



BASH Programming

Bash Script Include Another Bash Script

5 months ago • by Prateek Jangid

If you are a Linux enthusiast, then you have probably heard of the bash scripting concept. In bash scripting, you can automate the code block of the command sequence. However, sometimes, the code block in bash scripting becomes complex, and it can be tough for a user to execute it further.

That's why bash offers a feature that includes a script in another. It lets you run the commands efficiently and achieve an enhanced level of automation. In this blog, we will explain the concept and provide examples of how to include a bash script into another script.

The source Command

The source is one of the most common commands to include a bash script into another. For example, if you want to add the "info.sh" script into "main.sh", so all you have to do is add the following line into the "main.sh" script:

```
source info.sh
```

A terminal window with a dark background. The prompt is "#!/bin/bash". The user has entered the command "source info.sh". The output of the command is "echo 'You Need to AUTHORIZE'", where "echo" is in blue and the string is in orange.

```
#!/bin/bash
source info.sh
echo "You Need to AUTHORIZE"
```


The info.sh script contains the following information:

Once you run the main.sh script, it will execute the info.sh first and then run the other commands:

Dot (.) or Dot-Slash

You can use dot (.) instead of source command as both are similar and used to execute the command from the current shell.

You can also use the dot-slash (./) instead of dot (.) to execute the command from the current directory:



```
#!/bin/bash
./info.sh

echo "You Need to AUTHORIZE"
```

How to Include Multiple Scripts Into One Script

Let's create a script to check the current CPU load and execute the other script according to the load. Now, we need to create two scripts, one for the high load and the other is for the medium load:

For High Load:

```
#!/bin/bash
echo "High CPU load detected."
echo "Killing non-essential processes..."
pkill -9 -f firefox
echo "Non-essential processes have been terminated."
```

For Medium Load:

```
#!/bin/bash
echo "Moderate CPU load."
echo "Logging current process statuses..."
ps -aux > /tmp/current_processes.txt
echo "Process log created at /tmp/current_processes.txt."
```

The script above creates a tmp text file if the system has a medium CPU load. Let's now create a third script, which will have the above two scripts link:

```
#!/bin/bash
cpu_load=$(uptime | awk -F '[a-z]:' '{ print $2}' | cut -d, -f2 | xargs)
```