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