

## 位置づけ

- ネットワーク、特にソケットプログラムの演習。
- 講義ではない、各自が時間を見つけて取り組むこと。
- 利用頻度が特に高い HTTP を中心に進める。

## 準備

- 参考文献 [1] の 1-17 章を読む。
- 参考文献のサンプルコードをダウンロードする。  
スクラッチからコードを作成しても良いが、初心者はサンプルコードを流用して、必要最小限だけ注目してプログラムを作ることを勧める。
- HTTP の関連 RFC に目を通しておくこと。  
必要に応じて細かいところを読めばいいので、最初はリクエストやレスポンス等の用語を把握し、BNF による定義の存在場所を把握すること。
- クライアントやサーバのシステムコールの流れを記憶すること。

# 1 HTTP クライアント作成 (テキスト)

- 150.65.117.115 の HTTP サーバからコンテンツを取得するプログラムを作成せよ。URL は `http://150.65.117.115/` である。
- firefox 等のブラウザで上記 URL にアクセスして、コンテンツを確認する。
- `wget`, `curl` 等でも同様にコンテンツを取得して内容を確認する。
- `telnet` で上記サーバに接続し、リクエストを発行とそのレスポンスを確認する。
- サンプルコードの `TCPecho` や `TCPdelay` を変更すると目的のプログラムになる
  - `connect` の後に `write` で `GET /CRLF` の 7 文字を書き込む。リクエスト送出 (HTTP/0.9 相当) を意味する。
  - `write` 後、`read` で読み込み可能な間、繰り返す `read` 戻り値が 0 か負になるまで繰り返す。レスポンスの受理を意味する。
- 他のプログラムの結果と同じかどうか比較する。
- リクエストを `GET /HTTP/1.0CRLF` の 16 文字に変更する (HTTP/1.0 相当)。
- レスポンスのヘッダとボディの境界を見つけるように変更する。
- レスポンスのヘッダとボディをそれぞれ別ファイルに保存するよう変更する。

## 2 HTTP クライアント作成 (バイナリ)

- 前課題と同様に `http://150.65.117.115/z` を取得するプログラムを作成せよ。  
画像ファイルのなので単純にコンソールには出力できない点に注意
- ブラウザなどでコンテンツ内容を確認する。
- 作成したプログラムの結果を他のプログラムの結果と比較せよ。

## 3 HTTP サーバ作成

- HTTP リクエストの対して何らかのバイト列を返すプログラムを作成せよ。
- リクエストを解析し、対応するファイルを送出するよう変更せよ。たとえば、`/a` を受けたらカレントディレクトリの `a` を返す、等
- 並列に動作することを確認せよ。
  - 巨大ファイルを送出中に、次のリクエストを受けるとどうなるか観測すると良い。
- ソケットオプション `REUSE_ADDR` をセットするよう変更せよ。
- `SIGCHLD` を扱ってゾンビができないように変更せよ。
- リクエスト毎に `fork` している場合は、`prefork` を `prefork` している場合はリクエスト毎に `fork` するよう書き換えよ。
- 多量のリクエストを受けた際の性能を確認せよ。
  - 性能となる指標を 3 つ以上考えよ。
- マルチプロセスで作成した場合はマルチスレッドに書き換えよ。
- マルチスレッドで作成した場合はマルチプロセスに書き換えよ。

## 4 HTTP プロキシサーバ作成

- HTTP サーバを改造して、リクエストを外部のサーバに転送する (中継する) プログラムを作成せよ。
- 作成したプログラムがボトルネックになっていないことを確認せよ。
  - あるサイトをプロキシ経由の有無で比較すると良い。
- 複数クライアントでもボトルネックにならないことを確認する。  
少なくとも 4 クライアントまで試すこと。

## 5 libpcap 応用プログラムの作成

- pcap\_loop を使って、あるインターフェイスに流入する ethernet フレーム数を計上せよ。
- フレームレートを算出するよう変更せよ。
- pcap\_next\_ex を使って書き換えよ
- TCP を観測し、コネクション開始レートを算出するプログラムを作成せよ。
- 並列同時コネクション数を算出するプログラムを作成せよ。

## 6 ハッシュを用いた連想配列

- 文字列をキー、整数 (int) をバリューに持つ一次元配列をつくる。
- 単純な探索を想定した追加と削除の機能を付けよ。
- 配列長を超える追加の際に、配列の領域を再確保するよう変更せよ。  
外部記憶を使うことは想定しなくてよい
- ハッシュ関数の結果で配列中の場所を決めるよう変更せよ。  
場所がぶつかった場合 (使用中の場合) には隣を使うこと。
- 削除ができるか確認せよ。
- キーが char\* になるよう変更せよ。
- キーが void\* になるよう変更せよ。  
void\* は内容がわからない、内容表示の動作に注意せよ。

## 7 辞書を使ったテンプレート展開

- 文面定義ファイル、辞書内容定義ファイルを読み、辞書内容を文面上に展開するプログラムを作成せよ。  
辞書は、前出の連想配列を使ってよい。他の実現方法でもかまわない。
- 辞書内容定義ファイルを複数与えて、複数の結果が得られるよう変更せよ。

## 参考文献

- [1] D.Comer, D.Stevens: Internetworking with TCP/IP Volume 3: Client-server programming and applications (Linux/POSIX Socket Version) ISBN 0-13-032071-4
- [2] W.Stevens, B.Fenner, A.Rudoff: UNIX Network Programming Volume 1 (Third Edition): The Sockets Networking API, ISBN 0-13-141155-1