

応用プログラミング ex3 演習課題

1. Echo Client/Server をヘッダと関数でファイル分割 (前半の復習)

- echoClient.c, echoServer.c, dieWithError.c handleTCPClient.c の 定数定義(//**defn**//文) および関数のプロトタイプ宣言を一つにまとめた **echo.h** という名前のヘッダファイルを作成しなさい。
 - 作成した**echo.h** は、上記の各ソースファイルでインクルードしなさい。
 - ヘッダファイルとそのインクルードについては、[高橋先生の前半第4回講義資料](#)でも解説されています。

2. Echo Client/Server 用の Makefile を作成 <color white,red/white>[TA]</color>

- **echoClient** と **echoServer** の実行ファイルを作るための **Makefile** を作成しなさい。
 - 作成した Makefile を使って make コマンドを実行すると

```
make echoClient
```

で echoClient の実行ファイルを、

```
make echoServer
```

で echoServer の実行ファイルが作成され、また、

```
make
```

あるいは、

```
make all
```

では echoClient と echoServer の両方の実行ファイルが作成されるものとします。更に、

```
make clean
```

を実行すると make コマンドで作成される全てのオブジェクトファイル echoClient, echoServer, *.o が消去されるものとします。// 上記、1. で作成したインクルードファイル **echo.h** の依存関係も記述しなさい。

- %作成した Makefile の内容を (必要があれば make での実行結果を示しながら) [TA] に説明しなさい。-%

3. 実行状況の出力表示 <color white,red/white>[TA]</color>

- 次の各ポイントで、**echoClient** の状態を表示するように **echoClient.c** を変更しなさい。その際、各ポイントで echoClient の実行を1秒ずつ休止しなさい。

1. サーバIPアドレス、ポート、送信文字列の確定後
2. ソケット作成後
3. サーバ接続完了後
4. 送信文字列長の確定後
5. サーバへの文字列送信後
6. サーバ接続切断後

- サーバIPアドレス、ポート番号、送信文字列が確定した後の状況表示では、それぞれの値を表示しなさい。その際、コマンドライン引数文字列ではなく、内部変数に代入した後の値を表示すること（例えば `argv[1]` 等ではなく `servIP` などの値を用いる）。
- プログラムを途中で休止するためには `unistd.h` で定義されている関数 `sleep()` を利用しなさい。`sleep()` は引数として `unsigned int` 型の数値を取り、与えられた数値の秒数だけプログラムを休止します。1秒間休止する場合は次のように利用します。

```
sleep(1);
```

- 実行結果の一例を示す。

```
[sano@s1542f160 ~]$ ./echoClient 127.0.0.1 ABC 5007
*Variables:
  Server IP: 127.0.0.1
  Server Port: 5007
  echo String: ABC          <- IPアドレス、ポート、送信文字列を表示して、1秒休止
*Socket Ready.           <- ソケット作成して、1秒休止
*Server Connected.       <- サーバに接続して、1秒休止
*echo Length: 3          <- 送信文字列長表示、1秒休止
*Sent String: ABC        <- 送信した文字列を表示、1秒休止
Received: ABC
*Close Connection.      <- 接続切断して終了
[sano@s1542f160 ~]$
```

- %必要に応じてプログラムソースを参照しながら、背後でどのような処理と通信が行われているかを実行結果と共に TA に説明しなさい。%
- コメント付きの `echoClient.c`
 - [echoclient_wc.c](#)

X. 演習課題の予告) 送信文字列をキーボードから入力して、繰り返し送信できるようにする。

1. 送信文字列の設定をコマンドライン引数から、キーボード入力へと変更する。
 - 引数処理の変更
 - 送信文字列 `echoString` の宣言時領域確保（最大送信文字数の定義 (`echo.h`)
2. サーバ接続を維持したまま、複数回繰り返し `echoServer` との通信を行う。
 - 送信文字列入力が **quit** であれば、終了して接続を切断する。
3. 送信文字入力での文字数制限（キーボード入力が最大送信文字数を超えた場合の処理）

From:

<https://www-slab.math.ryukoku.ac.jp/> - **www-slab.math**

Permanent link:

<https://www-slab.math.ryukoku.ac.jp/lecture/apro/2016/ex3>



Last update: **2019/09/23 13:50**