STARS Client の例（Python 版）

2011/06/18 KEK-PF 小菅隆

”pyuserclient.py”及び“pyioclient.py”は、Python による簡単な STARS Client のサンプルです。私自身 Python 超初心者なので、非常に安易なサンプルになっています。たとえば、Keyword のチェックの部分は通常では複数のキーワードを用意して行いますが、ここでは keyword を 1 つだけにしてとにかく簡単に接続を行っています。

準備

他のクライアント同様、”pyuserclient.key”及び”pyioclient.key”を STARS Server の takaserv-lib の下にコピーしてください。なお、サンプルプログラムではこれらのファイルの読み込みは行っていません。
次に STARS Server(takaserv)を予め起動しておいてください。なお、STARS Server が動作する PC はサンプルプログラムを走らせる PC と同一の PC であると想定しています。

User Client 型サンプル

”pyuserclient.py”は I/O Client にコマンドを送ったりする、User Client のサンプルです。
このサンプルは STARS Server に接続し、hello コマンドを STARS Server に送信、答えを表示します。また、”pyioclient.py”にも hello コマンドを送信しますが、”pyioclient.py”を起動していない場合は、エラーが返ります。

pyuserclient.py ソースファイル 1/4

```
#! /usr/bin/python
# TEST user client type.

import socket

mynode    = 'pyuserclient'  #Node name of this client
server    = 'localhost'     #Host name of STARS Server
port      = 6057            #STARS server port. Default is 6057

# Connect STARS Server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((server, port))
```

STARS Server へは TCP/IP Socket の Client として接続します。
ポート番号は 6057 を使用します。

次に STARS Server が送信するランダムな番号を受け取り、自ノード名と、受け取ったこの番号に応じたキーワードを STARS Server に送信しますが、今回のサンプルでは常に同じキーワードを使用するようにしています。(yuserclient.key ファイルには“stars”というキーワードが一つだけ設定されています。

pyuserclient.py ソースファイル 2/4

```
# Get key number. This number means, which keyword should be used (0 is 1st).
# But only 1 keyword "stars" is prepared in this example (see pyuserclient.key).
# It means "stars" is used as the keyword at any time.
keynum = s.recv(1024)
print 'Key number:', keynum,
keyword = 'stars'

# Create connection with STARS server
s.sendall(mynode + ' ' + keyword + '¥n')
msg = s.recv(1024)
print 'Result:', msg,
```

上のように自ノード名とキーワードを STARS Server に送信し、STARS Server のチェックにパスすると、“自ノード名 Ok:”のようなメッセージが返されます。なお、ここでは予め自ノード名を設定してます。

STARS Server にコマンド hello 及び gettime を送信してみます。それぞれ“@”から始まるリプライが返されます。

pyuserclient.py ソースファイル 3/4

```
# Send hello command to STARS server
s.sendall('System hello¥n')
msg = s.recv(1024)
print msg,

# Send gettime command to STARS server
s.sendall('System gettime¥n')
msg = s.recv(1024)
print msg,
```

次にもう一つのサンプルプログラム “pyioclient.py”にコマンド hello を送ってみます。“pyioclient.py”では hello コマンドを受け取るようになっているため、Reply メッセージを返してきます。なお、サンプル最後の部分ではキーボード入力を待ち、その後 Socket を Close しています。

pyuserclient.py ソースファイル 4/4

```
# Send hello command to pyioclient.
# An error command will be returned if pyioclient is not running.
s.sendall('pyioclient hello¥n')
msg = s.recv(1024)
print msg,

print 'Hit enter to end.'
msg = raw_input()
s.close()
```

以下は “pyuserclient.py”実行の様子です。

pyuserclient.py の実行

```
$ python pyuserclient.py
Key number: 2199
Result: System>pyuserclient Ok:
System>pyuserclient @hello Nice to meet you.
System>pyuserclient @gettime 2011-06-18 00:31:05
pyioclient>pyuserclient @hello nice to meet you.
Hit enter to end.
```

I/O Client 型サンプル

”pyioclient.py”は User Client からコマンドに応じて答えを返す I/O Client 型のサンプルです。実際の STARS を使用したシステムでは I/O Client はハードウエアなどを制御するためのデバイスドライバー的存在になります。いくつかのシステムでは、この種のソフトウエアを TCP/IP Socket のサーバとしている場合がありますが、STARS ではあくまでも Client として STARS に接続します。接続の方法は User Client となんら変わりはありません。

pyuserclient.py ソースファイル 1/3

```
#! /usr/bin/python
# TEST IO client type.

import socket

mynode    = 'pyioclient'  #Node name of this client
server    = 'localhost'     #Host name of STARS Server
port      = 6057            #STARS server port. Default is 6057

# Connect STARS Server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((server, port))

# Get key number. This number means, which keyword should be used (0 is 1st).
# But only 1 keyword "stars" is prepared in this example (see pyuserclient.key).
# It means "stars" is used as the keyword at any time.
keynum = s.recv(1024)
print 'Key number:', keynum,
keyword = 'stars'

# Create connection with STARS server
s.sendall(mynode + ' ' + keyword + '¥n')
msg = s.recv(1024)
print 'Result:', msg,
```

以上、接続までのプロセスは自ノード名は違いますが、他は “pyuserclient.py” と同様です。

接続が完了すると、STARS Server からメッセージが送られてくるのを待ち、コマンド(もしくはリプライ、イベント)に応じて処理を行います。なお、while により処理を繰り返します。

pyuserclient.py ソースファイル 2/3

```
print 'Waiting commands from STARS server. ^C to end.'
# Handle commands
while True:
#Extract From, To, Message
    msg = s.recv(1024)
    msg = msg.rstrip('¥n')
    msg = msg.replace('>', ' ')   #Replace "From>To Command" into "From To Command"
    cmd = msg.split()
    print 'From:   ', cmd[0]
    print 'To:     ', cmd[1]
    print 'Message:', cmd[2]
```

上は、全メッセージから、from、to、Message(Command、Reply、Event)を抽出しています。

次に Message に応じて処理を行います。なお、以下の 9 行目ではリプライあるいはイベントが送られてきた際には無視するようにしています。STARS では、リプライ及びイベントに関して決してエラー等のメッセージを返してはいけません。
なお、“pyuserclient.py”は terminate コマンドを受け取ると while ループから抜け(6 行目)、制御を停止します。

pyuserclient.py ソースファイル 3/3

```
#Handle commands.
    if cmd[2] == 'hello':
        s.sendall(cmd[0] + ' @hello nice to meet you.¥n')
    elif cmd[2] == 'help':
        s.sendall(cmd[0] + ' @help hello terminate.¥n')
    elif cmd[2] == 'terminate':
        s.sendall(cmd[0] + ' @terminate Ok:¥n')
        break
    elif cmd[2][0] == '_' or cmd[2][0] == '@':
        continue
    else:
        s.sendall(cmd[0] + ' @' + cmd[2] + ' Er: Bad command or parameter.¥n')

s.close()
```

以下は “pyioclient.py”実行の様子です。

pyuserclient.py の実行

```
$ python pyioclient.py
Key number: 5982
Result: System>pyioclient Ok:
Waiting commands from STARS server. ^C to end.
From:    term1
To:      pyioclient
Message: hello
From:    term1
To:      pyioclient
Message: terminate
```

以上