

# Instructions for installing Unix / Linux

An important Unix concept is the environment, which is defined by environment variables. Some are set up by the system, others are set up by you, by the shell or any program you download.

An important Unix concept is the **environment**, which is defined by environment variables. Some are set up by the system, others are set up by you, by the shell or any program you download.

A variable is a character string that we assign a value to. The specified value can be a number, text, file name, device or any other data type.

For example, we first set up a variable TEST and then we access its value using the **echo** command.

```
$ TEST = "Unix Programming" $ echo $ TEST Unix Programming
```

Note that environment variables are set without using the \$ symbol, but while accessing them, we use the \$ symbol as a prefix. These variables keep their values until we exit the shell.

When you log into the system, Shell is under initialization mode to set up the environment. This is usually a 2-step process that includes Shell to read the following files:

**/ etc / profile**

**profile**

The process is as follows:

1. Check the shell to see if / etc / profile exists.
2. If it exists, Shell reads it. Otherwise, this file is ignored. No error message displayed.
3. The shell checks whether the .profile file exists or not in your home directory. The main directory is the directory that starts when you successfully login to the system.
4. If it exists, Shell reads it; otherwise, Shell skips. No error message will be displayed.

As soon as both files are read, the Shell displays a prompt:

```
$
```

This is the prompt where you can enter commands to execute them.

**Note** - The **Shell** initialization process listed in detail here applies to all **Bourne**- type Shells, but some added files are used by **bash** and **ksh** .

## **.profile File in Unix / Linux:**

The file / **etc** / **profile** is maintained by the system administrator and contains Shell initialization information that is required by all users on the system.

**.Profile** file is under your control. You can enter more Shell information as you wish to this file. The least information you need to set up includes:

The terminal type you are using

List of directories in which the commands are located

List of variables of your terminal type.

You can check **.profile** in the main directory. Open it using the **micro** editor and check all the variables set for your environment.

## Set terminal type in Unix / Linux

Usually the type of terminal you are using is automatically configured by the **login** or **getty** program. Sometimes, this automatic shaping process predicts your terminal type is incorrect.

If the terminal type is not set correctly, the result of the command may look strange, or you may not interact with the Shell correctly.

To ensure this does not happen, most users set their terminal type to the lowest common denominator form as follows:

```
$ TERM = vt100 $
```

## Set up PATH in Unix / Linux:

When you type any command on the command prompt, Shell must locate the command before it can be executed.

The PATH variable determines the location in which the Shell will search for commands. Usually it is set as follows:

```
$ PATH = / bin: / usr / bin $
```

Here each entry is set individually by a colon ":" as directories. If you ask the Shell to execute a command and it cannot search for that command in any directory provided with the PATH variable, an information similar to the following appears:

```
$ hello hello : not found $
```

The variables PS1 and PS2 are discussed in the next section.

## PS1 and PS2 variables in Unix / Linux

The characters that Shell displays when the command prompt is kept in the PS1 variable. You can change this variable to whatever you want. As soon as you change it, it will be used by the shell from that point on.

For example, if you tell the command:

```
$ PS1 = '=>' => => =>
```

The prompt may become =>. To set the value of PS1 so that it indicates the working directory, issue the following command:

```
=> PS1 = "[u @ hw] $" [ root @ ip - 72 - 167 - 112 - 17 / var / www / tutorial
```

The result of this command is a command prompt that displays the user's username, hostname, and working directory.

There are several escape sequences that are used as parameter values for PS1, so try to limit yourself so that the prompt does not overwhelm the information.

Sequence of escape Description **t** Current time, expressed in the form of HH: MM: SS **d** Current date, expressed in the form of Day of the Month Day of Day **n** New line **s** Current Shell environment **W** Working directory **w** Full path enough of working directory **u** Current property name of user **h** Hostname of the current device **#** Number of commands of the current command. Increase every time a new order is entered **\$** If the effective UID is 0 (that is, if you are logged in as root), end the prompt with the # character; otherwise, use \$.

You can make changes by yourself every time you log in, or you can have changes made automatically in PS1 by adding it to the **.profile** file .

When you notify an incomplete command, Shell will display a second command prompt and wait for you to complete the command and re-enter it.

The second default command prompt is greater than>, but can be changed by redefining the PS2 variable.

Here is an example of how to use the second command prompt:

```
$ echo "this is a> test" this is a test $
```

Next is an example that redefines PS2 with a customized prompt:

```
$ PS2 = "secondary prompt->" $ echo "this is a secondary prompt-> test" this is
```

## Environment variables in Unix / Linux

Here is a local list of important environment variables. These variables will be set and accessed as mentioned above.

Variables **DISPLAY** Description Contains an identifier for the display that X11 programs should use by default. **HOME** Indicates the current directory of the current user: The default parameter for the cd command is available. **IFS** Indicates the Internal Field Separator that is used by word analysis separating after expansion. **LANG** LANG extends the default system Locale; LC\_ALL can be used to override (with higher priority) this. For example, if its value is pt\_BR, then the language set is (Brazilian) Portuguese and the locale is Brazil. **LD\_LIBRARY\_PATH** On many Unix systems with a Dynamic Linker, contain a separate colon list of

Dynamic Linker directories that search for shared objects when building a process image after execution, before Search any other directory. **PATH** Indicates the path search for commands. It is a separate colon list of directories in which the Shell searches for commands. **PWD** Indicates the current working directory when set by the cd command. **RANDOM** Creates a random integer between 0 and 32767 each time it is mentioned. **SHLVL incremental** because every time bash is started. This variable is useful when determining whether or not the available exit command ends in the current entry. **TERM** Related display types. **TZ Relevant** time zone. It can receive values ??like GMT, AST . **UID** Expand the ID using the digital form of the current user, initializing when Shell starts.

Here is an example to illustrate several environment variables:

```
$ echo $ HOME / root ] $ echo $ DISPLAY $ echo $ TERM xterm $ echo $ PATH / us
```

### According to Tutorialspoint

Previous article: Permissions / File access mode in Unix

Next article: Basic utilities: print, email in Unix

You finished reading the article "**Instructions for installing Unix / 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.