



Linux | DB | Open Source | Web

≡ Menu

- [Home](#)
- [Free eBook](#)
- [Start Here](#)
- [Contact](#)
- [About](#)

6 Expect Script Examples to Expect the Unexpected (With Hello World)

by Balakrishnan Mariyappan on October 12, 2010



Like 44

Tweet

Expect scripting language is used to feed input automatically to an interactive program. It is easy to learn compared to other scripting languages. Using expect script sysadmins and developers can automate redundant tasks easily. It works by expecting specific strings, and sending or responding strings accordingly.

Following three expect commands are used when automating any interactive processes.

- send – to send the strings to the process
- expect – wait for the specific string from the process
- spawn – to start the command

Make sure to install expect packages on your system, as it does not get installed by default. Once installed, you'll see the expect interpreter as "/usr/bin/expect". Generally, expect script files has .exp as extensions.

1. Expect "Hello World" Example

The following expect script is expecting the specific string "hello". When it finds it (after user enters it), "world" string will be send as response.

```
#!/usr/bin/expect
expect "hello"
send "world"
```

2. Timeout On Expect String

By default, the expect timeout is 10 seconds. If you don't enter anything for the expect command, it times out in 20 seconds. You can also change the timeout as shown below.

```
#!/usr/bin/expect
set timeout 10
```

```
expect "hello"  
send "world"
```

3. Automate User Processes With Expect

With the help of expect, you can automate the user processes and get the expected output. For example, you might use this to simplify the project testing process by writing scripts for the test cases.

The below example does the addition program automation.

```
#!/usr/bin/expect  
  
set timeout 20  
  
spawn "./addition.pl"  
  
expect "Enter the number1 : " { send "12\r" }  
expect "Enter the number2 : " { send "23\r" }  
  
interact
```

Execute this as shown below.

```
$ ./user_proc.exp  
spawn ./addition.pl  
Enter the number1 : 12  
Enter the number2 : 23  
Result : 35
```

In case, if you have written the code without interact command, then the script would exit immediately after sending the string "23\r". Interact command does the control, hands over the job to the addition process, and produces the expected result.

4. Match and No Match Contents in \$expect_out Variables

On the successful matching of string expect returns, but before that it stores the matched string in \$expect_out(0,string). The string that are received prior plus the matched string are stored in \$expect_out(buffer). The below example shows you the value of these two variable on match.

```
#!/usr/bin/expect  
  
set timeout 20  
  
spawn "./hello.pl"  
  
expect "hello"  
send "no match : <$expect_out(buffer)> \n"  
send "match : <$expect_out(0,string)>\n"  
  
interact
```

The hello.pl program just prints only two lines as shown below.

```
#!/usr/bin/perl  
  
print "Perl program\n";  
print "hello world\n";
```

Execute it as shown below.

```
$ ./match.exp
spawn ./hello.pl
Perl program
hello world
no match : <Perl program

hello>
match : <hello>
```

5. Automate SU login into Other User Accounts

Expect allows you to pass the password for the Linux login account from the program, instead of entering the password on the terminal. In the below program, su login is automated to login into desired accounts.

```
#!/usr/bin/expect

set timeout 20

set user [lindex $argv 0]

set password [lindex $argv 1]

spawn su $user

expect "Password:"

send "$password\r";

interact
```

Execute the above expect program as shown below.

```
bala@localhost $ ./su.exp guest guest
spawn su guest
Password:
guest@localhost $
```

After running the above script, it logged into the guest user account from bala user account.

6. SSH Login into Another Machine

The example expect program shown below [automates the ssh login](#) from one machine to another machine.

```
#!/usr/bin/expect

set timeout 20

set ip [lindex $argv 0]

set user [lindex $argv 1]

set password [lindex $argv 2]

spawn ssh "$user@$ip"

expect "Password:"

send "$password\r";

interact
```

Execute the above expect program as shown below.

```
guest@host1 $ ./ssh.exp 192.168.1.2 root password
spawn ssh root@192.168.1.2
Password:
Last login: Sat Oct  9 04:11:35 2010 from host1.geetkstuff.com
root@host2 #
```



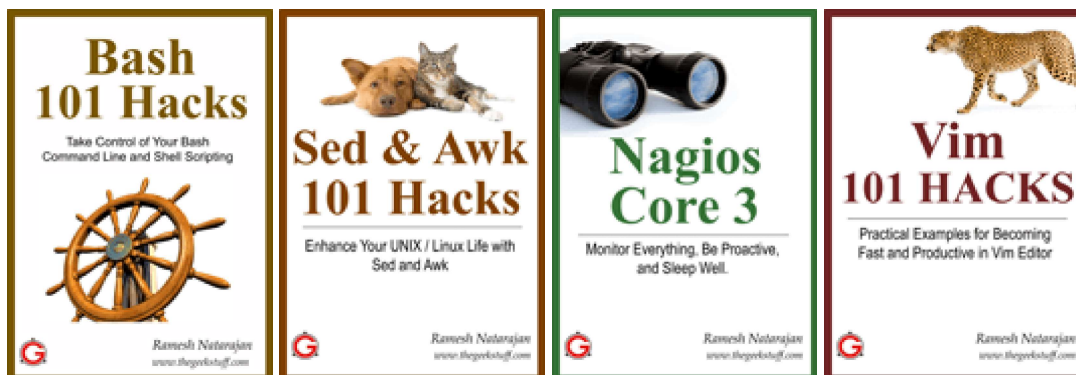
Tweet

Like 44

> [Add your comment](#)

If you enjoyed this article, you might also like..

1. [50 Linux Sysadmin Tutorials](#)
 2. [50 Most Frequently Used Linux Commands \(With Examples\)](#)
 3. [Top 25 Best Linux Performance Monitoring and Debugging Tools](#)
 4. [Mommy, I found it! – 15 Practical Linux Find Command Examples](#)
 5. [Linux 101 Hacks 2nd Edition eBook](#) **Free**
- [Awk Introduction – 7 Awk Print Examples](#)
 - [Advanced Sed Substitution Examples](#)
 - [8 Essential Vim Editor Navigation Fundamentals](#)
 - [25 Most Frequently Used Linux IPTables Rules Examples](#)
 - [Turbocharge PuTTY with 12 Powerful Add-Ons](#)



{ 37 comments... [add one](#) }

- Juan October 12, 2010, 4:59 am

You could also add one previous step to the SSH login for automated scripts: Test if remote key is already imported:

```
set timeout 5
expect "yes/no)?"
send "yes\r"
```

regards!

[Link](#)

- Greg Giese October 12, 2010, 7:02 am

Sure wish I could still print these. Some good stuff that I often don't have time to play with until later and then for get. When I printed them I could catch up on train or at home.

Thanks tho! Keep up the good work.

G

[Link](#)

- [Scott Rowley](#) October 12, 2010, 7:33 am

Something to keep in mind with these expect scripts using passwords is that anything typed on the command line goes into your history. So you could end up saving your password into your history file. Then others with root access can see these passwords. Hopefully if they have root access though they should be someone thats trusted. Still I don't want ANYONE knowing my passwords 😊

[Link](#)

- [Brian Kastor](#) October 14, 2010, 7:48 am

@Scott, good point.. but there is a way around the History file. I don't remember which option it is, but one of the HIST options will let you specify a space (literally hitting the space bar once) before the command and it won't put it in the History file. I did this at a company where they had an appliance and refused to have the three of us all use the root account. So I got tired of them watching what I was doing.. I need to go back and look that one up.

[Link](#)

- [Scott Rowley](#) October 14, 2010, 1:14 pm

@Brian, Interestingly enough I went to research this space method you mentioned and lo and behold, I was google routed right back to this website 😊

<http://www.thegeekstuff.com/2008/08/15-examples-to-master-linux-command-line-history/>
Item #10

[Link](#)

- [Pankaj](#) December 21, 2010, 7:47 am

@Brian/@Scott

The HIST options is: "unset HISTFILE"
We can set it as an alias in .bashrc file

```
alias xx='unset HISTFILE'
```

[Link](#)

- [james](#) December 21, 2010, 9:25 am

hello sir,

how can i install the expect packages? where i can get it?

[Link](#)

- [fwonce](#) January 18, 2011, 1:36 pm

The match.exp example didn't work out on my computer.

The output is:
spawn ./hello.pl
Perl program
hello world

I guess it's because the hello.pl is over too soon to give match.exp enough time to send...
After add a line of ";" to the hello.pl I can see the output exactly the same as yours.

[Link](#)

- [fwonce](#) January 18, 2011, 1:38 pm

the line added is <> sign.... it's omitted....

[Link](#)

- Prakash April 25, 2011, 3:59 am

How to send an exact string onto remote machine which contains metacharacters like \$ where it is not set in the local script?

send "RELEASE=ls -l|grep -i \$RELEASE_DATE"

Where in \$RELEASE_DATE is not set in the script

[Link](#)

- Thierry August 12, 2011, 2:39 am

For those who have trouble with the match.exp exemple like I did. Replace send "no match : \n" by puts stderr "no match : \n".

It works for me.

[Link](#)

- Thierry August 12, 2011, 2:44 am

Apparently this page doesn't support message including any strings that can be interpreted as HTML !! so my previous message is 100% unundestandable.

Let me try it that way:

replace : send "no match : \ \n"

by : puts stderr "no match : \ \n"

Hope this time it will work

[Link](#)

- [Thirumal Venkat](#) August 23, 2012, 12:43 am

in the 4th example. i think u should use send_user command instead of just send. the perl program ends after printing stuff and you are sending some input to the process even after ending. i'm having some problems on aix, then figured out that send_user outputs to the user and not the process which spawned it.

[Link](#)

- cliffy January 30, 2013, 12:52 am

Hi

I need to send a string with " part of the srting. Problem send "test "dos" " says its invalid. Could you help me

[Link](#)

- abhi May 7, 2013, 3:32 am

For script

```
#!/usr/bin/expect
expect "hello"
send "world"
```

I'm getting error

couldn't read file "hello": no such file or directory
expect.sh: line 3: send: command not found

[Link](#)

- Biju May 11, 2013, 11:00 pm

Please help me to handle Warning Message

Password:

Warning: your password will expire in 4 days

my script automatically comes out after warning message encounters

[Link](#)

- Ajit June 23, 2013, 11:40 pm

```
send "show running-config\r"
expect -ex " -More- " {send " "}
interact
```

I have this code which scrolls after expecting a " -More- " while running the show command. I want to put the expect code in a while loop so that it would loop whenever it finds " -More- "

I wrote it this way

```
while {1} {
expect -ex " -More- " {send " "}
if {expect "#"} break else continue
}
```

Another way I used a sample from another website:

```
set running 1
while {$running > 0} {
expect {
-exact " -More- " {send — " "}
"#"} {set running 0}
}
```

I am not able to make this work. Is there a way you could help me in this.

[Link](#)

- Rajani August 2, 2013, 12:39 am

Its simple and very good! 😊 I learnt expect scripting basics from these examples.
Thanks

[Link](#)

- S Ravikumar October 5, 2013, 10:25 am

Hi Sir,

I want to execute the "sas.servers status" in directory /u01/app/sas/configuration/Lev1 by using su sas.

```
#!/usr/bin/expect

set timeout 20

set user [lindex $argv 0]

set password [lindex $argv 1]

spawn su $user

expect "Password:"

send "$password\r";

send "cd /u01/app/sas/configuration/Lev1\r";

spawn "./sas.servers status"

interact
_____
Its not working could you please help.
```

Thanks,
Ravikumar Saikam.

[Link](#)

- Sunil October 7, 2013, 12:12 pm

Hi Ravi,

Try your command like this,

```
spawn su $user
expect "Password:"
send "$password\r"
expect "# " # Expected prompt
send "/u01/app/sas/configuration/Lev1/sas.servers status"
interact
```

-Sunil

[Link](#)

- Anonymous December 10, 2013, 8:38 am

hi,

I want to take the username and password in the variable and use it for doing scp/ssh.Can you help me on the same?

[Link](#)

- marty February 20, 2014, 9:35 am

for the ssh login....the script below do not timeout.....how to exit when password is wrong? and something like (wrong password)


```
#!/usr/bin/expect
set timeout 20
set ip [lindex $argv 0]
set user [lindex $argv 1]
set password [lindex $argv 2]
spawn ssh "$user@$ip"
expect "Password:"
send "$password\r";
interact
```

[Link](#)

- Ken Florentine March 31, 2014, 2:19 pm

if you don't want to worry about:

```
expect "yes/no)?"
```

Just use:

```
spawn ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $user@$host
```

and you will not see that message anymore.

[Link](#)

- Shilpa October 6, 2014, 3:36 am

Hi,

I am trying to replace an ip in a file. That 'sed' command i am sending but throwing error.

```
#!/usr/bin/expect
spawn ssh -p 20222 root@10.43.38.242
expect "\*root@10.43.38.242's password:"
send "root\r"
expect "#"
send "sed -i 's/. *host match.*\10\43\38\180\g' config_bk.xml\r"
expect eof
```

error- extra characters after close-quote

while executing

```
"send "sed -i 's/. *host match.*\10\43\38\180\g' config_bk.xml\r"
```

Could anyone please help me on this?

[Link](#)

- Anonymous March 2, 2015, 5:09 am

Thanks. This helps me a lot!

[Link](#)

- Ardian April 21, 2015, 11:29 pm

Hi,

For "5. Automate SU login into Other User Accounts"

We could see that the user password typed and can be seen. What if I want to type the unseen password for privacy, like "read -s password" command.

Thank you.

[Link](#)

- sathiya May 22, 2015, 12:52 am

why aix is not allowing us to use metacharacters as password?

[Link](#)

- Larry Dighera August 25, 2015, 10:18 am

For those interested in obtaining the ‘expect’ package or information related thereto, [here](#).

In the event that html may not be accepted within comments, you’ll find the expect home page at: expect.sourceforge.net

[Link](#)

- sandeep nagendra October 28, 2015, 1:30 am

Hi,

How can we expand aliases inside expect script ?

Ex: I have defined below alias in my .bashrc file.
alias some_host="ssh root@x.x.x.x"

How can i use “some_host” inside expect script ?

[Link](#)

- Thiyagu March 3, 2016, 2:55 am

It is good stuff for beginner!

[Link](#)

- [Kalum](#) April 28, 2016, 8:12 am

I am using your expect script to login the server and passing shutdown command. I have a separate file with list of 100 IPs and need your help to figure it out. How can I make the script to get each IP and login and pass the command to the servers.

```
#!/usr/bin/expect
set timeout 20
set user [lindex $argv 0]
set host [lindex $argv 1]
set password [lindex $argv 2]
spawn su $user
expect "Password:"
send "$password\r";
expect "# " # Expected prompt
send "sudo shutdown -h now"
expect "$password\r";
interact
```

[Link](#)

- jens May 13, 2016, 3:33 am

This saved my day! Thanks

[Link](#)

- vijay June 16, 2016, 10:23 pm

simple explanation for the beginner

[Link](#)

- Prasath January 26, 2017, 3:13 pm

Hi,

I need to run a script to execute a script on remote server. there are list of servers. I tried many ways but all are throwing error.

Can you please help me resolve this issue.

Thanks

[Link](#)

- gines February 22, 2017, 5:47 am

Hi, Done a script and I run, it runs with another user gives me spawn / bin / bash error, and if I open a session with the same user but from another pc I get the same error

[Link](#)

- daniel March 3, 2017, 8:00 am

I got an error at the 2nd exemple:

couldn't execute "./addition.pl": no such file or directory
while executing

[Link](#)

- [Sachin Gupta](#) April 8, 2017, 10:26 pm

We are having issue while developing the program who can pass enter to start and enter to stop (in loop). In developed logic, we are able to start the script and read first enter key(\n) after that it requires to sleep for 1 min and then pass enter to stop the process. But currently it is only passing first enter to start internal process in the script.

```
=====
#!/usr/bin/expect

spawn python abc.py
expect {
    -re "Press ENTER to start*" { send "\n" }
}
#sleep 1 minute
expect sleep 60
expect {
    -re "Press ENTER to stop*" { send "\n" }
}
```

interact

=====

Expect guru, Please help here

[Link](#)

Leave a Comment

Name

Email

Website

Comment

☐ Notify me of followup comments via e-mail

Next post: [Linux Beginners Guide to NFS Mount Using Exportfs](#)

Previous post: [6 Examples to Backup Linux Using dd Command \(Including Disk to Disk\)](#)

[RSS](#) | [Email](#) | [Twitter](#) | [Facebook](#) | [Google+](#)

EBOOKS

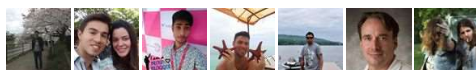
- **Free** [Linux 101 Hacks 2nd Edition eBook](#) - Practical Examples to Build a Strong Foundation in Linux
- [Bash 101 Hacks eBook](#) - Take Control of Your Bash Command Line and Shell Scripting
- [Sed and Awk 101 Hacks eBook](#) - Enhance Your UNIX / Linux Life with Sed and Awk
- [Vim 101 Hacks eBook](#) - Practical Examples for Becoming Fast and Productive in Vim Editor
- [Nagios Core 3 eBook](#) - Monitor Everything, Be Proactive, and Sleep Well



The Geek Stuff

17,254 likes

2 friends like this



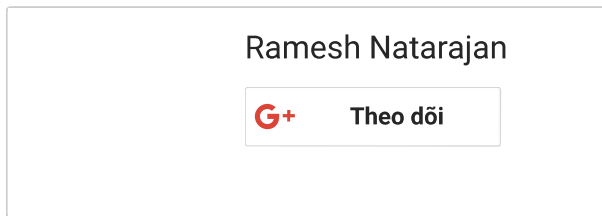
POPULAR POSTS

- [15 Essential Accessories for Your Nikon or Canon DSLR Camera](#)
- [12 Amazing and Essential Linux Books To Enrich Your Brain and Library](#)

- [50 UNIX / Linux Sysadmin Tutorials](#)
- [50 Most Frequently Used UNIX / Linux Commands \(With Examples\)](#)
- [How To Be Productive and Get Things Done Using GTD](#)
- [30 Things To Do When you are Bored and have a Computer](#)
- [Linux Directory Structure \(File System Structure\) Explained with Examples](#)
- [Linux Crontab: 15 Awesome Cron Job Examples](#)
- [Get a Grip on the Grep! – 15 Practical Grep Command Examples](#)
- [Unix LS Command: 15 Practical Examples](#)
- [15 Examples To Master Linux Command Line History](#)
- [Top 10 Open Source Bug Tracking System](#)
- [Vi and Vim Macro Tutorial: How To Record and Play](#)
- [Mommy, I found it! -- 15 Practical Linux Find Command Examples](#)
- [15 Awesome Gmail Tips and Tricks](#)
- [15 Awesome Google Search Tips and Tricks](#)
- [RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams](#)
- [Can You Top This? 15 Practical Linux Top Command Examples](#)
- [Top 5 Best System Monitoring Tools](#)
- [Top 5 Best Linux OS Distributions](#)
- [How To Monitor Remote Linux Host using Nagios 3.0](#)
- [Awk Introduction Tutorial – 7 Awk Print Examples](#)
- [How to Backup Linux? 15 rsync Command Examples](#)
- [The Ultimate Wget Download Guide With 15 Awesome Examples](#)
- [Top 5 Best Linux Text Editors](#)
- [Packet Analyzer: 15 TCPDUMP Command Examples](#)
- [The Ultimate Bash Array Tutorial with 15 Examples](#)
- [3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id](#)
- [Unix Sed Tutorial: Advanced Sed Substitution Examples](#)
- [UNIX / Linux: 10 Netstat Command Examples](#)
- [The Ultimate Guide for Creating Strong Passwords](#)
- [6 Steps to Secure Your Home Wireless Network](#)
- [Turbocharge PuTTY with 12 Powerful Add-Ons](#)

CATEGORIES

- [Linux Tutorials](#)
- [Vim Editor](#)
- [Sed Scripting](#)
- [Awk Scripting](#)
- [Bash Shell Scripting](#)
- [Nagios Monitoring](#)
- [OpenSSH](#)
- [IPTables Firewall](#)
- [Apache Web Server](#)
- [MySQL Database](#)
- [Perl Programming](#)
- [Google Tutorials](#)
- [Ubuntu Tutorials](#)
- [PostgreSQL DB](#)
- [Hello World Examples](#)
- [C Programming](#)
- [C++ Programming](#)
- [DELL Server Tutorials](#)
- [Oracle Database](#)
- [VMware Tutorials](#)



About The Geek Stuff



My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about [Ramesh Natarajan](#) and the blog.

Contact Us

Email Me : Use this [Contact Form](#) to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

[Follow us on Google+](#)

[Follow us on Twitter](#)

[Become a fan on Facebook](#)

Support Us

Support this blog by purchasing one of my ebooks.

[Bash 101 Hacks eBook](#)

[Sed and Awk 101 Hacks eBook](#)

[Vim 101 Hacks eBook](#)

[Nagios Core 3 eBook](#)

Copyright © 2008–2017 Ramesh Natarajan. All rights reserved | [Terms of Service](#)