP3 client, thus enabling the retrieval of email from any POP3 compliant server.

The POP3 implementation requires the transmission control protocol (TCP) and the TCP/IP stack described in application notes AN27 (TCP Virtual Peripheral Implementation) and AN23 (UDP/PPP Virtual Peripheral Implementation). This particular implementation uses the SX52BD communications controller.

2.0 POP3 Demonstration

Post Office Protocol version 3 (POP3) is the standard protocol for retrieving email sent from another TCP/IP capable device. It is a text-based protocol designed around a simple command-response language. POP3 is described in RFC 1725. The following example illustrates how POP3 works. The commands sent from the SX are in bold, the replies from the POP3 server are in italics:

Accepting POP connection from [192.168.11.1]
+OK scenix.com POP service ready
USER khsi
+OK khsi... Recipient ok
PASS ******
+OK khsi's mailbox has 1 total messages (1166 octets).
STAT
+OK 1 1166
RETR 1
Sending <C:\MDAEMON\USERS\ASi\MD00001.MSG> to [192.168.11.1]
Transfer Complete.
DELE 1
+OK message 1 deleted
QUIT
+OK khsi scenix.com POP Server signing off (mailbox empty)
POP session complete, 1171 bytes transferred!

With the debug port connected to another PC using the Hyperterminal program at 57,600 bps, one can see the message:

```
Received: from eSXDemo [127.0.0.1] by
scenix.com [192.168.11.2]
   with SMTP (MDaemon.v2.84.T)
   for <khsi@scenix.com>; Thu, 21 Oct 1999
   10:13:00 -0700
From: "eSX" <eSX@scenix.com>
To: "kh si" <khsi@scenix.com>
Subject: test of POP3
Date: Thu, 21 Oct 1999 10:12:56 -0700
X-MSMail-Priority: Normal
X-Priority: 3
X-Mailer: Microsoft Internet Mail 4.70.1155
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--
---=_NextPart_000_01BF0419.DFCCD5E0"
Content-Transfer-Encoding: 7bit
X-MDaemon-Deliver-To: khsi@scenix.com
X-Return-Path: eSX@scenix.com

This is a multi-part message in MIME format.

------=_NextPart_000_01BF0419.DFCCD5E0
Content-Type: text/plain; charset=ISO-8859-
1
Content-Transfer-Encoding: 7bit

POP POP POP
------=_NextPart_000_01BF0419.DFCCD5E0
Content-Type: application/octet-stream;
name="autoexec.bat"
Content-Transfer-Encoding: quoted-printable
Content-Description: autoexec.bat (MS-DOS
Batch File)
```

```
Content-Disposition: attachment;
filename="autoexec.bat"

@ECHO OFF
SET =
PATH=3D%PATH%;C:\PROGRA~1\BORLAND\CBUILD~1\
BIN;C:\PROGRA~1\BORLAND\CBUILD=~1\PROJECTS\
BPL

------=_NextPart_000_01BF0419.DFCCD5E0--
```

Note that the main contents of the email are highlighted in bold font. This particular email also has an attachment to demonstrate a more complex scenario.

The connection is initiated by the client, in this case the SX device. Each message from the server starts with a message indicating the status (either +OK or -ERR). The number following OK is used to indicate the number of messages after the STAT command is issued. If the SX receives any reply codes that it is not expecting, it aborts the transaction.

In this sample implementation, everytime a PPP connection is established, the SX will try to retrieve all the emails in its mailbox. After retrieval, the emails will all be deleted from the server.

3.0 Implementation

So that the data can be transferred in segments larger than the available RAM, the POP3 implementation uses the event driven architecture described in application note AN27 (TCP/IP Virtual Pripheral Implementation). The emails received are not buffered, but rather pumped out on the debug port. For an actual system, one would put them in RAM for storage purposes.

A state machine is implemented in software to track all the transactions. A jump table in the AppPacketOK routine decides what to send in reply.The state machine will toggle between retrieve and delete states until all messages are retrieved and deleted.

The transmit routines use counters to step through each canned packet which is stored in program memory. The message number of the email to be retrieved is internally tracked, converted to ASCII and send out with the RETR and DELE commands. A dymamic replacement is done in the AppTxByte routine for this purpose.

Lit#: SXL-AN28-03

For the latest contact and support information on SX devices, please visit the Scenix Semiconductor website at www.scenix.com. The site contains technical literature, local sales contacts, tech support and many other features.

SCENIX

1330 Charleston Road
Mountain View, CA 94043
Tel: (650) 210-1500
Fax: (650) 210-8715
http://www.scenix.com