Learn How to Set Your $PATH Variables Permanently in Linux

In Linux (also UNIX) **$PATH** is environment variable, used to tell the shell where to look for executable files. **$PATH** variable provides great flexibility and security to the Linux systems and it is definitely safe to say that it is one of the most important environment variables.

Programs/scripts that are located within the $PATH’s directory, can be executed directly in your shell, without specifying the full path to them. In this tutorial you are going to learn how to set $PATH variable globally and locally.

First, let’s see your current $PATH’s value. Open a terminal and issue the following command:

$ echo $PATH

The result should be something like this:

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games

The result shows a list of directories separated by colons. You can easily add more directories by editing your user’s shell profile file.

In different shells this can be:

1.Bash shell -> ~/.bash\_profile, ~/.bashrc or profile

2.Korn Shell -> ~/.kshrc or .profile

3.Z shell -> ~/.zshrc  or .zprofile

Please note that depending on how you are logging to the system in question, different file might be read. Here is what the bash manual says, keep in mind that the files are similar for other shells:

**/bin/bash**

The bash executable

**/etc/profile**

The systemwide initialization file, executed for login shells

**~/.bash\_profile**

The personal initialization file, executed for login shells

**~/.bashrc**

The individual per-interactive-shell startup file

**~/.bash\_logout**

The individual login shell cleanup file, executed when a login shell exits

**~/.inputrc**

Individual readline initialization file|

Considering the above, you can add more directories to the $PATH variable by adding the following line to the corresponding file that you will be using:

$ export PATH=$PATH:/path/to/newdir

Of course in the above example, you should change “/path/to/newdir” with the exact path that you wish to set. Once you have modified your .\*rc or .\*\_profile file you will need to call it again using the “source” command.

For example in bash you can do this:

$ source ~/.bashrc

Below, you can see an example of mine $PATH environment on a local computer:

marin@[TecMint]:[/home/marin] $ echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:**/home/marin/bin**

This is actually a good practice to create a local “bin” folder for users where they can place their executable files. Each user will have its separate folder to store his contents. This is also a good measure to keep your system secured.

If you have any questions or difficulties setting your $PATH environment variable, please do not hesitate to submit your questions in the comment section below.