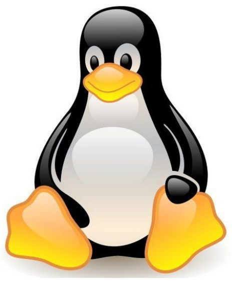
Linux Plus for AWS and DevOps



CLARUSWAY
WAY TO REINVENT YOURSELF

Linux Environment Variables

CLARUSWAY
WAY TO REINVENT YOURSELF

Table of Contents

CLARUSWAY
WAY TO REINVENT YOURSELF

1 What are Environment variables?

CLARUSWAY
WAY TO REINVENT YOURSELF

What are Environment variables?

- **Dynamic-named value** which is **defined in the OS** for **processes and applications** to use
- Can be **created, edited, saved, and deleted** and give information about the system behavior
- Allow you to **configure and customize the behavior** of applications and processes
- Variable names are **case-sensitive**. By convention, environment variables should be in **UPPER CASE**.

Lifecycle of an Environment Variable

- Many environment variables are **global and defined for every user**
  - created up log in
  - available in every session
- Users can create **temporary environment variables**
  - **Session-based and inheritable** by child-processes
- **Shell variables** can also be created by users
  - **Not inherited** by child processes
  - Valid only in the **current shell**

Common Environment Variables

Variable	Description
PATH	A **colon (:)-separated list of directorie**s in which your system looks for executable files.
USER	The username
HOME	**Default path** to the **user's home** directory
EDITOR	Path to the program which edits the content of files
UID	User's unique ID
TERM	Default terminal emulator
SHELL	Shell being used by the user
LANG	The current locales settings.

CLARUSWAY
WAY TO REINVENT YOURSELF

Summary of Commands

Command	Description
Display Variables	
echo	Display the value of the variable `> echo $VARIABLE`

printenv	Display the value of a specific or list all environment variables
	`> printenv $VARIABLE`
	`> printenv`

env	List all environment variables `> env`
set	List all environment variables, shell variables and shell functions `> set`
Manipulate Variables	
=	Create or update a shell variable `> MYVAR=value`
export	Create or update an environment variable `> export MYVAR=value`
unset	Delete an environment variable `> unset MYVAR`

CLARUSWAY
WAY TO REINVENT YOURSELF

2

Querying Variables

CLARUSWAY
WAY TO REINVENT YOURSELF

echo

- `echo $VARIABLE_NAME` **displays the variable value**
  - e.g. echo $PATH
- requires the “$” symbol
  - echo is expecting a literal string
  - must **use $ to indicate this is a variable**

```
altaz@DESKTOP-D4LE3NN:/etc$ echo $USER
altaz
altaz@DESKTOP-D4LE3NN:/etc$ echo $HOME
/home/altaz
altaz@DESKTOP-D4LE3NN:/etc$ echo $TERM
xterm-256color
altaz@DESKTOP-D4LE3NN:/etc$
```

CLARUSWAY
WAY TO REINVENT YOURSELF

`printenv` and `env`

- `printenv` **lists *all* of the current environment variables**
- `printenv VARIABLE_NAME` displays a single variable
  - alternative to echo
  - **do not use the $**, since printenv is expecting a variable name
- `env` **also lists all** of the current environment variables
  - global, persistent
  - temporary
  - does *not* list shell variables

`set`

- `set` **lists *all* of the current *environment AND shell* variables AND shell *functions***
- **do not** use set to assign variables
- `set <flags> [options]` is also used to configure **shell behavior**
  - out of scope for this discussion

3

Manipulating Variables

CLARUSWAY
WAY TO REINVENT YOURSELF

Shell Variables

- to create or update an environment variable, assign it using “=”
  - `MYVAR=my new variable`
  - variable name is **uppercase** by convention
  - there are **no spaces** around the “=”

- scope
  - this is a temporary **shell variable**, does not extend to new sessions
  - the variable **not inherited by child processes** (e.g. scripts)

CLARUSWAY
WAY TO REINVENT YOURSELF

`export` - Update or Create an Environment Variable

- to define or update an environment variable, use `export`
  - e.g. `export MYVAR=my new variable`
  - variable name is **uppercase** by convention
  - there are **no spaces** around the "="
- scope
  - this is a shell variable, does not extend to new sessions
  - the variable **IS inherited by child processes**

CLARUSWAY
WAY TO REINVENT YOURSELF

Creating Global Variables

- environment variables are defined when the **OS starts**
- for Linux scripts run at startup include:
  - /etc/profile, ~/.bash_profile,~/bash_login, ~/.bashrc
  - (note: for non-interactive sessions ~/.bashrc is run)
- **global variables can be set in one of these files**
  - typically **~./bash_profile**
  - use EXPORT VARNAME=varvalue

CLARUSWAY
WAY TO REINVENT YOURSELF

`unset` - Deleting a Session Variable

- to delete an environment variable, use `unset`
  - e.g. `unset MYVAR`
  - do not use the "$"
- note that `set` and `unset` **are not inverses** of one another
  - do not use `set` to create an environment variable

CLARUSWAY
WAY TO REINVENT YOURSELF

Recap: Summary of Commands

Command	Description
Display Variables	
echo	Display the value of the variable `> echo $VARIABLE`

printenv	Display the value of a specific or list all environment variables
	`> printenv $VARIABLE`
	`> printenv`

env	List all environment variables `> env`
set	List all environment variables, shell variables and shell functions `> set`
Manipulate Variables	
=	Create or update a shell variable `> MYVAR=value`
export	Create or update an environment variable `> export MYVAR=value`
unset	Delete an environment variable `> unset MYVAR`

CLARUSWAY
WAY TO REINVENT YOURSELF

4

QUOTING

Quoting

Quoting is used to disable special treatment of certain characters and words, as well as to prevent parameter expansion and preserve what is quoted.

The bash shell knows rare, important characters. For example, $var is used to extend the value of the element.

```
echo "$PATH"
echo "$PS1"
```

Quoting

Double Quotes

- The double quote ("quote") protects everything enclosed between two double quote marks except $, ', " and \.

echo "$SHELL"
echo "/etc/*.conf"

Single Quotes

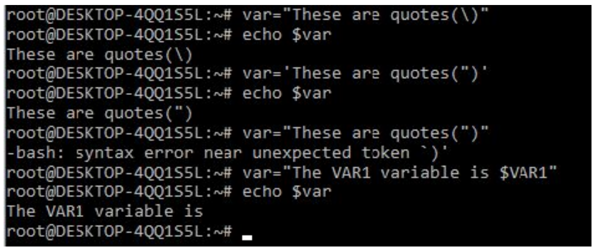
- The single quote ('quote') protects everything enclosed between two single quote marks.

echo '$SHELL’
echo '/etc/*.conf'

Backslash

- Use the backslash to change the special meaning of the characters or to escape special characters within the text such as quotation marks.

echo "Path is \$PATH"



```
root@DESKTOP-4QQ1S5L:~# var="These are quotes(\)"
root@DESKTOP-4QQ1S5L:~# echo $var
These are quotes(\)
root@DESKTOP-4QQ1S5L:~# var='These are quotes(")'
root@DESKTOP-4QQ1S5L:~# echo $var
These are quotes(")
root@DESKTOP-4QQ1S5L:~# var="These are quotes(")"
-bash: syntax error near unexpected token `)'
root@DESKTOP-4QQ1S5L:~# var="The VAR1 variable is $VAR1"
root@DESKTOP-4QQ1S5L:~# echo $var
The VAR1 variable is
root@DESKTOP-4QQ1S5L:~# _
```

CLARUSWAY
WAY TO REINVENT YOURSELF

5

PATH
An Important Environment Variable

CLARUSWAY
WAY TO REINVENT YOURSELF

What is PATH

- PATH tells the OS **where to look to find executables** (programs)

- for example, consider a program /bin/grep
  - program file is “grep” located in the directory “/bin”
  - to run the program without a PATH variable
    - **user must type /bin/grep**
  - after creating a PATH variable that includes /bin
    - **user only has to type “grep”**

- any time **new software is installed**, PATH should be updated

CLARUSWAY
WAY TO REINVENT YOURSELF

How the OS uses PATH

- directories listed in PATH are **searched sequentially**

- search will be **stopped as soon as a match is found**

- i.e. if a file exists in multiple directories, it is executed in the first directory found in the PATH variable

CLARUSWAY
WAY TO REINVENT YOURSELF

6

sudo Command

CLARUSWAY
WAY TO REINVENT YOURSELF

sudo Command

Commands	Meaning
sudo -l	List available commands.
sudo command	Run command as root.
sudo -u root command	Run command as root.
sudo -u user command	Run command as user.
sudo su	Switch to the superuser account.
sudo su -	Switch to the superuser account with root's environment.
sudo su - username	Switch to the username's account with the username's environment.
sudo -s	Start a shell as root
sudo -u root -s	Same as above.
sudo -u user -s	Start a shell as user.

CLARUSWAY
WAY TO REINVENT YOURSELF

sudo

- sudo (“superuser do”) command **elevates a users privilege to root** (admin)

- individual commands
    - place **sudo before** any command
    - that **command runs with elevated privileges**

- assuming root identity
    - alternatively, can **assume root identity**
    - use **sudo su** or **sudo su -**

- unsure if you’re running as root?
    - check the trailing character the prompt
    - if it's a pound sign (#), you're logged in as root.

- Typically requires a password in corporate environments

CLARUSWAY
WAY TO REINVENT YOURSELF

Exercise

Create a variable named **MYVAR** with the value of **“my value”**
Print value of the **MYVAR** variable to the screen
Assign **“new value”** to the **MYVAR** variable
Print value of the **MYVAR** variable to the screen
Delete **MYVAR** variable
Print value of the **MYVAR** variable to the screen

CLARUSWAY
WAY TO REINVENT YOURSELF

CLARUSWAY
WAY TO REINVENT YOURSELF

CLARUSWAY
WAY TO REINVENT YOURSELF

Break
return @ 8pm

Kahoot!

CLARUSWAY
WAY TO REINVENT YOURSELF

THANKS!

Any questions?

CLARUSWAY
WAY TO REINVENT YOURSELF