

How to execute Shell script in Linux

Executing shell scripts in Linux is one of the essential skills that every Linux user or administrator needs to master.

Knowing how to run scripts is incredibly useful when automating tasks, setting up cron jobs, or just trying to manage system activity. However, there is a certain joy in writing scripts that not only work, but are also elegant and neat. Let's learn how to execute shell scripts in Linux through the following article!

What is a shell script?

Before we dive into execution, let's quickly review what a shell script is. A shell script is a text file that contains a series of commands for a Unix-based shell, such as Bash, to execute. It's essentially a program written in the shell's command-line language.

How to execute Shell script in Linux

Step 1: Create a simple shell script

The first step to executing a script is, of course, writing it. For this example, let's create a simple script that reads "Hello, World!" and the current date.

Open a terminal on your Ubuntu system and use your favorite text editor to create a new file. For example, nano is for quick edits because it's simple, but you might prefer vim if you need something more powerful.

```
nano hello_world.sh
```

Inside the editor, enter the following script:

```
#!/bin/bash echo "Hello, World!" echo "Today's date is: 30/09/2024"
```

This script does two things: It prints "Hello, World!" and then displays the current date. The `#!/bin/bash` at the top is called the shebang and it tells the system that this script should be run in the Bash shell.

Step 2: Make the script executable

By default, the script you just created is not executable. You need to make it executable using the `chmod` command. This is one of those steps that is easy to forget but very important.

```
chmod +x hello_world.sh
```

The **+x** option makes the script executable by the user, group, and others, depending on your permissions settings.

Step 3: Execute the shell script

Now, you are ready to run the script. There are several ways to execute a script in Linux and this article will cover the most common methods:

Method 1: Execute using relative path

This is the simplest method if you are in the same directory as the script:

```
./hello_world.sh
```

`./` tells the shell to look for the script in the current directory. If all goes well, you should see the following output:

```
Hello, World! Today's date is: Tue Aug 8 14:22:31 UTC 2024
```

Method 2: Execute with absolute path

Sometimes you will need to specify the full path to the script, especially if the script is located in another directory.

```
/home/username/scripts/hello_world.sh
```

This method is very convenient when running scripts from cron jobs or in other scripts.

Method 3: Execute using a specific shell

Although the shebang specifies which shell to use, you can override this by running the script in a different shell. For example, if you want to run the script in `sh` instead of `bash`, you can execute:

```
sh hello_world.sh
```

Or, explicitly in `bash`:

```
bash hello_world.sh
```

This can be useful when testing how your script works in different shells.

Troubleshooting common problems

Even simple scripts can have problems. Here are some common problems and how to fix them:

1. **Permission denied** : If you see this error, double check that you have executed the script using **chmod +x** command .
2. **Command not found** : This usually means that the script's shebang line is missing or incorrect, or the script is not in your `PATH`.
3. **Script not running as expected** : Use `bash -x` followed by the script name to run the script in debug mode. This will display each command as it is executed, which is helpful for figuring out where the error

is.

Executing shell scripts in Linux is a basic task that opens up a world of automation and efficiency. For tasks like server management or just trying to automate a few things on your local machine, shell scripting is a skill worth mastering. It's satisfying to see a well-written script run flawlessly, saving time and reducing the chance of human error.

You finished reading the article "**How to execute Shell script in Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.