例子

1. #!/usr/bin/expect  -f // expect安装目录
3. set timeout 10
4. set username [lindex $argv 0]  //保存脚本参数到变量中
5. set password [lindex $argv 1]
6. set hostname [lindex $argv 2]
8. spawn ssh -l root 172.16.128.16
9. expect "password:"
10. send "passwd\r" <span style="white-space:pre">    </span>//用户的登陆密码，这里是root用户，密码是passwd
12. expect "#"  //引号里写成登录后的shell提示符
13. send "ls -lth /home/\r"
15. expect "#"  //引号里写成上一个命令执行后的shell提示符
16. send "exit\r"
18. interact  //定时器作用，与前面 set timeout 配合使用，用于超时退出

命令

Expect中最关键的四个命令是send,expect,spawn,interact。

send：用于向进程发送字符串

expect：从进程接收字符串

spawn：启动新的进程

interact：允许用户交互

**1. send命令**

send命令接收一个字符串参数，并将该参数发送到进程。

expect1.1> send "hello world\n"

hello world

**2. expect命令**

**(1)基础知识**

expect命令和send命令正好相反，expect通常是用来等待一个进程的反馈。expect可以接收一个字符串参数，也可以接收正则表达式参数。和上文的send命令结合，现在我们可以看一个最简单的交互式的例子：

expect "hi\n"

send "hello there!\n"

这两行代码的意思是：从**标准输入**中等到hi和换行键后，向**标准输出**输出hello there。

tips： $expect\_out(buffer)存储了所有对expect的输入，<$expect\_out(0,string)>存储了匹配到expect参数的输入。

比如如下程序：

expect "hi\n"

send "you typed <$expect\_out(buffer)>"

send "but I only expected <$expect\_out(0,string)>"

当在标准输入中输入

test

hi

是，运行结果如下

you typed: test

hi

I only expect: hi

**(2)模式-动作**

expect最常用的语法是来自tcl语言的模式-动作。这种语法极其灵活，下面我们就各种语法分别说明。

单一分支模式语法：

expect "hi" {send "You said hi"}

匹配到hi后，会输出"you said hi"

多分支模式语法：

expect "hi" { send "You said hi\n" } \

"hello" { send "Hello yourself\n" } \

"bye" { send "That was unexpected\n" }

匹配到hi,hello,bye任意一个字符串时，执行相应的输出。等同于如下写法：

expect {

"hi" { send "You said hi\n"}

"hello" { send "Hello yourself\n"}

"bye" { send "That was unexpected\n"}

}

**3. spawn命令**

上文的所有demo都是和标准输入输出进行交互，但是我们跟希望他可以和某一个进程进行交互。spawm命令就是用来启动新的进程的。spawn后的send和expect命令都是和spawn打开的进程进行交互的。结合上文的send和expect命令我们可以看一下更复杂的程序段了。

set timeout -1

spawn ftp ftp.test.com //打开新的进程，该进程用户连接远程ftp服务器

expect "Name" //进程返回Name时

send "user\r" //向进程输入anonymous\r

expect "Password:" //进程返回Password:时

send "123456\r" //向进程输入don@libes.com\r

expect "ftp> " //进程返回ftp>时

send "binary\r" //向进程输入binary\r

expect "ftp> " //进程返回ftp>时

send "get test.tar.gz\r" //向进程输入get test.tar.gz\r

这段代码的作用是登录到ftp服务器ftp ftp.uu.net上，并以二进制的方式下载服务器上的文件test.tar.gz。程序中有详细的注释。

**4.interact**

到现在为止，我们已经可以结合spawn、expect、send自动化的完成很多任务了。但是，如何让人在适当的时候干预这个过程了。比如下载完ftp文件时，仍然可以停留在ftp命令行状态，以便手动的执行后续命令。interact可以达到这些目的。下面的demo在自动登录ftp后，允许用户交互。

spawn ftp ftp.test.com

expect "Name"

send "user\r"

expect "Password:"

send "123456\r"

interact

**解决方法**

上文中提到：

如何从机器A上ssh到机器B上，然后执行机器B上的命令？如何使之自动化完成？

下面一段脚本实现了从机器A登录到机器B，然后执行机器B上的pwd命令，并停留在B机器上，等待用户交互。具体含义请参考上文。

#!/home/tools/bin/64/expect -f

set timeout -1

spawn ssh $BUser@$BHost

expect "\*password:" { send "$password\r" }

expect "$\*" { send "pwd\r" }

interact