

郵便番号クライアント・仕様/プロトコルの拡張

樋口さぶろお

龍谷大学理工学部数理情報学科

応用プログラミング☆実習 L13(2017-12-26 Tue)

最終更新: Time-stamp: "2017-12-26 Tue 17:47 JST hig"

今日の目標

- 1回の接続で複数回の recv/send を行う TCP クライアント-サーバのプログラムが書ける.
- fgets を 1行の入力を読み込むのに使える
- 機能に応じたプロトコルを設計できる



<http://hig3.net>

ソケットサーバ (疑似コード)

```

1  s=socket 作成;
2  bind(s ポート番号);
3  listen(s);
4  while(1){
5      ws=accept(s);
6      while(1){
7          recv(ws, リクエスト);
8          send(ws, レスポンス);
9          if(終了命令) break;
10     }
11     close(ws);
12 }

```

クライアントを待たせておいて、次々に `serve` するプログラム。
同時に `serve` することはできない。

また、モンスタークライアントが来ると、サーバプロセスが異常終了して、
待ってるクライアントに `serve` できなくなることがある。

これを防ぐには、クライアントごとにサーバプロセスを複製する (`fork` する) ことがかんがえられる

計算機システム II

サーバ群を統括するスーパーサーバ `inetd`, `xinetd` という考え方もある。

ソケットクライアント (疑似コード)

```
1  socket 作成;  
2  connect で IPアドレス:ポートに接続;  
3  while(1){  
4      ユーザから入力を受け取る  
5      if(終了命令) break;  
6      send リクエスト;  
7      recv レスポンス;  
8      レスポンス結果を表示;  
9  }  
10 close;
```

関数 fgets

ファイル入出力などから、改行文字までを文字列変数に読み込む。空白の位置やフォーマットによらない。

```
char *fgets(char line [], int size, FILE *fp)
```

fp から size-1 番目まで、または最初の改行まで読んで、それを line に入れる。size 番目に、または改行の直後に終端文字 '\0' を書く。

- line 保存先の文字列変数
- size s に保存する上限の長さ
- fp 読み込み先のストリーム。例: stdin=キーボード
- 返り値 成功なら line へのポインタ, 最初の文字を読もうとして EOF(=end of file) に到達したら NULL, 他の失敗も NULL.

SIZE=10 のとき

stdin からの入力 → fgets 実行後の line

0	1	2	3	4	\n
---	---	---	---	---	----

→

0	1	2	3	4	\n	\0			
---	---	---	---	---	----	----	--	--	--

0	1	2	3	4	5	6	7	8	9	0	1	\n
---	---	---	---	---	---	---	---	---	---	---	---	----

→

0	1	2	3	4	5	6	7	8	\0
---	---	---	---	---	---	---	---	---	----

EOF

→ line=不定, 返り値=NULL

fgets の典型的使用

```

1 char line[SIZE];
2 char *np;
3
4 if (fgets(line, sizeof(line), stdin) == NULL)
5     die("fgets"); /* 何か失敗 */
6 if ((np = strchr(line, '\n')) != NULL){
7     ; /* 改行まで読んでた! 成功 */
8 } else {
9     while(getchar() != '\n'); /* 途中で SIZE まで来たら改行まで捨てる
10 }
```

特別な IP アドレス

IP アドレス, ホスト名

127.0.0.1 localhost

はループバックアドレス. あるホスト上で使われたときは自分自身 (への通信) を意味する.

IP アドレスはコンピュータ (ホスト) を一意的に区別する, と言ったが例外あり. 127.0.0.1 localhost は '自分'. 'プライベートアドレス' 10.*.*.*, 192.168.*.* (説明してない記法) 172.16.0.0/12 は内線のようなもので, '閉じた' 組織内では自由に使ってよい=複数の組織内で同じ IP アドレスが使われることがある.

テスト用サーバ/クライアント

サーバを書いているとき、低機能でもバグのないクライアントがあると便利。クライアントを書いているとき、低機能でもバグのないサーバがあると便利。

そういうテスト用の関数やモジュールを、ドライバー (呼び出し側代理), スタブ (呼び出され側代理) という。

クライアント

```
$ telnet serverip serverport
```

- 標準入力 (タイプした) 文字列を, そのままサーバに send する.
- recv した文字列を, そのまま標準出力する (画面に印字).

サーバ

echo サービス (RFC862)

- recv した文字列を, そのまま send する (こだま)

常時 133.83.83.6:7 が使えます (学内のみ). ユーザが起動する必要なし.

機能とプロトコル

機能	プロトコル		サーバ内の処理
	リクエスト	レスポンス	
1 個の郵便番号の住所を表示する	search 整数 …	found 郵便番号 住所 notfound	int LSearch(...) を 1 回使用
練習	ignore 整数 …	notfound	
近い番号表示	greaterthan 整数	found 郵便番号 住所	別の関数を使用
—	上にあてはまらないもの	error	
⋮			

- クライアント-サーバ間の通信規則をあいまいさなく定めたもの. リクエストとレスポンスの意味とあいまいさのない書式
- プロトコルに従うなら, クライアント, サーバの実装を交換しても機能する (はず). 例: telnet 対 zip-client

RFC でのプロトコルの仕様の書き方

和訳 (<http://srgia.com/docs/rfc867j.html>) を引用。

仕様 (specification) で定めている部分, 実装 (implementation) に任せる部分がある。

Network Working Group
Request for Comments: 867

J. Postel
ISI
May 1983

Daytime プロトコル

この RFC は ARPA インターネットコミュニティのための標準を規定する。Daytime プロトコルの実装を選択する ARPA インターネット上のホストは、この標準を採用し、実装することを期待される。

daytime サービスは有効なデバッグング・計測のツールのひとつである。daytime サービスは入力に関係なく現在の日付と時刻とを文字列として単純に送信する。

TCP ベースの Daytime サービス

ひとつ目の daytime サービスは、TCP 上のコネクションベースアプリケーションとして定義される。サーバーは TCP ポート 13 で TCP 接続を監視する。接続が確立すると、現在の日付と時刻が ASCII 文字列として返される (受信したデータは捨てられる)。サービスはその文字列を送信した後、接続を閉じる。

(中略)

Daytime の文法

daytime のための具体的な構文は存在しないが、ASCII の印刷可能文字・空白・改行・復帰に限定することを推奨する。日付・時刻は一行にするべきである。

よく知られている構文:

```
Weekday, Month Day, Year Time-Zone
```

例:

```
Tuesday, February 22, 1982 17:37:43-PST
```

SMTTP で使用されているもうひとつのよく知られている構文:

```
dd mmm yy hh:mm:ss zzz
```

例:

```
02 FEB 82 07:59:01 PST
```

注: 機械的に処理しやすい時刻のためには Time プロトコル (RFC-868) を使用すること。

実習の進め方計画

L13

- 4人チーム (=指定のサーバ2名, クライアント2名) で 課題 p131 (チームで1個を Moodle に提出)
- 4人チームで 課題 p132 用の 機能-プロトコル-(...) の表 (チームで1個を Moodle に提出)

L14

- サーバ2名, クライアント2名のサブチームに分かれて, 課題 p132 のプロトコルを別々に実装 (サブチーム別提出)

L15

- クライアントサーバ間の通信の検証, 調整 (プロトコルの修正), プログラム提出 (チーム別), 報告書 (参加者別) 提出.

お知らせ

- 樋口オフィスアワー月 3.5 – 4.5(1-502), 金 4(1-502)
- 今日の昼休みにお弁当食べられる休憩室: 1-534.
- L14 の最初でも非参照のテストやります
 - ▶ fgets の振る舞い
 - ▶ 文字列関数の分類
 - ▶ サーバの 2 重のループの意味
- Learn Math Moodle に予習問題を載せるので、それで準備してね.