Engr. Rudolph Nfor, [4/15/2022 6:56 AM]:

how to get environmental variables in linux.

copy and paste the below link in your browser to explore environmental variables and how to use or set them.

How To Read and Set Environmental and Shell Variables on Linux:

Introduction:

When interacting with your server through a shell session, there are many pieces of information that your shell compiles to determine its behavior and access to resources. Some of these settings are contained within configuration settings and others are determined by user input.

One way that the shell keeps track of all of these settings and details is through an area it maintains called the environment. The environment is an area that the shell builds every time that it starts a session that contains variables that define system properties.

In this guide, we will discuss how to interact with the environment and read or set environmental and shell variables interactively and through configuration files.

To follow along with this tutorial using a terminal in your browser, click the Launch an Interactive Terminal! button below:

Launch an Interactive Terminal!

Otherwise if you’d like to follow along using your local system or a remote server, open a terminal and run the commands from this tutorial there.

How the Environment and Environmental Variables Work

Every time a shell session spawns, a process takes place to gather and compile information that should be available to the shell process and its child processes. It obtains the data for these settings from a variety of different files and settings on the system.

The environment provides a medium through which the shell process can get or set settings and, in turn, pass these on to its child processes.

The environment is implemented as strings that represent key-value pairs. If multiple values are passed, they are typically separated by colon (:) characters. Each pair will generally look something like this:

KEY=value1:value2:...

If the value contains significant white-space, quotations are used:

KEY="value with spaces"

The keys in these scenarios are variables. They can be one of two types, environmental variables or shell variables.

Environmental variables are variables that are defined for the current shell and are inherited by any child shells or processes. Environmental variables are used to pass information into processes that are spawned from the shell.

Shell variables are variables that are contained exclusively within the shell in which they were set or defined. They are often used to keep track of ephemeral data, like the current working directory.

By convention, these types of variables are usually defined using all capital letters. This helps users distinguish environmental variables within other contexts.

Common Environmental and Shell Variables

Some environmental and shell variables are very useful and are referenced fairly often. Here are some common environmental variables that you will come across:

SHELL: This describes the shell that will be interpreting any commands you type in. In most cases, this will be bash by default, but other values can be set if you prefer other options.

TERM: This specifies the type of terminal to emulate when running the shell. Different hardware terminals can be emulated for different operating requirements. You usually won’t need to worry about this though.

USER: The current logged in user.

PWD: The current working directory.

OLDPWD: The previous working directory. This is kept by the shell in order to switch back to your previous directory by running cd -.

LS\_COLORS: This defines color codes that are used to optionally add colored output to the ls command. This is used to distinguish different file types and provide more info to the user at a glance.

MAIL: The path to the current user’s mailbox.

PATH: A list of directories that the system will check when looking for commands. When a user types in a command, the system will check directories in this order for the executable.

LANG: The current language and localization settings, including character encoding.

HOME: The current user’s home directory.

\_: The most recent previously executed command.

In addition to these environmental variables, some shell variables that you’ll often see are:

BASHOPTS: The list of options that were used when bash was executed. This can be useful for finding out if the shell environment will operate in the way you want it to.

BASH\_VERSION: The version of bash being executed, in human-readable form.

BASH\_VERSINFO: The version of bash, in machine-readable output.

COLUMNS: The number of columns wide that are being used to draw output on the screen.

DIRSTACK: The stack of directories that are available with the pushd and popd commands.

HISTFILESIZE: Number of lines of command history stored to a file.

HISTSIZE: Number of lines of command history allowed in memory.

HOSTNAME: The hostname of the computer at this time.

IFS: The internal field separator to separate input on the command line. By default, this is a space.

PS1: The primary command prompt definition. This is used to define what your prompt looks like when you start a shell session. The PS2 is used to declare secondary prompts for when a command spans multiple lines.

SHELLOPTS: Shell options that can be set with the set option.

UID: The UID of the current user.