

SkyEye with TINET

OpaqueRPC サンプルのビルドと実行の手順

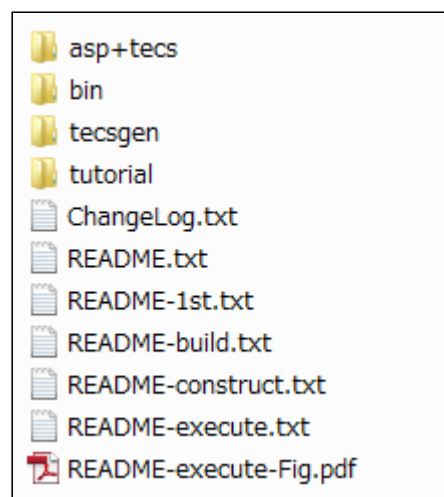
本書は README-build.txt と README-execute.txt の skyeye 版同士を接続する場合について、抜粋しまとめたものです

TAP-Win32 の設定方法については、README-TAP.pdf をご覧ください。2つTAPを作成し、それぞれ”mytap”, “mytap2” に名前を変更して、ブリッジ接続しておきます。

2011年6月4日

tecs_packageの中身

- asp+tecs TOPPERS/ASPカーネル
- bin バイナリファイル
- tecsgen TECSのインタフェースジェネレータ



skyeeye版サーバのビルド

- % cd tecs_package/asp+tecs
- % ln -s ../bin/tecsgen.exe .
- % cd sample_server
- % make tecs
- % make depend
- % make all

skyeeye版クライアントのビルド

別のウィンドウを用いて

- % cd tecs_package/asp+tecs/sample_client
- % make tecs
- % make depend
- % make all

IPアドレスの調整

- サーバとクライアントのIPアドレスの設定

- ~/tecs_package/asp+tecs/sample_server/tinet_app_cofig.h

```
267|#define IPV4_ADDR_LOCAL                MAKE_IPV4_ADDR(192,168,99,1)
```

- ~/tecs_package/asp+tecs/sample_client/tinet_app_cofig.h

```
267|#define IPV4_ADDR_LOCAL                MAKE_IPV4_ADDR(192,168,99,2)
```

- ~/tecs_package/asp+tecs/sample_client/tSimpleClient.c

```
136|                                     dstaddr.ipaddr = MAKE_IPV4_ADDR(192,168,99,1);
```

※複製したのでIPアドレスに重複が発生する為必ずどちらかを変更しなければいけない
また、サーバの設定したIPアドレスにクライアントから接続するので
上と下の2つのIPアドレスは同一である必要がある
注意) これを変更した場合、再度ビルドすること

skyeye.confの編集

- 必要に応じてskyeyeの設定ファイルであるskyeye.confの記述を変更する

- ~/tecs_package/asp+tecs/sample_server/skyeye.conf

```
11|net: type=rtl8019, mac=0:ff:15:6f:6d:b4,  
    ethmod=tuntap, tapname=mytap, hostip=192.168.99.3
```

- ~/tecs_package/asp+tecs/sample_client/skyeye.conf

```
11|net: type=rtl8019, mac=0:ff:15:6f:6d:b5,  
    ethmod=tuntap, tapname=mytap2, hostip=192.168.99.4
```

- “tapname=mytap”が無ければ記述を追加する。
 - “mytap”の部分に接続したいTAP名を記述する
 - TAP名がmytapかmytap2ではなければTAP名の部分を修正する

※sample_clientとsample_serverのMACアドレスの値は同一だとエラーが
起こるので違う値にする。
また、同じTAPに接続してしまった際もエラーが起こるのでそれについても注意する。

skyeyeの実行

- 各コマンドライン上で

```
% ../../bin/skyeye.exe -e asp.exe
```

- ネットワーク接続を開き、skyeye.confで指定したTAPをブリッジ
- サーバで

```
[NET/TIMER:3] started.  
[TCP OUTPUT:6] started.
```

- が出る前にクライアントがopenするなら
- %~/tecs_package/asp+tecs/sample_client/tSimpleClient.c

```
131|      dly_tsk( 300 );
```

- 300の部分を変更する(300なら0.3sec)

skyeyeの実行結果

サーバ

```
[ETHER INPUT: 5] started on MAC Addr: 00:ff:15:6f:6d:b4.  
[ETHER OUTPUT:4] started.
```

```
TINET 1.5.1 (Apr 21 2011, 13:44:25)  
Copyright (C) 2001-2009 by Dep. of Computer Science and  
Engineering  
Tomakomai National College of Technology,  
JAPAN
```

```
[NET/TIMER:3] started.  
[TCP OUTPUT:6] started.
```

```
func1: 10  
hello  
func2  
message for func3  
func3  
this  
is  
a  
pen  
.  
sta[0]->a=11 sta[0]->b=121  
sta[0]->len=16
```

クライアント

```
[ETHER INPUT: 5] started on MAC Addr: 00:ff:15:6f:6d:b5.  
[ETHER OUTPUT:4] started.
```

```
TINET 1.5.1 (Apr 21 2011, 13:47:11)  
Copyright (C) 2001-2009 by Dep. of Computer Science and  
Engineering  
Tomakomai National College of Technology,  
JAPAN
```

```
[NET/TIMER:3] started.  
[TCP OUTPUT:6] started.
```

```
starting client  
opening  
SimpleClient: opening Channel  
main1  
SimpleClient: cCall_func1  
SimpleClient: cCall_func2  
SimpleClient: cCall_func3  
SimpleClient: cCall_func4  
main5  
SimpleClient: cCall_func5
```

注意)この実行には、かなり時間がかかります(Cygwin 版は、一瞬で終了します)

応用

- このサンプルは sSimple.cdl に書かれている sSimple シグニチャの関数を呼び出します
 - 網羅的なテスト用なので、難しい定義が含まれています
- 自分の関数を呼び出してみる
 - sSimple.cdl の sSimple シグニチャに関数を追加、削除する
 - sample_server/tSimpleServer.c、sample_client/tSimpleClient.c に関数を追加削除する
 - sSimple.cdl を変更後 make tecs するとテンプレートが作成されるので、tecsmerge でマージする
- これだけで通信プログラムを改変できるのがRPCのメリットです