小顾de杂记

0(∩_∩)0~|Across the great wall we can reach the every corner in the world.

读取中

expect实现交互初步

Posted on 2010年12月27日 4,625 阅读

使用expect脚本可以完成一定程度上的自动交互，不过又要学习了。
关于expect的简单介绍。可以看这里 http://zh.wikipedia.org/zh/Expect
下面我记录一下基本用法

expect脚本可以接受从bash传递过来的参数.可以使用[lindex $argv n]获得，n从0开始，分别表示第一个,第二个,第三个....参数
看下面的expect脚本的例子

```
#!/usr/bin/expect
set username [lindex $argv 0]
set password [lindex $argv 1]
set server [lindex $argv 2]
send_user  "UserName is $username\n"
send_user  "PassWord is $password\n"
send_user  "Server is $server\n"
send_user "Total arg num is $argc\n"
send_user "last but one arg is[lindex $argv [expr $argc-1]]\n"
if { $argc != 3 && $argc != 2 } {
    send_user "Usage:username password \[server\] \n"
    send_user "\tthe default server is 211.65.64.1 \n"
    exit
}
```

执行这个文件./launch.exp 1 2 3
屏幕上就会分别打印出参数
send_user用来发送内容给用户。

参数运用方面还有很多技巧
比如$argc 存储了参数个数，args被结构化成一个列表存在argv。$argv0 被初始化为脚本名字。
除此之外，如果你在第一行(#!那行)使用-d (debug参数)，可以在运行的时候输出一些很有用的信息
比如你会看见

argv[0] = /usr/bin/expect argv[1] = -d argv[2] = ./launch.exp argv[3] = 1 argv[4] = 2 argv[5] = 3

使用这些也可以完成参数传递

另外在spawn后面加一个interact,会从自动交互状态退出到输入状态，由用户完成剩余的操作.
ecpect也支持使用逻辑结构。基本语法和大多数shell语言；类似，不过使用{}而不是()
另外花括号前后的空格不容小觑，不写就会报错.

-- 是用来为划定选项尾的。
当需要像使用选项一样传一个参数，但希望这个参数不要被当作选项解释时，就需要用到这个选项。当阻止其他选项时，可以把它放在”#!”行中
#!/usr/local/bin/expect --
会让所有参数（包括脚本文件名）都存储在argv中。

openVPN 59min重连脚本
学校的OpenVPN服务器非常恶心，每当用户登录时间经过60 min 09 s后就把用户踢下线。为了解决这个问日，我使用这个expect脚本来完成这个目的

```
#!/usr/bin/expect
#openVPN 59min 自动重连脚本
#By http://ihipop.info
#2010-12-26 11:21

set timeout 30
set username [lindex $argv 0]
set password [lindex $argv 1]
#set server [lindex $argv 2]
#send_user  "UserName is $username\n"
#send_user "Filename is $argv0 ,and args No.1-2 is :[lrange $argv 1 2] \n"
```

```
#send_user "Total arg num is $argc\n"
if { $argc != 3 && $argc != 2 } {
    send_user "Usage:username password \[server\] \n"
    send_user "\tthe default server is 211.65.64.1 \n"
    exit
}
spawn openvpn --config  rnas-school.ovpn
expect "Enter Auth Username:"
send "$username\r"
expect "Enter Auth Password:"
send "$password\r"
expect "Initialization Sequence Completed"
#interact
#等待59min后结束spawn
#exec sleep 3540
set timeout 3540
exit
```

然后上层用一个bash的永真循环来调用这个脚本，传递参数即可。

并行结构
上面的脚本是串行结构，其结果就是，当事态不按照他的发展来的时候，比如出现了异常输出输入请求，但是expect没有匹配到，expect就会一直等待到timeout(默认是10s)
所以为了增加程序健壮性，应该考虑使用并行结构

```
#!/usr/bin/expect
#openVPN 59min 自动重连脚本
#By http://ihipop.info
#2010-12-26 11:21

set timeout 30
set username [lindex $argv 0]
set password [lindex $argv 1]
#set server [lindex $argv 2]
#send_user  "UserName is $username\n"
#send_user "Filename is $argv0 ,and args No.1-2 is :[lrange $argv 1 2] \n"
#send_user "Total arg num is $argc\n"
if { $argc != 3 && $argc != 2 } {
    send_user "Usage:username password \[server\] \n"
    send_user "\tthe default server is 211.65.64.1 \n"
    exit
}
spawn openvpn --config  rnas-school.ovpn
expect {
 "Enter Auth Username:" {
  send "$username\r"
  exp_continue
 }
 "Enter Auth Password:" {
   send "$password\r"
   exp_continue
 }
 "Initialization Sequence Completed" {
   #interact
   set timeout 3540
   #set timeout 3
   exp_continue
 }
 "AUTH: Received AUTH_FAILED control message" {
   #interact
   send_user "AUTH_FAILED\n"
   exit
 }
 eof {
   send_user "eof\n"
   exit
 }

 timeout {
   send \003
   send_user "timeout \n"
   exit
 }

}
```

上面看到的是把expect脚本单独运行的例子
其实也可以和bash脚本结合的很好
看下面的例子

```
#!/bin/bash
auto_smart_ssh () {
    expect -c "set timeout -1;
                spawn ssh -o StrictHostKeyChecking=no $2 ${@:3};
                expect {
                    *assword:* {
                        send $1\r;
                        expect {
                            *denied* {
                                exit 2;
                            }
                            eof
                        }
                    }
```

```
15                      eof {
16                          exit 1;
17                      }
18                  }
19                  "
20      return $?
21  }
22
23  auto_smart_ssh passwd user@host ls /var
24  echo -e "\n---Exit Status: $?"
```

readmore:http://linux.die.net/man/1/expect

Author Info :

- From:expect实现交互初步
- URL:http://blog.ihipop.info/2010/12/1949.html
- Please Reserve This Link,Thanks!

相关阅读

- OpenVPN的topology subnet模式，最节约地址的方式
- Debian SID 中OpenVPN 无法在status log正确记录客户端IPV6地址导致freeradius不统计IPV6用户流量的临时解决办法
- OpenVPN With FreeRadius Via Radiusplugin on Debian
- OpenVPN客户端在UDP模式下的explicit-exit-notify
- OpenVPN & reneg-sec & 一小时断线一次 & Solution

此条目由**ihipop**发表在**Linux Manger**分类目录，并贴了**expect**、**OpenVPN**、**spawn**标签。将**固定链接 [/2010/12/1949.html]** 加入收藏夹。