Network Programming, Fall, 2003

Project 3: Batched Remote Access System (ras) in HTTP

9123571 Chung-Wei Hang 杭仲瑋
at Learning Lab, CIS, NCTU 交大資科學習科技實驗室
gis91571@cis.nctu.edu.tw

Nov, 24, 2003

Requirement 程式需求

In this homework, you are asked to write a batched ras HTTP system.

1

Write a CGI program to receive an HTTP request described as follows. The parameters of the HTTP request are:

h1=140.113.210.101	# the first ras server's IP.
p1=7070	# the first ras server's port.
f1=batch_file1	# the batch file name redirected to the first ras server.
h2=140.113.210.103	# the second ras server's IP.
p2=7070	# the second ras server's port.
f2=batch_file2	# the batch file name redirected to the second ras server.
h3=140.113.210.103	# the third ras server's IP.
p3=7070	# the third ras server's port.
f3=batch_file3	# the batch file name redirected to the third ras server.
h4=	# no more ras server.
h5=	# no more ras server.

Then, the CGI program connects to the three ras servers and then redirects the batch file (stored in the HTTP server) as input to these servers. When receiving messages, send these messages back to the browser as the returning web page. In each line, you must mark which server and line number. For example,

```
140.113.210.101:1> %                          # ls
140.113.210.102:1> bin/ test.html
140.113.210.102:2> %                          # ls bin
140.113.210.103:1> ls cat removetag number
140.113.210.101:2> %                          # printenv PATH
140.113.210.102:3> PATH=/bin
140.113.210.103:2> %                          # cat test.html
...
```

In order to see the effect of select, you can either use ¡br¿ to break each line or simply use the header ”Content-type: text/plain”.

Requirement

- In your web page, you must be able to accept at least 5 batches.
- You can use your own ras server (in HW#1) for testing.
- You must use nonblocking I/O (including connect()) for connections to ras servers.
- It is suggested that in your ras server you call ”sleep(10)” before accept(), in order to test non-blocking.
- It is suggested that you use a large batch file to test output blocking.
- You do not need to consider nonblocking connections back to servers.
- The CGI program must be working with Apache server in Unix.

2

Write a simple http server in Unix to support CGI.

Requirement

- This must work with TA’s CGI program and the previous CGI you wrote.

3

In Windows, write a console program for 1 working with httpsvr in Windows.

Requirement

- You can get httpsvr from MSDN. TA will also post.

4

In Windows SDK, write an asynchronous version of 1.

Requirement

- You should use WSAAsyncSelect or CAsyncSocket.
- TA will give you a sample code of using WSAAsyncSelect (next week) that you can use.

Due Date

- 11/24 for 1 2.
- 12/08 for 3 4.

Program Overview 程式概觀

CGI with BSD Socket

4.1 Source File: cgi.c 原始碼

這部份主要利用 concurrent client 的架構, 利用教科書上面的 connectTCP.c 和 connectsock.c 以及講義上的範例, 在 nonblocking 的模式下藉著呼叫 select 來同時處理連接多個 rasd service. 其中主要有下列幾個重要的 function:

- **parseServerInfo** 負責 parse 不論是 GET 從 QUERY_STRING 環境變數中或者 POST 從 stdin 傳進來的參數, 包括 rasd service 的 host name, port, 還有 batch file.

- **connectRasd** 在這 function 裡面不斷的呼叫 select 之後, 判斷是那些連線需要 read 或 write, 分別交給 reader 和 writer 來處理.
- **reader** 負責從需要 reader 的連線把所有的資料讀進來, 接著進行處理, 隨即 output 到 client (browser) 那端. 此外, 如果已經沒有資料可以讀[1], reader 也負責作關閉連線的動作.
- **writer** 負責在連線可以被寫入的時候, 試著從指定的 batch file 讀入一行行的命令, 接著先紀錄起來[2], 再嘗試寫給 rasd service[3].

HTTP Server with BSD Socket

4.2 Source File: webd.c 原始碼

利用類似 project1 的 server paradigm, 包括 passiveTCP.c 以及 passivesock.c, 接受 client (browser) 的連線和 GET, POST, . . . 等 request, 處理包括 html, jpeg, gif, 文字檔, 和執行 cgi 程式, 其中主要的 function 如下:

- **web** 這部份要程式最主要的 function, 包括接受 client request, 處理路徑, 設定環境變數, . . . 等[4].
- **env** 這 function 幫助建立環境變數的格式, 也就是 ”key=val” 的樣子.

Implementation Issues 實作要點

CGI with BSD Socket

reader Flow 流程

1. 對 socket 作 read 的動作, 判斷讀回來的字元數 n:
    - $n < 0$ 發生 error.
    - $n = 0$ 表示 end of file, 關閉連線.
    - $n > 0$ 處理 buffer 中換行字元和 ”%” (promt) 兩種字元, 並視情況加入先前 write 出去的 commands.

[1] read 回傳值是 0

[2] 以便 reader 在收到 promt 符號時一起 output 給 client

[3] 如果沒有寫完, 會留到下次可以寫入的時候再寫

[4] 請參照 implementation issue 的流程

2. 處理完把一行行的結果, 加入正確的 tag 後印出到 client (browser).

3. 如果先前讀的資料沒把 buffer 讀滿, 表示可能還有資料沒讀完, 則繼續再作一次.

writer Flow 流程

1. 檢查屬於每個 rasd 連線的 write buffer, 如果已經沒東西了, 就去 batch file 裡面抓下一行:
   - 如果 batch file 碰到 end of file, 則關閉這個連線的 write[5].
   - 如果抓了一行新的命令, 先紀錄起來以供 read 處理 promt 時候使用.

2. 把他寫出去, 如果沒寫完, 留到下次可以 write 的時候再寫.

Other Issues 其他注意事項

- 當 select 表示一個 fd 可以被 read, 那麼這些需要被 read 的資料一定要被 read 完畢, 否則會使得 select 不會再對那個 fd 作偵測.

- 在 batch file 中每一行命令不可以超過 BUF_SIZE - 1.

- 經由 GET 或 POST 送過來的參數數量不能超出 BUF_SIZE 個.

HTTP Server with BSD Socket

web Flow 流程

1. 把 client (browser) 的 request 讀進來.

2. 對收到的 request 進行 parse 的動作.

3. 針對 GET 或 POST 設定好參數:
   - **GET** 直接把參數串設成 ”QUERY_STRING” 環境變數.
   - **POST** 把參數串寫到 pipe 裡面.

4. 設定所有的環境變數.

5. 設定路徑及判斷 content type:

[5] shutdown(fd, 1);

    - **cgi** 將剛剛寫入參數串的 pipe 導到 stdin, 將和 client (browser) 的連線導到 stdout, 執行 cgi.
    - **html, jpeg, gif, plain text** 將相對應的檔案寫給 client (browser).

Other Issues 其他注意事項

- HTTP request 的長度不可以超過 BUF_SIZE.
- 環境變數的數量不可以超過 50.

Reference 參考資料

- Man pages of select, connect, select_tut, read, write.