# Expect 命令脚本语言简单指南

## 1.1 什么是Expect？

Expect是基于Tcl的相对简单的一个免费的脚本编程语言工具，用来实现自动和交互任务程序进行通信，无需人手工干预，比如SSH，FTP等这些程序正常情况下都需要手工与他们进行交互，而实用expect就可以摸你人手工交互的过程。实现自动的和远程的程序交互

Expect是一个用来实现自动交互功能的软件套件

虽然实用C，Perl等一样可以实现这样的交互功能，然而exec 图做的更专业，更出色慢，更简单，而且粗了支持Unix／Linux平台外还支持windows平台

## 1.2 Expect程序工作流程

Expect的工作流程可以理解为spawn启动进程＝＝》expect期待关键字＝＝》send向进程发送字符，退出结束

## 1.3 安装Expect软件

首先配置好yum安装源，比企鹅确认机器可以上网，然后执行yum install expect －y即可安装expect软件

[root@test ~]# rpm -qa expect

[root@test ~]# yum install expect -y

Loaded plugins: fastestmirror, security

Setting up Install Process

Loading mirror speeds from cached hostfile

\* base: mirrors.aliyun.com

\* epel: mirrors.aliyun.com

\* extras: mirrors.aliyun.com

\* updates: mirrors.aliyun.com

base | 3.7 kB 00:00

epel | 4.3 kB 00:00

extras | 3.4 kB 00:00

nginx | 2.9 kB 00:00

updates | 3.4 kB 00:00

Resolving Dependencies

--> Running transaction check

---> Package expect.x86\_64 0:5.44.1.15-5.el6\_4 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

Package Arch Version Repository Size

================================================================================

Installing:

expect x86\_64 5.44.1.15-5.el6\_4 base 256 k

Transaction Summary

================================================================================

Install 1 Package(s)

Total download size: 256 k

Installed size: 553 k

Downloading Packages:

expect-5.44.1.15-5.el6\_4.x86\_64.rpm | 256 kB 00:01

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : expect-5.44.1.15-5.el6\_4.x86\_64 1/1

Verifying : expect-5.44.1.15-5.el6\_4.x86\_64 1/1

Installed:

expect.x86\_64 0:5.44.1.15-5.el6\_4

Complete!

[root@test ~]# rpm -qa expect

expect-5.44.1.15-5.el6\_4.x86\_64

[root@test ~]#

## 1.4 先看Expect小实例

用ssh连接远程服务器

[root@test ~]# ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

oldboy@172.16.1.102's password:

eth1 Link encap:Ethernet HWaddr 00:0C:29:49:32:FE

inet addr:172.16.1.102 Bcast:172.16.1.255 Mask:255.255.255.0

inet6 addr: fe80::20c:29ff:fe49:32fe/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:2749 errors:0 dropped:0 overruns:0 frame:0

TX packets:3091 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:231811 (226.3 KiB) TX bytes:343922 (335.8 KiB)

[root@test ~]#

ecpect脚本文件

[root@test ~]# cat /service/scripts/expect\_test.sh

#!/usr/bin/expect

# This is for test expect

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

set timeout 60

expect "\*password:"

send "123456\n"

expect eof

exit

[root@test ~]#

如果是可执行权限，直接执行，结果如下：

[root@test ~]# /service/scripts/expect\_test.sh

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

oldboy@172.16.1.102's password:

eth1 Link encap:Ethernet HWaddr 00:0C:29:49:32:FE

inet addr:172.16.1.102 Bcast:172.16.1.255 Mask:255.255.255.0

inet6 addr: fe80::20c:29ff:fe49:32fe/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:2835 errors:0 dropped:0 overruns:0 frame:0

TX packets:3168 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:242758 (237.0 KiB) TX bytes:357179 (348.8 KiB)

[root@test ~]#

如果是普通文件，须要用expect调用文件

[root@test ~]# chmod -x /service/scripts/expect\_test.sh

[root@test ~]# expect /service/scripts/expect\_test.sh

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

oldboy@172.16.1.102's password:

eth1 Link encap:Ethernet HWaddr 00:0C:29:49:32:FE

inet addr:172.16.1.102 Bcast:172.16.1.255 Mask:255.255.255.0

inet6 addr: fe80::20c:29ff:fe49:32fe/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:2862 errors:0 dropped:0 overruns:0 frame:0

TX packets:3191 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:246309 (240.5 KiB) TX bytes:361434 (352.9 KiB)

[root@test ~]#

## 1.5 Expect语法

Expect中命令是最重要的的部分了，命令的实用方法如下

命令 ［选项］ 参数

### 1.5.1 spawn

spawn命令是Expect的初始命令，他用于启动一个进程，之所以expect操作都在这个进程中进行如果没有spawn语句，整个这个expect就无法执行，spawn实用方法如下：

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

在spawn命令后，直接加上要启动的进程，命令等信息，除此之外spawn还支持其他选项如：

-open 启动文件进程，具体说明省略

-ignore 忽略某些信号，具体说说明省略

### 1.5.2 expect

使用方法：

expect 表达式 动作 表达式 动作

expect命令用于等候一个相匹配内容的输出，一旦匹配上就只想expect后面的动作或命令，这个命令接收几个特有参数，用的最多的就是-re表示使用正则表达式的方式匹配，使用起来就像这样

spawn ssh -p 52113 oldboy@172.16.1.102

expect "psssword:" {send "123456\r"}

从上面的例子可以看出。Expect是依附于spawn命令的，当执行ssh命令后，expect就匹配关键字password，如匹配到官架子就会执行后边包含砸｛｝的exp\_send动作，匹配以及动作可以放在两行，这样就不需要使用｛｝了，就像下面这样，功能和上边是一样的

spawn ssh -p 52113 oldboy@172.16.1.102

expect "\*password:"

send "123456\n"

expect命令还有一种用法，它可以在一个expect匹配中多次匹配关键字，并给出处理动作，只需在关键字放在一个｛｝就可以了，当然要有exp\_continue

[root@test ~]# cat /service/scripts/expect\_test.sh

#!/usr/bin/expect

# This is for test expect

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

set timeout 60

expect "\*password:" {

-timeout 1

"yes/no" { exp\_send "yes\r"; exp\_continue }

"\*password:" { exp\_send "123456\r" }

timeout {puts "expect was timeout by odlboy."; return}

}

expect eof

exit

[root@test ~]#

### 1.5.3 exp\_send和send

在上面的介绍中，我们意境看到了exp\_send命令的使用，exp\_send命令是expect中的动作，它还在有一个完成同样工作的同胞：send，exp\_send命令剋有发送一些特殊符号，我们看到了\r（回车）还有一些其他的比如\n（换行）\t(制表符)

### 1.5.4 exp continue

这个命令一般用在动作中，它被使用的条件比较苛刻

[root@test ~]# cat /service/scripts/expect\_test.sh

#!/usr/bin/expect

# This is for test expect

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

set timeout 60

expect "\*password:" {

-timeout 1

"yes/no" { exp\_send "yes\r"; exp\_continue }

"\*password:" { exp\_send "123456\r" }

timeout {puts "expect was timeout by odlboy."; return}

}

expect eofs

exit

[root@test ~]#

在这个例子中国可以发现exp\_continue命令的使用方法，手下那他要出狱一个expect命令中没然后他属于一种动作命令，完成的工作就是从头开始遍历，也就是说如果没有这个命令，匹配第一个关键字之后酒会继续匹配第二个关键字，但是有了这个命令之后，匹配第一个关键字之后，第二次匹配仍从第一个关键字开始

### 1.5.5 send user

send\_user命令用来把后面的参数输出到标准输出中去，默认的send,exp\_send命令都是将参数输出到程序中去，用起来就像这样

[root@test scripts]# cat expect\_test.exp

#!/usr/bin/expect

# This is for test expect

if { $argc != 3 } {

send\_user "usage:expect scp-expect.exp file host dir \n"

sexits

}

# difind var

set file [lindex $argv 0]

set host [lindex $argv 1]s

set dir [lindex $argv 2]

set password "123456"

#spawn scp -P52113 /etc/hosts root@172.16.1.102:/etc/hosts

spawn scp -P 52113 $file root@172.16.1.102:$dir

expect "\*password:" {

-timeout 1

"yes/no" { exp\_send "yes\r"; exp\_continue }

"\*password:" { exp\_send "$password\r" }

timeout {puts "expect was timeout by odlboy."; return}

}

expect eofs

exit

[root@test scripts]#

### 1.5.6 exit

exit命令功能很简单，就是终结退出脚本，但是可以利用这个命令对脚本做一些扫尾工作，比如下面这样：

root@test ~]# cat /service/scripts/expect\_test.sh

#!/usr/bin/expect

# This is for test expect

spawn ssh -p 52113 oldboy@172.16.1.102 /sbin/ifconfig eth1

set timeout 60

expect "\*password:"

send "123456\n"

expect eof

exit –onexit {

exec rm $tmpfile

send\_user “Good bye\n”

}

## 1.6 Expect变量

[root@test scripts]# cat expect\_test.exp

#!/usr/bin/expect

# This is for test expect

if { $argc != 3 } {

send\_user "usage:expect scp-expect.exp file host dir \n"

sexits

}

# defind var

set file [lindex $argv 0]

set host [lindex $argv 1]s

set dir [lindex $argv 2]

set password "123456"

## 1.7 Expect关键字

### 1.7.1 Eof

eof(end-of-file)关键字用于匹配结束符，比如文件结束符，ftp传输停止等情况，在这个关键字后跟着动作来做进一步的控制，特别死ftp交互操作界面，她的作用很大

root@test ~]# cat /service/scripts/expect\_test.sh

#!/usr/bin/expect

# This is for test expect

spawn ftp oldboy@172.16.1.102

expect {

“password:” { exp\_send “who I’m I” }

eof {ftp connetct close }

}

interact {} |

### 1.7.2 Timeout

Timeout是expect中的一个重要变量，踏实一个全局性的时间控制开关，可以通过这个变量复制来规定整个expect操作时间，注意一个变量是服务与expect全局的，他不会纠缠于某一条命令即使命令没有任何错误，但时间仍然会激活这个变量，但这个时间到达以后除了激活一个文件之外不会做其他的事情，如果处理是脚本编写人员的额事

## 1.8 生产场景Expect实战

### 1.8.1 实战1:使用expect实现批量分发/etc/hosts文件

### 1.8.2 实战2:使用expect实现批量分发ssh秘钥文件

### 1.8.3 实战3:使用expect实现批量查看客户机的负载内存

## 1.9 有关expect说明