

组织: 中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者: lou_goodman (lou_goodman lou_oxygen@163.com)

译文发布时间: 2001-5-8

版权: 本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载, 但必须保留本文档的翻译及版权信息。

Network Working Group
Request for Comments: 937

M. Butler
J. Postel
D. Chase
J. Goldberger
J. K. Reynolds

Obsoletes: RFC 918

ISI
February 1985

邮局协议 (版本 2)

(RFC 937 POST OFFICE PROTOCOL - VERSION 2)

一: 提要:

这篇 RFC 文档提供了一种使工作站动态的从邮件服务器获取邮件的简单方法.它着重阐述了符合 ARPA 标准的因特网的邮件协议,并为它的进一步发展提供了建议和讨论.它是 RFC 918 的更新.你可以自由传播这个文档.

二: 引言

邮局协议(版本 2)的目的是为了让用户的工作站从邮件服务器获取邮件.它也应该允许邮件从工作站通过简单邮件传递协议(SMTP)发送到邮件服务器.更多内容请参考 POP2 821[1] 和 POP2 822[2].

这个协议假定已经存在了一个可靠的数据流,比如由 TCP 协议或其它协议提供的数据流.如果是 TCP 协议,则 POP2 协议服务器从 109 端口进行监听.

三: 系统模型和分析

当我们给一工作站分配一个 IP 地址从而把它当作一个因特网主机时,我们并不希望它包含用户的邮箱.我们希望有一个专门的有服务器来提供邮箱服务.

当然我们知道提供这种服务的机器的最重要的要求是它要健壮.一台经常宕机的机器是

不合作邮件服务器的。

POP2 协议是为局域网(比如 Ethernets)上工作站和服务器之间的短时延,高传输率的通讯而设计的。POP2 协议也可以工作在其它类型的网络上。但如果网络环境有了很大的改变,一个新的协议可能是必要的。

设想有一个用户叫 John Smith, 它的计算机叫 FIDO, 它的邮件服务器叫 DOG-HOUSE。则它的邮件地址是 J S m i t h @ D O G - H O U S E . A R P A (而非 J S m i t h @ F I D O . A R P A)

这也就是说, 邮件是在邮件服务器的邮箱中存放。!!! 协议只不过提供了一种浏览邮箱中邮件的机制。

用户的工作站和服务器之间没有任何的关系, 用户的工作站也不作为邮件地址的一部分。

这是一个非常简单的协议。它不是一个用户界面。我们希望我们的计算机上有一个友好的程序。但事实上, 该协议一点也不友好。它的一个基本规则是: 如果存在任何错误就断开连接。

POP2 协议根本不对消息做任何处理, 它也部分分析消息头 (Date:, From:, To:, Cc:, or Subject:). POP2 协议之间的将消息从服务器传到客户工作站。

四: POP2 协议

POP2 协议是一系列命令和回答的集合。它来自 A R P A 网络的其它许多已经存在的协议。

服务器必须监听一个连接。当一个连接被建立时, 服务器发送一个 `g r e e t i n g` 消息并等待用户的其它命令。当命令到达时, 服务器进行处理, 并给与用户答复。

客户打开一个连接, 等待 `g r e e t i n g` 消息, 然后发送 `H E L O` 命令, 包括用户的姓名和密码的参数给服务器进行认证从而获取邮件。服务器返回默认邮箱的邮件数目给用户。

用户可以阅读默认邮箱的邮件, 也可以用 `F O L D` 命令来选择邮箱。同样, 服务器将返回用户所选择的邮箱的邮件数。

用户用 `READ` 命令开始阅读邮件。READ 命令应该包含一个参数, 用来指定要阅读的邮件序号, 默认值是当前邮件(即一个新文件夹打开时的第一条邮件)。然后, 服务器开始向用户发送邮件的内容。

用户可以用 `RETR` 命令来请求获得消息的目录。

当所有的数据都被接收时, 用户发送确认消息。这就是 `ACKS`, `ACKD` 和 `NACK`。

`ACKS` 意味着“我已经成功的收到了消息, 请将它保存在邮箱中”。

`ACKD` 意味着“我已经成功的收到了消息, 请将它从邮箱中删除”。

`NACK` 意味着“我没有受到消息, 请将它保存在邮箱中”。

接收到 `ACKS` 和 `ACKD` 命令后, 服务器将消息指针指向下一条消息; 接收到 `NACK`

命令后, 消息指针保持不变.

所有情况下, 服务器返回当前消息的内容.

用户用 QUIT 命令关闭一个对话, 服务器返回一个消息--- OK.

正常的对话过程如下表示:

用户	服务器
	等待连接
打开连接 -->	
	<-- 服务器进入就绪状态, 等待命令
HELO 命令和账号. 密码 -->	
	<-- 发送 #13 消息, 等待命令
读取#13 消息 -->	
	<-- =537 characters in that message
	等待命令
RETR 命令 -->	
	<-- (发送数据)
	等待命令
ACKS 消息 -->	
	<-- =0 数据传输完毕
	Wait for Command
QUIT -->	
	<-- + OK
关闭连接 -->	<-- 关闭连接
	等待连接 (回到初始状态)

参数

下面的参数有确定的含义:

user - 用户账号

password - 账号密码

mailbox - 邮箱名 (或文件夹名)

默认邮箱

TOPS-20 系统

MAIL.TXT.1 - from login directory

UNIX 系统

/usr/spool/mail/user

和

/usr/user/Mail/inbox/*

“user” 是由 HELO 命令提供的参数。

文件结束标志

文件结束标志是由一个回车符 (CR) 和换行符 (LF) 表示的。在文件中用“CRLF”表示。这个标志在命令和回答中都会用到。

消息长度

对 READ 命令和确认命令 (ACKS, ACKD, NACK) 的回答是下一个要传递的消息的长度, 包括所有消息字符。CRLF 被认为是两个字符。长度为零的消息意味着消息不存在或为空。请求传递一个零长度的消息意味着关闭连接。这是在因特网标准 RFC 822 和 NVT-ASCII 中定义的。

邮件数

对 HELO 和 FOLD 命令的回答是被选邮箱中邮件的数目的统计。READ 命令需要一个参数来指定要阅读的邮件。所有的数都用十进制来表示, 并且从一开始。这意味着邮箱中第一条邮件用“1”来表示。

这篇文档中的所有数字都用十进制来表示。

转义引用

一些情况下, 参数中需要用特殊的字符, 二是语法所不允许的。例如: 密码中的一个空格。为了做到这一点, 我们需要用转义会话。不幸的是, 我们不得不采用让人讨厌的字符。在本协议中, 我们用反斜杠“\”。参数中的一个空格字符需要用两个字符来表示“\空格”。这种转义引用只用在命令参数中, 回答 RETR 命令的邮件数据不需要这种方法。

回答串

第一个字符必须时特别指定的 (如 “+”, “-”, “=”, “#”)。紧跟的选项串可以是任意值。

命令和回答的定义

命令和回答总汇

命令	回答
HELO user password	+ OK
FOLD mailbox	- Error
READ [n]	#xxx

RETR =yyy
 ACKS
 ACKD
 NACK
 QUIT

命令解释

HELO user password

HELO 命令用来认证用户. 它的信息被服务器用来确定用户对邮箱的使用权. 该命令的关键字是 HELO , 后面是用户的账号和密码, 然后是 CRLF.

可能的应答:

"#nnn" nnn 是默认邮箱的邮件数.

"- error report" 错误报告并断开连接.

FOLD mailbox

该命令选择一个邮箱或一个文件夹. 服务器必须判断用户是否有权限访问该邮箱或文件夹. 如果邮箱为空或不存在, 消息数为 0. 该命令的关键字是 "FOLD", 后面跟 mailbox 参数, 然后是 CRLF.

可能的应答:

"#nnn" nnn 是邮箱中邮件总数.

READ [nnn]

READ 命令开始传递数据. 如果 READ 命令中未给参数, 意味着读取当前邮件. 一个 READ 命令首先设置邮件指针为参数值, 然后服务器返回邮件长度. 如果没有邮件可读, 返回的邮件长度等于零. 如果邮件已经被 ACKD 命令 删除, 也返回零长度. 跟在 READ 命令后的是 RETR 命令, READ 命令, FOLD 命令或 QUIT 命令. 不要企图用 RETR 命令来操作一个零长度的邮件. 该命令的关键字是 "READ". 接在后面的是邮件数参数和 CRLF.

可能的应答:

"=ccc" ccc 是对应的邮件的长度

RETR

该命令确定用户已经做好接收邮件的准备. 它必须跟一个确认命令. 服务器会断开连接, 如果发送一个长度为零的消息 (例如一个不存在的消息). 该命令的关键字是 "RETR", 后面跟一个 CRLF.

可能的应答:

数据流或关闭连接

ACKS

该命令确认用户已经接收到邮件. 该命令结束邮件数据的传输. 邮件数据被保存在邮箱中, 邮件指针增加 1. 服务器返回下一个邮件数据长度或零 (如果下一个邮件不存在或已经被删除). 该命令的关键字是 "ACKS", 后跟 CRLF

可能的应答:

"=ccc" ccc 是下一个邮件的长度

ACKD

该命令确认用户已经接收到邮件, 结束邮件数据的传输, 如果用户有对邮箱的

写权限则将对应的邮件从邮箱中删除。事实上,只有在会话结束,邮箱被释放或用户选择另一个邮箱时才会发生改变。如果用户没有对邮箱的写权限,则邮箱不发生改变。服务器的应答是相同,不管邮件是否被删除。然后,服务器将邮件指针指向下一个邮件,并返回它的长度。相应的,如果邮件不存在或已经被删除,则返回零长度。该命令的关键字是“ACKD”然后是 CRLF。

可能的应答:

“=ccc” ccc 是下一个邮件的长度

NACK

用户用该命令向服务器报告它未收到数据。该命令结束邮件数据的传送。将邮件数据保存在邮箱中,保持邮件指针不变。服务器返回当前邮件的长度。因为当前邮件必然存在,所以邮件长度是一个大于零的数。该命令的关键字是“NACK”,然后是 CRLF。

可能的应答:

“=ccc” ccc 是当前邮件的长度。

QUIT

该命令表明用户已经完成会话。服务器将发送一个“OK”应答给用户,并关闭连接。该命令的关键字是“QUIT”,然后是 CRLF。

可能的应答:

“+ OK” 并关闭连接

服务器回答

GREETING

任何一个连接被建立时,服务器都将发送该应答给用户。该消息由一个加号,然后是协议的名字(POP2),服务器的名字,文本, CRLF 组成。

“+”

加号意味着成功的连接。后面跟文本和 CRLF。

“-”

减号意味着连接失败。和加号一样,后面跟文本和 CRLF。

“=”

等号返回命令中指定的邮件的长度。后面跟一个数字,然后是文本和 CRLF。

“#”

该符号返回指定邮箱或文件夹中的邮件总数。后面跟一个数字,然后是文本和 CRLF。

TIMEOUTS

任何一个类似的协议中都有 TIMEOUTS。如果不想让其中一方陷入无穷的等待中。用户希望较快的应答,因此应该由一个短的超时时间。它被称作 T1。

对一些服务器来说,应该有时间计算邮箱中的邮件数或邮件的长度或重整存储的数据以用来传输。因此,超时时间应该大于这些时间。在命令 RETR 中也应该考虑这个问题。

题.

服务企业不希望无期地等待用户的下一个命令, 因此服务企业由一个超时时间 T2.

当一个用户给客户机编一段程序, 使它自动的给服务器发送命令时, 可以很合理的想象, 此时服务器的 T2 时间应该很小.

相应的, 如果是一个人在给服务器发送命令, 服务器的 T2 时间应该很大.

讨论

TOPS-20 上的服务器的评价:

这种情况下, 邮箱是一个简单的文件. 信邮件被链接在文件后面. 不同邮件间有一个分割行.

这种服务器的特别之处是它的删除操作. 只有当用户拥有对文件的写权限时, 才能进行该操作. 要注意避免两个问题: (1) 阻止其它用户对文件的写操作; (2) 防止意外删除用户为看的邮件.

一个建议

当一个邮箱被选中时, 如果用户有写权限, 给邮箱重新起一个临时名. 这样新邮件就会被放在临时文件中. 对临时邮箱进行所有的 POP2 操作. 这样当 POP2 会话结束或大卡另一个邮箱时, 任何为删除的邮件会存在信邮箱文件夹中.

命令行和回答行的大小

命令行的最大长度时 512 字符(包括命令字和 CRLF).

回答行的最大长度时 512 字符(包括+, -, =, #和 CRLF).

邮件数据行的最大长度时 1000 字符(包括 CRLF).

ISI 已经将 POP2 协议作为服务器用于 TOPS-20 和 Berkeley 4.2 Unix 系统; 作为客户机用于 IBM-PC 和 Berkeley 4.2 Unix 系统.

不足之处

POP2 协议不检查数据的内容, 甚至不分析数据头.

服务器中部存在邮件的任何状态信息(例如: 我们不从一个会话到另一个都发生了什么)这样服务器不知道那一个邮件是新邮件.

五: 例子

例子 1:

```

Client          Server
-----          -----
                                Wait for connection
Open connection -->
                                <-- + POP2 USC-ISIF.ARPA Server
HELO POSTEL SECRET -->
                                <-- #2 messages in your mailbox
READ            -->
                                <-- =537 characters in message 1
RETR            -->
                                <-- [data of message 1]
ACKD            -->
                                <-- =234 characters in message 2
    RETR        -->
                                <-- =234 characters in message 2
RETR            -->
                                <-- [data of message 2]
ACKD            -->
                                <-- =0 no more messages
QUIT           -->
                                <-- + OK, bye, bye
Close connection --> <-- Close connection
                                Go back to start

```

例子 2:

```

Client          Server
-----          -----
                                Wait for connection
Open connection -->
                                <-- + POP2 ISI-VAXA.ARPA server here
HELO smith secret -->
                                <-- #35 messages
FOLD /usr/spool/mail/smith -->
                                <-- #27 messages
READ 27         -->
                                <-- =10123 characters in that message
RETR            -->
                                <-- [data of message 27]
ACKS            -->
                                <-- =0 no more messages
QUIT           -->
                                <-- + bye, call again sometime.
Close connection --> <-- Close connection

```

Go back to start

例子 3:

Client	Server
-----	-----
	Wait for connection
Open connection -->	
	<-- + POP2 ISI-VAXA.ARPA server here
HELO Jones secret -->	
	<-- #0 messages
READ -->	
	<-- Close connection
Close connection -->	
	Go back to start

六：形式语法

1. 选项及参数

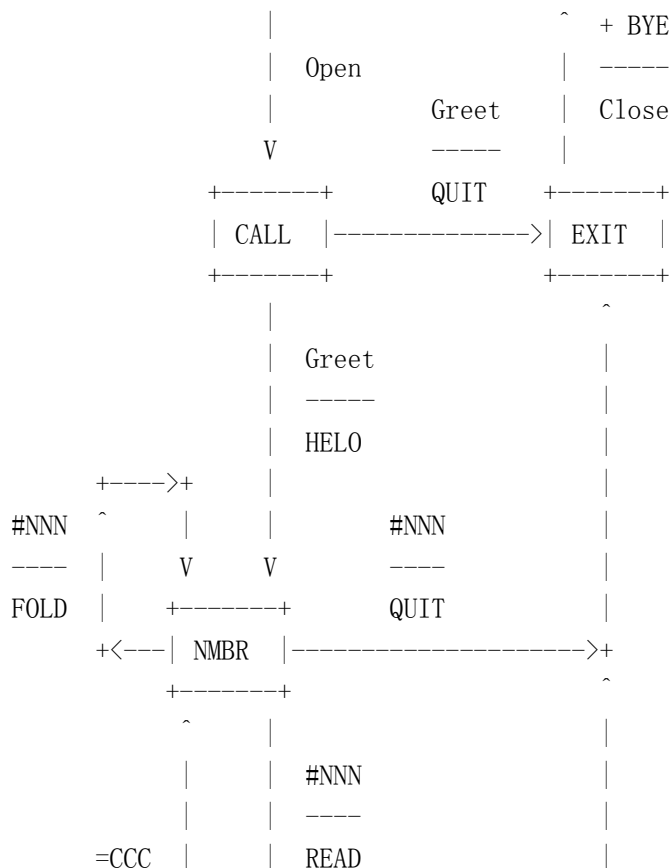
<digit> = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
 <letter> = A | B | C | ... | Z
 a | b | c | ... | z
 <punct> = ! | " | # | \$ | % | & | ' | (|) | * |
 + | , | - | / | : | < | = | > | ? | @ |
 [|] | ^ | _ | ` | { | | | } | ~
 <quote> = \
 <any> = any one of the 128 ASCII codes
 <CR> = carriage return, code 10
 <LF> = line feed, code 13
 <SP> = space, code 32
 <CRLF> = <CR> <LF>
 <print> = <letter> | <digit> | <punct> | <quote> <any>
 <char> = <print> | <SP>
 <word> = <print> | <print> <word>
 <string> = <char> | <char> <string>
 <ld> = <letter> | <digit>
 <ldh> = <letter> | <digit> | -
 <ldhs> = <ldh> | <ldh> <ldhs>
 <name> = <letter> [[<ldhs>] <ld>]
 <host> = <name> | <name> . <host>
 <user> = <word>

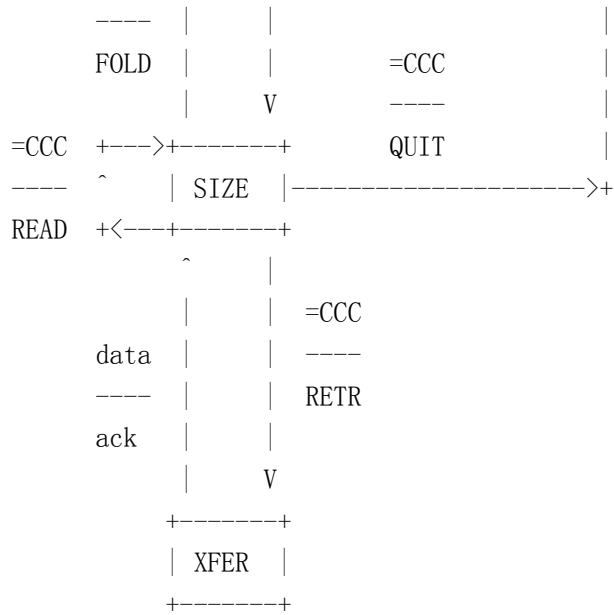
```

<password> = <word>
<mailbox>  = <string>
<number>   = <digit> | <digit> <number>
<helo>     = HELO <SP> <user> <SP> <password> <CRLF>
<fold>     = FOLD <SP> <mailbox> <CRLF>
<read>     = READ [<SP> <number>] <CRLF>
<retr>     = RETR <CRLF>
<acks>     = ACKS <CRLF>
<ackd>     = ACKD <CRLF>
<nack>     = NACK <CRLF>
<quit>     = QUIT <CRLF>
<ok>       = + [<SP> <string>] <CRLF>
<err>      = - [<SP> <string>] <CRLF>
<count>    = # <number> [<SP> <string>] <CRLF>
<greet>    = + <SP> POP2 <SP> <host> [<SP> <string>] <CRLF>
<length>   = = <number> [<SP> <string>] <CRLF>
<command>  = <helo> | <fold> | <read> | <retr> |
             <acks> | <ackd> | <nack> | <quit>
<reply>    = <ok> | <err> | <count> | <length> | <greet>

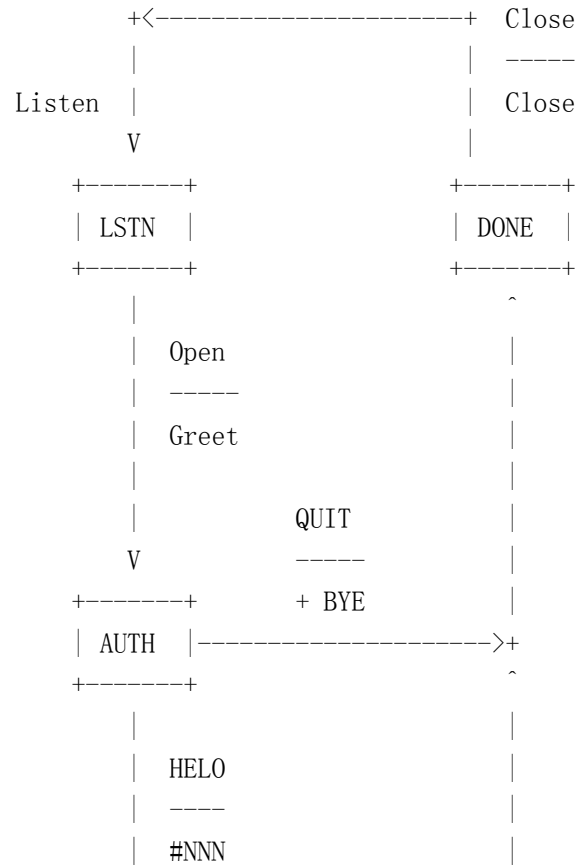
```

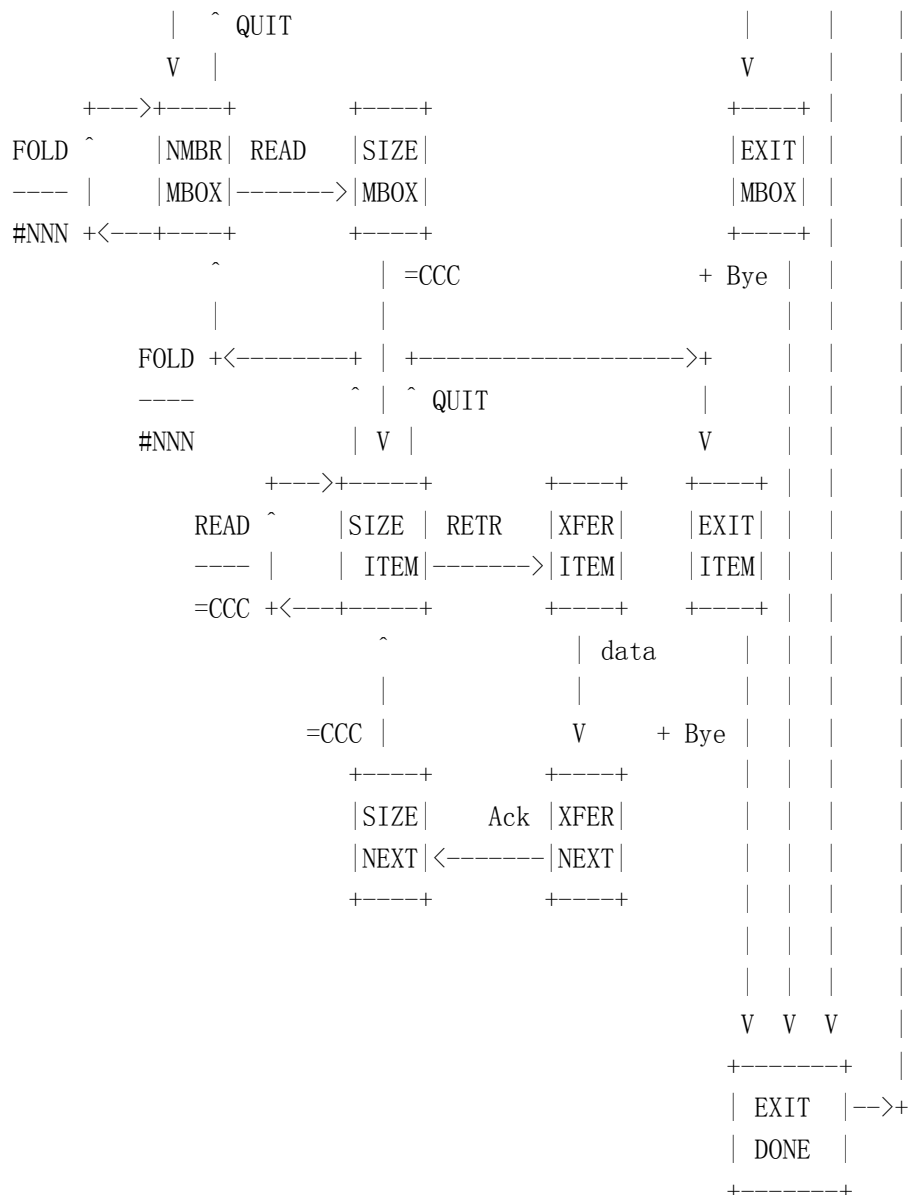
2. 用户流程图





3. 服务器流程图





5. 用户策略表

	STATE					
INPUT	CALL	NMBR	SIZE	XFER	EXIT	
Greet	2	1	1	1	6	
#NNN	1	3	1	1	6	
=CCC	1	1	4	1	6	

data	1	1	1	5	6
+ Bye	1	1	1	1	6
Close	1	1	1	1	6
other	1	1	1	1	6
Timeout	1	1	1	1	6

用户行为

1. 这是一个无用信息. 发送"QUIT"命令, 进入 EXIT 状态.
2. (a) 如果 greeting 成功, 发送"HELO", 进入 NMBR 状态.
(b) 否则, 发送"QUIT", 进入 EXIT 状态.
3. 如果用户选定文件夹且 NNN>0, 发送"READ", 并进入 SIZE 状态.
如果用户选定文件夹且 NNN=0, 发送"QUIT", 并进入 EXIT 状态.
如果用户重新选定文件夹, 发送"FOLD", 并返回 NMBR 状态.
4. 如果用户选定邮件且 CCC>0, 发送"RETR", 并进入 XFER 状态.
如果用户选定邮件且 CCC=0, 发送"QUIT", 并进入 EXIT 状态.
如果用户重新选定邮件, 发送"READ", 并进入 SIZE 状态.
5. 如果用户想保存邮件, 发送"ACKS", 并进入 SIZE 状态.
如果用户想删除邮件, 发送"ACKD", 并进入 SIZE 状态.
如果用户想重发邮件, 发送"NACK", 并进入 SIZE 状态.
6. 关闭连接.

6. 服务器策略表

	STATE						
	INPUT	LSTN	AUTH	MBOX	ITEM	NEXT	DONE
Open	2	1	1	1	1	1	1
HELO	1	3	1	1	1	1	1
FOLD	1	1	5	5	1	1	1
READ	1	1	6	6	1	1	1
RETR	1	1	1	7	1	1	1
ACKS	1	1	1	1	8	1	1

ACKD	1	1	1	1	8	1
NACK	1	1	1	1	8	1
QUIT	1	4	4	4	1	1
Close	1	1	1	1	1	9
other	1	1	1	1	1	1
Timeout		1	1	1	1	1

服务器行为

1. 这是一个无用信息. 发送“-error”, 并断开连接.
2. 发送 greeting , 进入 AUTH 状态.
3. 如果是认证用户, 发送“#nnn”, 进入 M B O X 状态.
否则, 发送“-error”, 并断开连接.
4. 发送 “+Bye”, 进入 D O N E 状态.
5. 发送“+NNN”, 进入 MBOX 状态.
6. 发送“=CCC”, 进入 ITEM 状态.
7. 如果邮件存在, 传输数据, 进入 NEXT 状态. 否则, 断开连接.
8. 回答命令 ACKS/ACKD/NACK, 进入 ITEM 状态.
9. 断开连接.

七: 感谢

非常感谢 RFC 918 的邮局协议的第一版和兴趣小组提供的 POP2 协议的草案.

八: 参考文献:

- [1] Postel, J., “Simple Mail Transfer Protocol”, RFC 821, USC/Information Sciences Institute, August 1982.
- [2] Crocker, D., “Standard for the Format of ARPA-Internet Text Messages”, RFC 822, University of Delaware, August 1982.
- [3] Reynolds, J.K., “Post Office Protocol”, RFC 918, USC/Information Sciences Institute, October 1984.
- [4] Reynolds, J.K., and J. Postel, “Assigned Numbers”, RFC 923, USC/Information Sciences Institute, October 1984.