Computer Networks
Term Project (Part I)

公佈日期: 2021/11/29

繳交日期: 2021/12/20 14:10 前 (待確定) (當天有事無法至課堂繳交者，請提早繳交或是請同學代為繳交)

繳交方式: 將報告列印後交給助教或於當日上課時繳交，**程式**與**報告**檔案必須於繳交時間內上傳至 E-Learning 系統，請先將所有檔案壓縮成一個 zip 檔，檔名格式為"學號_姓名.zip"，學號英文字母請大寫。

題目:
使用 C++撰寫一個簡易的 HTTP/1.0 伺服器，功能需求如下：

基本功能：

1. 支援 GET Request，並具備以下功能：
   a. 根據收到的 request 內容，將正確的內容以 HTTP Response 訊息格式回覆給 client 端。請注意，回傳的 HTTP Response 必須包含正確的 Content-Type，方便 client 端收到後進行後續的處理。
   b. 若 request 要求的物件不存在，需回覆符合格式的錯誤訊息 (即 404)。
   c. 若 request 的語法有誤，需回覆符合格式的錯誤訊息 (即 400)。
2. 可於啟動時指定伺服器接受的通訊埠號碼，預設值為 80。(例如: 啟動時執行 mywebd -p 9527 代表伺服器接受的通訊埠號碼為 9527)
3. 同一時間只需要接受一條 TCP 連線，不需要利用多程序或多執行緒處理同時間的多條 TCP 連線。

進階功能：

1. 將所有存取紀錄輸出至 log 檔案，log 格式請參考附件 1。

評分方式:

- 程式 (80 points)
  1. 具有題目所要求的基本功能 (60 points)
  2. 具有題目所要求的進階功能 (15 points)
  3. 程式可讀性, 如註解, 縮排, 變數命名等 (5 points)
- 作業報告 (20 points)
  - ■ 內容是否充實 (15 points)
  - ■ 整齊美觀 (5 points)

  作業報告至少需包含下列內容:
  1. 封面
  2. 操作說明: 說明程式的操作方式
  3. 完整程式與說明: 附上完整程式並說明程式的架構與功能, 若有使用特別的技巧

也請說明。為節省紙張，建議程式在不影響閱讀狀況下，使用較小的字體列印。

4. 參考資料: 若有參考教科書以外的書籍或網路上的資料, 請在此部份詳列

注意事項:

1. 程式檔案中請包含已編譯好的執行檔案。
2. 必須自行撰寫程式解析 HTTP request 訊息與產生 HTTP response 訊息，不可使用程式語言提供的或是其他現成的 HTTP 相關函式庫。
3. 程式僅可實作題目要求的功能，若包含題目未要求功能的程式片段，將視程度予以扣分，嚴重者可能被扣至零分。
4. 若有參考或是使用他人所撰寫的部份程式片段，請務必於報告中的參考資料部份詳列，否則視為抄襲，此次作業分數為零分。
5. 參考他人程式請斟酌程度，勿直接複製貼上，否則仍會被扣分。情況嚴重者，將視為抄襲。

附件 1: Access Log Format (以下內容取自 Apache HTTP Server Project，並有經過稍微修改。https://httpd.apache.org/docs/2.4/logs.html)

A typical configuration for the access log might look as follows.

```
"%h %l %u %t \"%r\" %s %b"
```

The format string consists of percent directives, each of which tell the server to log a particular piece of information. Literal characters may also be placed in the format string and will be copied directly into the log output. The quote character (") must be escaped by placing a backslash before it to prevent it from being interpreted as the end of the format string. The format string may also contain the special control characters "\n" for new-line and "\t" for tab.

The above configuration will write log entries in a format known as the Common Log Format (CLF). This standard format can be produced by many different web servers and read by many log analysis programs. The log file entries produced in CLF will look something like this:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

Each part of this log entry is described below.

- **127.0.0.1 (%h)**: This is the IP address of the client (remote host) which made the request to the server. If HostnameLookups is set to On, then the server will try to determine the hostname and log it in place of the IP address. However, this configuration is not recommended since it can significantly slow the server. Instead, it is best to use a log post-processor such as logresolve to determine the hostnames. The IP address reported here is not necessarily the address of the machine at which the user is sitting. If a proxy server exists between the user and the server, this address will be the address of the proxy, rather than the originating machine.
- **- (%l)**: The "hyphen" in the output indicates that the requested piece of information is not available. In this case, the information that is not available is the RFC 1413 identity of the client determined by identd on the clients machine. This information is highly unreliable and should almost never be used except on tightly controlled internal networks. Apache httpd will not even attempt to determine this information unless IdentityCheck is set to On.
- **frank (%u)**: This is the userid of the person requesting the document as determined by HTTP authentication. The same value is typically provided to CGI scripts in the REMOTE_USER environment variable. If the status code for the request (see below) is 401, then this value should not be trusted because the user is not yet authenticated. If the document is not password protected, this part will be "-" just like the previous one.
- **[10/Oct/2000:13:55:36 -0700] (%t)**: The time that the request was received. The format is:
  [day/month/year:hour:minute:second zone]
  day = 2*digit
  month = 3*letter
  year = 4*digit
  hour = 2*digit

minute = 2*digit

second = 2*digit

zone = (`+' | `-') 4*digit

It is possible to have the time displayed in another format by specifying %{format}t in the log format string, where format is either as in strftime(3) from the C standard library, or one of the supported special tokens. For details see the mod_log_config format strings.

- **"GET /apache_pb.gif HTTP/1.0" (\"%r\")**: The request line from the client is given in double quotes. The request line contains a great deal of useful information. First, the method used by the client is GET. Second, the client requested the resource /apache_pb.gif, and third, the client used the protocol HTTP/1.0. It is also possible to log one or more parts of the request line independently. For example, the format string "%m %U%q %H" will log the method, path, query-string, and protocol, resulting in exactly the same output as "%r".
- **200 (%s)**: This is the status code that the server sends back to the client. This information is very valuable, because it reveals whether the request resulted in a successful response (codes beginning in 2), a redirection (codes beginning in 3), an error caused by the client (codes beginning in 4), or an error in the server (codes beginning in 5). The full list of possible status codes can be found in the HTTP specification (RFC2616 section 10).
- **2326 (%b)**: The last part indicates the size of the object returned to the client, not including the response headers. If no content was returned to the client, this value will be "-".