

expect - 自动交互脚本

expect参数

- expect教程中文版
- expect中文手册
- expect说明

启用选项

- -c :执行脚本前首先执行的命令，可多次使用。
- -d :debug模式，可以在运行时输出一些诊断信息，与在脚本开始处使用 `exp_internal 1` 相似。
- -D :启用交换调试器,可设一整数参数。
- -f :从文件读取命令，仅用于使用`#!`时。如果文件名为“-”，则从`stdin`读取(使用“/”从文件名为-的文件读取)。
- -i :交互式输入命令，使用“exit”或“EOF”退出输入状态。
- -- :标示选项结束(如果你需要传递与expect选项相似的参数给脚本时)，可放到`#!`行：`#!/usr/bin/expect --`。
- -v :显示expect版本信息。

常用命令

```
1 # 命令行参数
2 # $argv, 参数数组, 使用[index $argv n]获取, $argv 0为脚本名字
3 # $argc, 参数个数
4 set username [lindex $argv 1] # 获取第1个参数
5 set passwd [lindex $argv 2]  # 获取第2个参数
6
7 set timeout 30 # 设置超时
8
9 # spawn是expect内部命令, 开启ssh连接
10 spawn ssh -l username 192.168.1.1
11
12 # 判断上次输出结果里是否包含"password:"的字符串, 如果有则立即返回, 否则就等待一段时间(timeout)后返回
13 expect "password:"
14
15 # 发送内容ispass(密码、命令等)
16 send "ispass\r"
17
18 # 发送内容给用户
19 send_user "$argv0 [lrange $argv 0 2]\n"
20 send_user "It's OK\r"
21 # 执行完成后保持交互状态, 控制权交给控制台(手工操作)。否则会完成后会退出。
22 interact
```

命令介绍

- close:关闭当前进程的连接。
- debug:控制调试器。
- disconnect:断开进程连接(进程仍在后台运行)。

```
• 定时读取密码、执行priv_prog
1 send_user "password?\ "
2 expect_user -re "(.)\n"
3 for {} 1 {} {
4     if {[fork] != 0} {sleep 3600; continue}
5     disconnect
6     spawn priv_prog
7     expect Password:
8     send "$expect_out(1,string)\r"
9     ...
10    exit
11 }
```

- exit:退出expect。
- exp_continue [-continue_timer]:继续执行下面的匹配。
- exp_internal [-f file] value:

expect范例

- 自动telnet会话

```
1 #!/usr/bin/expect -f
2 set ip [lindex $argv 0] # 接收第1个参数,作为IP
3 set userid [lindex $argv 1] # 接收第2个参数,作为userid
4 set mypassword [lindex $argv 2] # 接收第3个参数,作为密码
5 set timeout 10 # 设置超时时间
6
7
8 # 向远程服务器请求打开一个telnet会话, 并等待服务器询问用户名
9 spawn telnet $ip
10 expect "Username:"
11 # 输入用户名, 并等待服务器询问密码
12 send "$userid\r"
13 expect "password:"
14 # 输入密码, 并等待键入需要运行的命令
15 send "$mypassword\r"
16 expect "*"
17 # 输入预先定好的密码, 等待运行结果
18 send "$mycommand\r"
19 expect "*"
20 # 将运行结果存入到变量中, 显示出来或者写到磁盘中
21 set results $expect_out(buffer)
22 # 退出telnet会话, 等待服务器的退出提示EOF
23 send "exit\r"
24 expect eof
```

- 自动建立FTP会话

```
1 #!/usr/bin/expect -f
2 set ip [lindex $argv 0] # 接收第1个参数,作为IP
3 set userid [lindex $argv 1] # 接收第2个参数,作为userid
4 set mypassword [lindex $argv 2] # 接收第3个参数,作为密码
5 set timeout 10 # 设置超时时间
6
7 # 向远程服务器请求打开一个FTP会话, 并等待服务器询问用户名
8 spawn ftp $ip
9 expect "Username:"
10 # 输入用户名, 并等待服务器询问密码
11 send "$userid\r"
12 expect "password:"
13 # 输入密码, 并等待FTP提示符的出现
14 send "$mypassword\r"
15 expect "ftp>"
16 # 切换到二进制模式, 并等待FTP提示符的出现
17 send "bin\r"
18 expect "ftp>"
19 # 关闭ftp的提示符
20 send "prompt\r"
21 expect "ftp>"
22 # 下载所有文件
23 send "mget *\r"
24 expect "ftp>"
25 # 退出此次ftp会话, 并等待服务器的退出提示EOF
26 send "bye\r"
27 expect eof
```

- 自动登录ssh执行命令

```
1 #!/usr/bin/expect
2 set IP [lindex $argv 0]
3 set USER [lindex $argv 1]
4 set PASSWD [lindex $argv 2]
5 set CMD [lindex $argv 3]
6
7 spawn ssh $USER@$IP $CMD
8 expect {
9     "(yes/no)?" {
10         send "yes\r"
11         expect "password:"
12         send "$PASSWD\r"
13     }
14     "password:" {send "$PASSWD\r"}
15     "* to host" {exit 1}
16 }
17 expect eof
```

- 自动登录ssh

```
1 #!/usr/bin/expect -f
2 set ip [lindex $argv 0] # 接收第1个参数,作为IP
3 set username [lindex $argv 1] # 接收第2个参数,作为username
4 set mypassword [lindex $argv 2] # 接收第3个参数,作为密码
5 set timeout 10 # 设置超时时间
6
7 spawn ssh $username@$ip # 发送ssh请求
8 expect { # 返回信息匹配
9     "(yes/no) { send "yes\r"; exp_continue} # 第一次ssh连接会提示yes/no,继续
10     "password:" { send "$mypassword\r" } # 出现密码提示,发送密码
11 }
12 interact # 交互模式,用户会停留在远程服务器上
```

- 批量登录ssh服务器执行操作范例, 设定增量的for循环

```
1 #!/usr/bin/expect
2 for (set i 10) {$i <= 12} {incr i} {
3     set timeout 30
4     set ssh_user [lindex $argv 0]
5     spawn ssh -i .ssh/$ssh_user abc$i.com
6
7     expect_before "no)?" {
8         send "yes\r" }
9     sleep 1
10    expect "password*"
11    send "hello\r"
12    expect "*"
13    send "echo hello expect! > /tmp/expect.txt\r"
14    expect "*"
15    send "echo\r"
16 }
17 exit
```

- 批量登录ssh并执行命令, foreach语法

```
1 #!/usr/bin/expect
2 if {$argc != 2} {
3     send_user "usage: ./expect ssh_user password\n"
4     exit
5 }
6 foreach i {11 12} {
7     set timeout 30
8     set ssh_user [lindex $argv 0]
9     set password [lindex $argv 1]
10    spawn ssh -i .ssh/$ssh_user root@xxx.yy.com
11    expect_before "no)?" {
12        send "yes\r" }
13    sleep 1
14
15    expect "Enter passphrase for key*"
16    send "password\r"
17    expect "*"
18    send "echo hello expect! > /tmp/expect.txt\r"
19    expect "*"
20    send "echo\r"
21 }
22 exit
```

- 另一自动ssh范例, 从命令行获取服务器IP, foreach语法, expect嵌套

```
1 #!/usr/bin/expect
2 # 使用方法: script_name ip1 ip2 ip3 ...
3
4 set timeout 20
5 if {$argc < 1} {
6     puts "Usage: script IPs"
7     exit 1
8 }
9
10 # 替换你自己的用户名
11 set user "username"
12 #替换你自己的登录密码
13 set password "yourpassword"
14
15 foreach IP $argv {
16     spawn ssh $user@$IP
17
18     expect \
19         "(yes/no)?" {
20             send "yes\r"
21             expect "password:?" {
22                 send "$password\r"
23             }
24             "password:?" {
25                 send "$password\r"
26             }
27         }
28     # 替换你要执行的命令
29     send "last\r"
30     expect "\$?"
31     sleep 10
32     send "exit\r"
33     expect eof
34 }
```

- 批量ssh执行命令, 用shell调用tcsh方式、多进程同时执行

- tcsh - Simple shell containing Tcl interpreter

```
1 #!/bin/sh
2 # -*- tcl -*- \
3 exec tclsh $0 "$@"
4 package require Expect
5 set username [lindex $argv 0]
6 set password [lindex $argv 1]
7 set argv [lrange $argv 2 end]
8 set prompt "${%}\n${%} $"
9
10 foreach ip $argv {
11     spawn ssh -t $username@$ip sh
12     lappend ids $spawn_id
13 }
14 expect_before -i ids eof {
15     set index [lsearch $ids $expect_out(spawn_id)]
16     set ids [lreplace $ids $index $index]
17     if [llength $ids] exp_continue
18 }
19 expect -i ids "(yes/no)\n" {
20     send -i $expect_out(spawn_id) yes\r
21     exp_continue
22 } -i ids "Enter passphrase for key" {
23     send -i $expect_out(spawn_id) \r
24     exp_continue
25 } -i ids "assword:" {
26     send -i $expect_out(spawn_id) $password\r
27     exp_continue
28 } -i ids -re $prompt {
29     set spawn_id $expect_out(spawn_id)
30     send "echo hello; exit\r"
31     exp_continue
32 } timeout {
33     exit 1
34 }
35 }
```

- ssh登录过程常规提示文字

```
The authenticity of host '192.168.17.35 (192.168.17.35)' can't be established.
RSA key fingerprint is 25:e8:4c:89:a3:b2:06:ee:de:66:c7:7e:1b:fa:1c:c5.
Are you sure you want to continue connecting (yes/no)?
```

```
Warning: Permanently added '192.168.17.35' (RSA) to the list of known hosts.
```

```
Enter passphrase for key '/data/key/my_dsa':
```

```
Last login: Sun Jan 26 13:39:37 2014 from 192.168.11.143
```

```
[root@master003 ~]#
```

```
root@192.168.16.90's password:
```

```
Last login: Thu Jan 23 17:50:43 2014 from 192.168.11.102
```

```
[root@lvsmaster ~]#
```

- ssh自动登录expect脚本:ssh.expect

```
1 #!/usr/bin/expect -f
2 # Author:YuanXing
3 # Update:2014-02-08
4 if {$argc < 4} {
5     send_user "Usage:\n $argv0 IPaddr User Passwd Port Passphrase\n"
6     puts stderr "argv error!\n"
7     sleep 1
8     exit 1
9 }
10
11 set ip [lindex $argv 0]
12 set user [lindex $argv 1]
13 set passwd [lindex $argv 2]
14 set port [lindex $argv 3]
15 set passphrase [lindex $argv 4]
16 set timeout 6
17 if {$port == ""} {
18     set port 22
19 }
20 #send_user "IP:$ip,User:$user,Passwd:$passwd,Port:$port,Passphrase:$passphrase"
21 spawn ssh -p $port $user@$ip
22
23 expect_before "(yes/no)\n" {
24     send "yes\r"
25 }
26
27 expect \
28     "Enter passphrase for key*" {
29         send "$passphrase\r"
30         exp_continue
31     } " password:?" {
32         send "$passwd\r"
33         exp_continue
34     } "*(\n\${%})" {
35         interact
36     } "* to host" {
37         send_user "Connect failed!"
38         exit 2
39     } timeout {
40         send_user "Connect timeout!"
41         exit 2
42     } eof {
43         send_user "Lost connect!"
44         exit
45     }
46 }
```

- Mikrotik backup script using ssh and expect

<http://www.pmoghadam.com/homepage/HTML/mikrotik-backup-script-ssh-expect.html>

```
1 #!/bin/bash
2 # BY: Pejman Moghadam
3 # TAG: mikrotik, ssh, expect, lftp
4 # DATE: 2012-05-27 14:42:14
5
6 BACKUP_DIR="/var/backups"
7 HOSTNAME="192.168.88.1"
8 PORT="22"
9 USER="admin"
10 PASS="123456"
11 TMP=$(mktemp)
12 TODAY=$(date +%F)
13 FILENAME="$HOSTNAME-$TODAY"
14 PATH="/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin"
15
16 # create expect script
17 cat > $TMP << EOF
18 #exp_internal 1 # Uncomment for debug
19 set timeout -1
20 spawn ssh -p$PORT $USER@$HOSTNAME
21 match_max 100000
22 expect -exact "password:"
23 send -- "$PASS\r"
24 sleep 1
25 expect " > "
26 send -- "/export file=$FILENAME\r"
27 expect " > "
28 send -- "/system backup save name=$FILENAME\r"
29 expect " > "
30 send -- "quit\r"
31 expect eof
32 EOF
33
34 # run expect script
35 #cat $TMP # Uncomment for debug
36 expect -f $TMP
37
38 # remove expect script
39 rm $TMP
40
41 # download and remove backup files
42 # "xfer:clobber on" means overwrite existing files
43 cd ${BACKUP_DIR}
44 echo " "
45 set xfer:clobber on
46 get ${FILENAME}.rsc
47 rm ${FILENAME}.rsc
48 get ${FILENAME}.backup
49 rm ${FILENAME}.backup |
50 lftp -u $USER,$PASS $HOSTNAME
```