

# Client-Server BBS 之 Protocol

林昱仁、沈嘉峻  
交通大學資訊科學所  
分散式系統實驗室  
helios@cis.nctu.edu.tw  
gis84502@cis.nctu.edu.tw

## 1. 摘要

在最近短短幾年之中，TANet BBS 的發展非常快速，由初期自國外引進時不到百人的使用人口，至目前由於國內對該系統不斷的自行修改與開發新功能，以至如今百花爭鳴蔚為風尚的地步，整體的使用者人口成長到幾萬至幾十萬人之眾。隨著網路環境的變遷，多媒體的使用環境已是必然趨勢，並且由於 BBS 使用人口的快速成長等等因素，這些接踵而來的需求在目前 TANet BBS 的設計使用方式上就衍生出不少問題以及發展上的障礙出現。一個以主從架構方式的 BBS 正足以解決上述這些所遭遇的問題。作者在此文中針對於主從架構 BBS 的起因、相關背景、設計概念、協定內容、與系統分析等部份加以一一的探討與說明，希望能藉此加速國內有關主從架構 BBS 的發展進度，使得標準統一的協定能早日出現。

## 2. 前言

在 TANet BBS 的發展過程之中，可說是超乎大家的預期；在使用人口方面，由當初開始時幾百位使用者到目前的幾萬至幾十萬個使用者；在架設的 BBS 站數量方面，由最初只有一兩個 BBS 站到如今已有上百個 BBS 站的存在；在使用的系統方面，由原本使用國外所設計的系統至目前自行改進與設計，已超越了國外最初的设计，這種種都在國內網路的使用上創造了不少特例。

BBS(Bulletin Board System) 最初的设计目的是一整合了討論區(Board)、信件(Mail)與交談(talk & chat)功能的系統，本是屬於小型區域性的使用特性。然而，隨著國內在使用者需求與功能設計上的改變，目前已漸漸朝大型廣域的使用特性上轉變。原本 BBS 與外界互不相關的特性也產生變化，TANet 上各 BBS 之間以及 BBS 與其它網路系統之間都已產生互動相互的影響。各 BBS 討論區中的信件已可經由 News 系統來達到互相交換的目的，而區域性的 Mail 功能也擴展到可以做 Inetnet E-Mail 的傳輸功能，Chat 功能也擴展到 IRC 的使用方式，這種種的改變使得 TANet BBS 對網路的影響力大增。

伴隨著 BBS 的發展，使用者的增加快速使得各 BBS 站的使用呈現飽和狀態達到系

統極限，並且因應網路目前發展趨勢能夠有視窗化介面的操作方式與具備多媒體的特性，以及將系統工作能有效的加以分散而不致於像如今般全然集中於單一的機器之上，為了解決上述這些問題，因此以主從架構方式的 BBS 變成為大家所想到的一個解決辦法。

## 3. 相關背景

目前主從架構 BBS 的發展在國內各校中皆有不同的設計與研究發展在進行中，然而，如今有正式公開且擁有較多人使用者的系統主要有下列二者：

### 3.1 PowerBBS 系統：

PowerBBS 系統為由國內自行設計，整套系統皆是以主從架構為主體，使用者在使用此系統時需有其專屬的 client 端程式方可使用。由於此系統不同的設計，其操作方式與功能特性與一般多數的 TANet BBS 並不相同，而是較接近仿間單純以電話撥接使用的 BBS。在此系統協定的設計方面，雖然其不是以 telnet 的方式來上線使用，但在所使用的主從協定設計上則是與 telnet 所用協定的基本精神有些相似處，主要是以畫面資料傳輸為主，意即是當 server 端接收到 client 端使用者的操作命令後，將其想要讓使用者見到的畫面資料傳給 client 端來顯示。在此種方式的設計下，server 端將具有較大的變化彈性，也就是指 server 端可更改使用者所見到的畫面形式，但是相對來說在 client 端即擁有較小的彈性，無法自行決定其想要的資料編排方式。在此種方式下主要的工作將仍然由 server 端來負責完成，無法有效分散 server 的負擔，除非在 client 端以 cache 的設計方式來記憶保留些許資料，若使用者有需要重覆使用時可以不需再至 server 端取用。

### 3.2 FormosaBBS 系統：

FormosaBBS 系統是在目前使用的 telnet 方式系統之下，再設計一主從架構的協定給其 client 與 server 程式使用。此種方式是與原先的 telnet 方式

保持並存，兩者對相同的資料來進行存取，因此除了在使用方式上有所差異外，對於連接上同一 BBS 的使用者來說，其所見資料皆是相同的，而兩種方式的使用者間仍然可以互通訊息，如進行線上的交談。這種設計的方式在基本上與作者所提出的協定是相同的。不同之處只是其在設計上是以原 FormosaBBS 為考量，因此在協定設計上有不少是以其專屬的特性為考量因素，在某些設計上並不完全適用於其它各版本 BBS 程式的設計特性。

#### 4. 系統分析

主從架構方式的 BBS 與現行以 telnet 為主的 BBS 在各方面皆有不同之處，以下將就各方面的不同一一加以說明。

在上站方式上，目前的 telnet 方式對很多人來說簡單方便，這是因為在多數網路環境下皆有此程式可以直接使用。而主從架構的方式則須要再另外安裝特別的專屬 client 端程式方可使用，因此若是版本更新時使用者需再安裝新版的程式；更甚者若是各 BBS 由於協定的不同而有不同的 client 端使用程式，則使用者要上不同的 BBS 尚需要不同的程式才行，如此將造成使用者的麻煩。

在操作方式上，telnet 之使用皆為單純的鍵盤操作且為文字模式，而在主從架構下則可以設計為圖形介面的使用者環境並且運用滑鼠的方式來加以操作控制，此外尚可加入多媒體的特性發展成為一多媒體 BBS。

在系統的負擔上，主從架構的設計優點之一是可減輕 BBS 系統本身的負擔，將原先部份工作分配給 client 端的程式來加以處理，這樣可以減少系統的負載，避免將所有工作集中於單一的主機上來執行。

在系統程式的開發上，現今 TANet BBS 的程式開發與修改工作皆是由各站的系統管理者來負責，因此管理者的工作繁重。而以主從架構方式的 BBS 由於有公開的協定，其 client 端的應用程式可由使用者自行加以開發，server 端僅具有擔任提供使用者所需基本資料的角色，至於資料的再次加工處理與展現方式則由 client 端的設計者自行決定。在此種方式之下，BBS 系統管理者由原先需兼具有系統管理與程式開發兩者的工作轉變為只需專注於對系統的管理一項工作即可，如此可以減輕許多負擔。

在系統的使用人數上，以主從架構的設計方式不會受現行安裝系統 tty 數量的限制，可以增加系統同時可容納的線上使用人數。這是因為 UNIX 系統本身的設計限制，在目前的 SunOS 4.1.x 上經過 patch 後也只能

容納到 512 人。

在應用發展上，以主從架構的方式所設計的 BBS 來說，對於 client 端若要其將所連接的各不同 BBS server 之間的資源加以整合以做特殊的統一利用將更為方便。

#### 5. BBSTP(Bulletin Board System Transfer Protocol) 傳輸規格標準

##### 5.1 指令

指令是由一指令字組所組成，而某些指令字組後可能跟隨著數個指令參數。這些指令字組是不分大小寫的。指令字組與各參數間是以一個或多個的空白或是 tab 字元來加以分隔。每一個指令列是以一對 CR-LF(Carriage Return - Line Feed)來做為結束字元符號，並且總長度不能超過 512 個字元。

##### 5.2 回應

有兩種類別的回應，分別是文字回應和狀態回應。

###### 5.2.1 文字回應

文字類別的回應是回應一連串的文字列，每列文字以一對 CR-LF 為結束。而此文字回應以單行包含單一的句點(.)，其後跟隨著一對 CR-LF 做為整個文字回應的結束。若該行文字內容第一字為句點，則在傳送時需將此一句點重覆，也就是傳送兩個句點。在接收端若發現該行不是代表結束的單一句點資料，且其第一字為句點的話，則去除掉此一句點(也就是發送端所重覆的資料)。

###### 5.2.2 狀態回應

在回應端的狀態回應列的啓始是 3 位數的代碼，此部份代碼的意義就採用與 NNTP 相同的分類意義如下：

- 1xx - 訊息資料。
- 2xx - 指令正確。
- 3xx - 指令尚未完成，需繼續傳送其它資料。
- 4xx - 指令正確但因某些原因而無效。
- 5xx - 指令未提供、錯誤或是嚴重的程式錯誤發生。

這樣分類的好處之一是將所有狀態分類好，Client 端在判斷時，除非有特別需要，否則 2xx、3xx 表示成功，4xx、5xx 就是失敗。如此比較清楚，而不會在一串連續的數字中，成功失敗交叉混著，這樣判斷的狀態會較複雜化。

在此代碼後跟隨著一些參數，依回應的狀態而有所不同，各參數間是以一個以上的空白或 tab 字元來加以區隔，最後同樣以一對 CR-LF 為結束。

##### 5.3 一般性的狀態回應

在一般性的狀態回應約有如下這幾個：  
100 指令說明  
200 bbs 電子佈告欄系統 Server BBSTP Ready  
400 Service unavailable  
401 需先執行其它指令(如 Login 動作)  
410 指令參數錯誤  
500 指令錯誤  
501 指令語法錯誤  
502 access restriction or permission denied

## 6. 詳細的指令與回覆

在指令描述中，以中括號[] 括起者表示其中的資料是選擇性資料，可選擇任一參數指令。

### 6.1 連接回應

200 bbs Bulletin Board System Server BBSTP Ready  
400 Connection refused for some reason

### 6.2 指令 FILE

#### 6.2.1 FILE

FILE [issue | welcome | signature | plan | override]

此指令是用來要求 Server 傳送一些系統或使用者的基本檔案資料。其中，issue 為連上 BBS 但尚未 Login 前，Server 想告知 Client 之資料；welcome 為 Login 成功後，Server 對於 Client 的公告資料；signature 為使用者的簽名檔；plan 為使用者的計畫檔；override 為使用者的朋友名單資料。

#### 6.2.2 回應

202 File transfer - xxxx

在此一狀態回應之後則跟隨著實際內容的文字回應資料。

### 6.3 指令 USER

#### 6.3.1 USER

USER userid

此指令是用來登錄系統之用，需輸入登錄使用者之代碼。視不同的 BBS 系統設計，此使用者代碼的大小寫可能相關也可能無關。若使用者是要求註冊一新的帳號，則是以回應 310 的狀態進行交談式的新帳號資料輸入。

#### 6.3.2 回應

210 user ok

211 User Login Success

/\* 使用者成功登錄(不需密碼) \*/

310 New User Register, request data

/\* 輸入新使用者資料 \*/

319 New User Register, 保密資料

/\* 表 Client 端輸入時不顯示 \*/

411 此資料不被允許

#### 6.3.3 新帳號註冊範例

C): USER new

S): 310 使用者代碼

C): sysop

S): 411 此使用者代碼已有人使用

C): helios

S): 319 使用者密碼

C): xxxxxxxx

S): 310 您想要取的綽號

C): 我的綽號

S): 310 真實姓名

C): 林昱仁

.....

S): 310 螢幕模式

(ansi,vt100,vt102,h19a,dumb,...)

C): vt100

S): 211 Login Success

### 6.4 指令 PASS

#### 6.4.1 PASS

PASS passwd

此指令是用來輸入使用者密碼。

#### 6.4.2 回應

211 User Login Success

/\* 使用者成功登錄 \*/

401 USER assign first

/\* 尚未指定使用者代碼 \*/

### 6.5 指令 LOGFAIL

#### 6.5.1 LOGFAIL

LOGFAIL [get | clear]

此指令是用來要求/清除使用者 Login 失敗記錄資料，其中 get 為要求資料；clear 為清除資料。

#### 6.5.2 回應

224 data follows

/\* 之後為文字回應之資料 \*/

210 loginfail clear ok

### 6.6 指令 MULTILog

#### 6.6.1 MULTILog

MULTILog [get | kill] [辨識代碼]

此指令是用來要求(get)使用者的重覆 Login 資料，亦或是清除(kill)使用者之重覆 Login。

#### 6.6.2 回應

224 data follows

/\* 之後為文字回應之資料 \*/

210 Multi-Login kill ok

#### 6.6.3 文字回應資料

在 MULTILog get 中的文字回應資料每列格式如下：

[辨試資料]tab[上站地點]tab[目前狀態]

### 6.7 指令 TIME

### 6.7.1 TIME

TIME [system | welcome | bnote]

此指令是用來要求 Server 有關時間的資訊。其中，system 是目前系統的時間；welcome 是系統 welcome 資料的最近更新時間；bnote 則是目前所選擇討論區之 welcome 資料的最近更新時間。所回應的 time-stamp 值為從 00:00:00 GMT, Jan, 1, 1970 所算起的秒數，若無此資料則回應的 time-stamp 為 0。

### 6.7.2 回應

203 [time-stamp] time

## 6.8 指令 MAIL

### 6.8.1 MAIL

MAIL [userid]

此指令是用來詢問使用者信箱的基本資料，包含了未讀的信件數(#unread)與全部的信件數(#total)。若不加 [userid] 參數表示詢問的是登錄使用者本身資料。

### 6.8.2 回應

220 [#unread] [#total] mail

## 6.9 指令 BOARD

### 6.9.1 BOARD

BOARD [boardname] [zap|unzap]

此指令是用來選定所要的討論區，參數為討論區名稱，並可選擇此討論區是否要隱藏(zap)起來。回應的狀態資料有此討論區的文章總數(#total)與該使用者最近使用該討論區的時間(last-visit-time-stamp)。

### 6.9.2 回應

211 [#total] [last-visit-time-stamp]  
boardname

## 6.10 指令 LIST

### 6.10.1 LIST

LIST [boards]

此指令是用來要求列出目前所有討論區的相關資料。

### 6.10.2 回應

224 data follows

### 6.10.3 文字回應資料

在 LIST boards 中的文字回應資料每列格式如下，其中，[等級]部份資料，‘y’表此討論區可以張貼，‘n’表不可以張貼。[投票狀態]部份資料，‘V’表投票中，‘R’表投票結束。[Zap狀態]部份資料，‘Z’表此討論區使用者是設定隱藏起來，‘Y’表是非隱藏狀態。而各欄位的資料長度基本上並無限制。

[討論區名稱]tab[等級]tab[描述]tab[板

主]tab[投票狀態]tab[Zap狀態]

## 6.11 指令 OVER

### 6.11.1 OVER

OVER [board | mail] LowNumber

[HighNumber]

此指令是用來要求對目前討論區(board)或私人信箱(mail)中之單封文章或是一個範圍的文章(加上 HighNumber 參數)之簡短描述資料。

### 6.11.2 回應

224 data follows

### 6.11.3 文字回應資料

在 OVER [board | mail] LowNumber [HighNumber] 中的文字回應資料每列格式如下，其中，[序號]表文章編號從‘1’開始。[閱讀狀態]為‘,’,’N’,’m’,’M’,...等狀態。[作者]部份站內使用者為使用者代碼，站外使用者為其 E-mail Address。[時間]為 time-stamp 格式。

[序號]tab[閱讀狀態]tab[標題]tab[作者]tab[時間]tab[檔案名稱]

## 6.12 指令 POST

### 6.12.1 POST

POST [board | mail] [local]

此指令是用來在目前所選擇討論區中張貼文章(board)，或是送出一封私人信件(mail)。

### 6.12.2 回應

340 send article to be posted. End with  
<CR-LF>.<CR-LF>

440 post not allowed

### 6.12.3 文章(信件)格式

在 POST [board | mail] 中當狀態回應為 340 時，Client 需繼續送文章資料，此文章資料格式可參考 RFC850 中之規定，內容為文章的標頭資料加上文章的本文所組成，若最後有加上 local 參數表此文章設定為 local save 狀態。文章的標頭資料有 Subject: 與 To: 兩者，舉例如下：

340 send article to be posted. End with

<CR-LF>.<CR-LF>

Subject: this is a test subject

To: helios, sysop, gis83504@cis.nctu.edu.tw

/\* 或是多個 boardname \*/

This is article body

.

## 6.13 指令 INFO

### 6.13.1 INFO

INFO [system | menu | user] [userid]

此指令是用來提供系統及使用者資訊之用。在 INFO user 指令中若加入一 userid 參數，則表示要查詢的是其它使用者的資料(相對於目前的 Query 功能)，否則是顯示目前使用者的基本資料。

#### 6.13.2 回應

202 File Transfer -- xxx

#### 6.13.3 文字回應資料

在 INFO system 中的回應資料為提供系統資訊之用，其格式為 keyword = value 的型態，一般的 keyword 為 BBSID, BBSNAME, BBSDOMAIN ... 等資料。而在 INFO menu 中的回應資料為此 BBS 站的 menu 編排資訊(此資訊的格式採用 PhoenixBBS 4.0 的 menu 格式，此資料只有文字模式的 Client 有用處)。至於 INFO user 的指令，若其後有加上 userid 參數查詢其它使用者的資料，則傳回的文字資料為一畫面無特定格式意義存在；若是不加 userid 參數查詢使用者本身資料，此資料以如同 INFO system 相似的格式傳回，為 keyword = value 的方式，而可能的 keyword 為 ID, NickName, Email, Login#, Post#, ... 等。

### 6.14 指令 ANNOUNCE、BDIGEST

#### 6.14.1 ANNOUNCE

ANNOUNCE [path]

此指令是用來獲得該 BBS 站公佈欄之資料。此指令之 Protocol 與 Gopher 相似，當 [path] 參數不給時所獲得的是最起始頁的資料。而 BDIGEST 指令與 ANNOUNCE 相似，所不同的是 BDIGEST 所獲得的是目前所選擇討論區的精華資料。

#### 6.14.2 回應

224 data follows

#### 6.14.3 文字回應資料

回應的文字資料格式如下：[type] 表此項目的種類，‘0’表示為檔案，‘1’表示為目錄。[path] 為要獲得此項資料的 path 參數。[site] 若有的話表示此項為一 Gopher Link，可依 path, site, port 三者之資料依 Gopher protocol 與指定 Gopher 連接獲得資料。[更新時間] 為此項目的最近更動時間。

[type]tab[描述]tab[path]tab[site]tab[port]  
tab[更新時間]

### 6.15 指令 EGROUP

#### 6.15.1 EGROUP

EGROUP [path]

此指令是用來獲得該 BBS 站討論區分類群組之資料。當 [path] 參數不給時所獲得的是最起始頁的資料。

#### 6.15.2 回應

224 data follows

#### 6.15.3 文字回應資料

回應的文字資料格式如下：[type] 表此項目的種類，‘0’表示為討論區，‘1’表示為分類群組。[path] 為要獲得此項資料的 path 參數，若此項為討論區則 path 為此討論區之名稱。  
[type]tab[描述]tab[path]

### 6.16 CHATROOM Protocol

在 CHATROOM Protocol 設計上是以新的 EagleBBS 3.0 的 chatroom 為參考依據，使用者可以由 INFO system 指令中的 CHATPORT keyword 得知 chatroom port 所在，自行與 CHAT Daemon 溝通。

當連上 chatroom daemon 後，所輸入的第一列指令如下：

```
!/ <unum> <userid> <chatid> <passwd>
```

輸入此行後，chatroom daemon 的回應為兩個字元，如下所定義：

```
#define CHAT_LOGIN_OK      "OK"  
#define CHAT_LOGIN_EXISTS "EX"  
#define CHAT_LOGIN_INVALID "IN"  
#define CHAT_LOGIN_BOGUS  "BG"
```

其中，OK 表示順利進入。EX 或 IN 表示 chatid 不允許，可重換 chatid 再次輸入 !/ 的指令。而 BG 表示資料錯誤(包含 unum, userid, passwd)，此情況 chatroom daemon 將會把你的連接 socket 關掉。

進入之後，由 chatroom daemon 所傳回來的資料，每行的第一個字元若不是 '/' 則都顯示於螢幕，若是以 '/' 開頭則表示特殊意義如下：

/c : 要求 client 端清除螢幕

/n : 傳回你新的 chatid

/r : 你進入的 room 名稱

/t : 你所在 room 的話題

而這些指定的資料都從第三個字元開始，就是緊跟在 /x 之後。

### 6.17 指令 TALK

#### 6.17.1 TALK

TALK [userid] [talkport] [userno]

#### 6.17.2 回應

210 Command OK

450 TALK not allow for some reason

#### 6.17.3 運作方式

基本上，目前在 EagleBBS 系列的 BBS 中，其 Talk 的方式本就是透過 socket 來傳輸資料，當 A 使用者要與 B 使用者交談時，則會先建立一個

socket 來等待對方連接，然後送出一個 Signal 信號通知 B 有人要與其交談，而 B 使用者收到此訊號後，由系統中查出是 A 要與其交談，以及 A 所建立的 socket port 為何，此時 B 就與 A 所建立的 socket 建立連線，然後透過此連線告知 A 是否要與之交談。因此當擴充此方式的 Client-Server 架構時，所不同的是 A 通知 B 的方式，以及 B 向系統查詢是誰要與其交談資料這兩者在 Client 端會有所不同。

由於每個 Client 在 BBS 主機上都有一對映的 process 存在，因此在設計上此一 process 與其它使用者在主機上 process 的運作關係就與原架構相同，所要擴充的是在 Client 端的 process 與其對映主機上 process 彼此間的溝通方式。

當 Client 端想與其它使用者交談時，則以 TALK [userid] [talkport] [userno] 指令通知 Server，而 Server 收到此指令後則像原本一樣送一個 Signal 給所要交談對象的 process 知道。

當有人想要與 Client 使用者交談時，Client 在主機上對映的 process 會先收到通知，而此 process 就可向系統查詢對方的資料，然後將此資料以下列格式傳給 Client 端知道。則 Client 端可以依此資料自行與對方連接告知是否要與對方交談。

```
page [userid]tab[username]tab[fromhost]tab[talkport]
```

## 6.17 指令 WRITE

### 6.17.1 WRITE

```
WRITE [userid]tab[Msg]tab[userno]
```

### 6.17.2 回應

```
210 Command OK
450 WRITE not allow for some reason
```

### 6.17.3 運作方式

WRITE 的運作方式與 TALK 是相同的架構，當 Client 端要發出 Write 訊息時則以如上指令的方式送出。而當 Client 端要接收 Write 資料時，在主機上所對映的 process 則以如下的格式送出資料告知 Client。

```
write [userid]tab[username]tab[Msg]
```

## 6.18 指令 VOTE

### 6.18.1 VOTE

```
VOTE [control | data | result] [boardname]
[vote-str]
```

此指令是用於投票的相關功能上，可分為三類：一是獲得投票資訊

(control)；二是投票(data)；三是得知投票結果(result)。若是系統投票以特殊的 boardname (#system) 來取得。

### 6.18.2 回應

```
210 VOTE OK
202 VOTE result data follows
340 Send vote data. End with
<CR-LF>.<CR-LF>
```

### 6.18.3 文字回應資料

```
VOTE control [boardname] [vote-str]
```

1.其中，當 vote-str 不給的時後是得到目前此板所舉行 vote 簡短說明，

傳回格式如下：(若無投票傳回的內容為空的)

```
[vote-str]tab[vote-short-desc]
```

```
836461537 投票簡短描述
```

2. 若加上上述所列 vote-str 後，則回傳這項投票的相關控制訊息，如下：

```
End-Time: Wed Jul 10 20:21:47 CST
```

```
1996
```

```
Ballots: 2
```

```
Items: A 我天天來看耶....always..
```

```
Items: B 我幾天才來看一次啦.....often...
```

```
Items: C 我有時....sometimes..
```

這是一次測試投票，這裡是投票的完整描述說明

.

預計是以像 Post 的方式傳回相關控制訊息，一個 Header: Value 的方式，End-Time: 這個 Header 顯示結束時間。Ballots: 表示可投票數。Items: 是此次投票可選的項目，其中後面 Value 的第一個字母是投票控制用的字元，當然不定要是 A,B,C,.. 要用 1,2,3 或 @,\$,^ 也都是可以的。在所有標頭結束後空一行空白，接下來是整個投票的完整描述說明，最後以一行 '! 當結束。

```
vote data [boardname]
```

在下完此指令後 Server 回應狀態 340 表示繼續接收資料，此時 Client 可送的資料格式如下：

```
Ballots: AC
```

```
Advice: 這次投票好爛...
```

.

同樣是用類似一封 post 的方式，一個 Header: Value。Ballots: 所投的票(如上是投 A,C 共兩票)。Advice: 對此次投票給的意見。

```
vote result [boardname]
```

傳回目前所有的投票結果。

## 6.19 指令 PUT

### 6.19.1 PUT

PUT [signature | plan | override]

此指令是用來更新使用者個人資料檔案，包含了簽名檔、計畫檔與朋友名單。

### 6.19.2 回應

340 Send data to be transfered. End with

<CR-LF>.<CR-LF>

240 Data transfer OK

### 6.19.3 文字傳送資料

在 PUT [signature | plan] 下所傳送的資料無特定格式，而在 PUT override 下所傳送的資料格式，每列的第一個欄位資料為使用者代碼。

## 6.20 指令 BODY

### 6.20.1 BODY

BODY [board | mail] Number [filename]

此命令是獲得文章或信件的內容資料。

### 6.20.2 回應

202 num<filename> body

## 6.21 指令 DELETE

### 6.21.1 DELETE

DELETE [board | mail] filename

LowNumber [HighNumber]

此指令是用來刪除一封或一個範圍的文章資料。

### 6.21.2 回應

210 article deleted

## 6.22 指令 SEARCH

### 6.22.1 SEARCH

SEARCH [board | mail] [author | title] [a | p]

[+ | - | \*] Number Pattern

此指令是用來搜尋文章之用。

[author | title] 表示是要對作者或標題搜尋。[a | p] 表示搜尋是做全部符合或部分符合。Number 表示由第幾封文章開始搜尋。[+ | - | \*] 表示搜尋的方向是向前、向後、或是全部搜尋。Pattern 為所要搜尋的樣本資料。

### 6.22.2 回應

224 data follows

### 6.22.3 文字回應資料

所回覆的文字回應資料格式與 OVER 指令是相同的。

## 6.23 指令 SET

### 6.23.1 SET

SET [pager | color] [on | off]

SET userdata

此指令是用來設定使用者的基本個人資料與雜項屬性資料。以 SET userdata 參數可以用相似於新使用者註冊相同的對談式 protocol 來更改使用者的個人基本資料(此時若輸入空白表示不更改此項資料)。而其它 SET [pager | color] [on | off] 等則可用來切換設定使用者的各項屬性資料。

### 6.23.2 回應

210 User character change ok

310 Modify User Data, Send Data (Empty for not change)

## 6.24 指令 USERS

### 6.24.1 USERS

USERS

此指令是用來獲得目前站上所有線上使用者的列表。

### 6.24.2 回應

256 online user list follows

### 6.24.3 文字回應資料

此指令的文字回應資料格式如下：  
[userno] 為此使用者的識別碼。[pagemode] 為此使用者的呼叫狀態。[mode] 為使用者目前狀態。[idletime] 為此使用者的 Idle 時間。

[userno]tab[userid]tab[username]tab[from]tab[pagemode]tab[mode]tab[idletime]

## 6.25 指令 HELP

### 6.25.1 HELP

HELP

此指令為查詢系統所提供的指令。

### 6.25.2 回應

100 Legal Commands

## 6.26 指令 QUIT

### 6.26.1 QUIT

QUIT

此指令是用來離開此系統，結束使用。

### 6.26.2 回應

205 connection close -- good bye

## 7. 結論

主從架構 BBS 的發展在國內來說已有超過兩年以上的歷史，但是在其使用的人數及推廣上來說至今尚不普遍，這其中有種種的原因存在，最主要的因素之一就是因為目前並沒有一個統一的主從架構 BBS 的協定出現，也由此無法有統一的力量來推動。作者藉由對目前在 Internet 上數種服務的通訊協定加以參考也同時考慮國內各版本 BBS

程式設計的特性與各相關管理者的意見後，而提出此套主從架構 BBS 通訊協定，希望能因此獲得廣大快速的回響，使得一個標準的主從架構 BBS 協定能夠早日得到共識，進而加速主從架構 BBS 的使用在國內的推廣進度，讓眾多的使用者能有一個最佳的 BBS 使用環境。

## 8.參考資料

1. B. Kantor, P. Lapsley, "Network News Transfer Protocol: A Proposed Standard for the Stream-Based Transmission of News", RFC 977.
2. F. Anklesaria, M. McCahill, P. Lindner, D. Johnson, D. John, D. Torrey, B. Alberti, "The Internet Gopher Protocol (a distributed document search and retrieval protocol)", 03/18/1993, RFC 1436.
3. J. Postel, "Simple Mail Transfer Protocol", 08/01/1982, RFC 821.