Chapter-6: Customization and Filters (8L)

Customization

Use of environment variables, Some common environment variables (HOME, PATH, LOGNAME, USER, TERM, PWD, PS1, PS2),

Aliases, Brief idea of command history

Environment variables / System Variables:

When a Unix program starts up, the environment accessible to it includes a set of name to value associations (names and values are both strings). Some of these are set manually by the user; others are set by the system at login time, or by your shell or terminal emulator (if you're running one).

Under Unix, environment variables tend to carry information about file search paths, system defaults, the current user ID and process number, and other key bits of information about the runtime environment of programs.

These are set by the system at the boot sequence or after logging in time. **The set statement displays of all environment variables.**
For example

1. HOME(your home directory)
2. LOGNAME(your username)
3. MAIL and MAILCHECK
4. PATH
5. PS1 and PS2(The prompt strings)
6. SHELL
7. TERM

At a shell prompt, typing **set** followed by a newline will list all currently defined shell variables.

```
debasis@LAPTOP-H3N6JCNE:~$ set | more
```

COMMAND NAME: env

env - run a program in a modified environment

```
debasis@LAPTOP-H3N6JCNE:~$ env | more
```

System Environment Variables:

There are a number of well-known environment variables you can expect to find defined on startup of a program from the Unix shell.

USER:

Login name of the account under which this session is logged in (Berkeley Software Distribution (**BSD**) convention).

```
debasis@LAPTOP-H3N6JCNE:~$ echo $USER

debasis
```

LOGNAME:

Login name of the account under which this session is logged in .

```
debasis@LAPTOP-H3N6JCNE:~$ echo $LOGNAME

debases
```

HOME:

Home directory of the user running this session.

```
debasis@LAPTOP-H3N6JCNE:~$ echo $HOME

/home/debasis
```

COLUMNS:

The **number of character-cell columns** on the controlling terminal or terminal-emulator window.

```
debasis@LAPTOP-H3N6JCNE:~$ echo $COLUMNS

112
```

LINES:

The **number of character-cell rows on the controlling terminal** or terminal-emulator window.

```
debasis@LAPTOP-H3N6JCNE:~$ echo $LINES

27
```

SHELL:

The name of the user's command shell (often used by shellout commands).

```
debasis@LAPTOP-H3N6JCNE:~$ echo $SHELL

/bin/bash
```

BASH:

NAME

bash - GNU Bourne-Again SHell

SYNOPSIS

bash [options] [command_string | file]

COPYRIGHT

Bash is Copyright (C) 1989-2018 by the Free Software Foundation, Inc.

DESCRIPTION

Bash is an sh-compatible command language interpreter that executes commands read from the standard input or from a file. Bash also incorporates useful features from the Korn and C shells (ksh and csh).

PATH:

The list of directories that the shell searches when looking for executable commands to match a name.

```
debasis@LAPTOP-H3N6JCNE:~$ echo $PATH
```

To add directory in a path:

```
debasis@LAPTOP-H3N6JCNE:~$
PATH=$PATH:/home/debasis/STUDENT
```

TERM:

Name of the terminal type of the session console or terminal emulator window. TERM is special in that programs to create remote sessions over the network (such as *telnet* and *ssh*) are expected to pass it through and set it in the remote session.

NAME

term - format of compiled term file.

SYNOPSIS

term

```
debasis@LAPTOP-H3N6JCNE:~$ echo $TERM

xterm-256color
```

PWD:

pwd - print name of current/working directory

```
debasis@LAPTOP-H3N6JCNE:~$ echo $PWD

/home/debasis
```

How to Customize BASH Prompt in LINUX:

In Linux, much of your work occurs from a command prompt, also known as the **shell**, or **BASH** (Bourne-Again Shell). The shell interprets your commands and passes them to the operating system for execution.

The default BASH prompt is the one you see when you first open a terminal or command line. It usually looks something like this:

```
username@hostname:~$
```

The tilde sign `~` indicates that the current working directory is the current user's home directory.

```
debasis@LAPTOP-H3N6JCNE:~$ cd ~

debasis@LAPTOP-H3N6JCNE:~$ pwd

/home/debasis

debasis@LAPTOP-H3N6JCNE:~$
```

The dollar sign `$` means the current user is a standard user.
A root user would be identified with a hash sign `#`.

Customize Bash Prompt in LINUX:

Like most Linux applications, BASH reads a configuration file to determine its behavior. This file is in the home directory:

```
~/.bashrc
```

```
debasis@LAPTOP-H3N6JCNE:~$ vi ~/.bashrc
open the bashrc file and change the prompt.
PS1="Student>"
```

Create a temporary change to BASH Prompt:

You can change the BASH prompt temporarily by using the **`export`** command. This command changes the prompt until the user logs out.

Set the BASH prompt to only display the username by entering the following:

```
debasis@LAPTOP-H3N6JCNE:~$ PS1='STUDENT>'

STUDENT>
```

```
debasis@LAPTOP-H3N6JCNE:~$ export PS2='@@@'

debasis@LAPTOP-H3N6JCNE:~$ echo $PS2

@@@
```

COMMAND NAME- alias:

alias

An alias is a **short cut** command to a longer command. Users may type the alias name to run the longer command with less typing. Without arguments, **alias** prints a list of defined aliases. A new alias is defined by assigning a string with the command to a name. Alias are often set in the `~/.bashrc` file.

SYNOPSIS

alias [-p] [name[=string]]

OPTIONS:

-p

Display all aliases in a format suitable for input to the shell.

```
debasis@LAPTOP-H3N6JCNE:~$ alias -p

alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal ||
echo error)" "$(history|tail -n1|sed -e '\"s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\")"'

alias egrep='egrep --color=auto'

alias fgrep='fgrep --color=auto'

alias grep='grep --color=auto'

alias l='ls -CF'
alias la='ls -A'

alias ll='ls -alF'

alias ls='ls --color=auto'
```

```
debasis@LAPTOP-H3N6JCNE:~$ ls -l | wc -l

75

debasis@LAPTOP-H3N6JCNE:~$ alias count='ls -l | wc -l'

debasis@LAPTOP-H3N6JCNE:~$ count

75
```

COMMAND - history:

NAME:

history - GNU History Library

history command is used to view the previously executed command.
This feature was not available in the Bourne shell.
Bash and Korn support this feature in which every command executed is treated as the event and is associated with an event number using which they can be recalled and changed if required.
These commands are saved in a history file. In Bash shell **history** command shows the whole list of the command.

Syntax:

history

```
debasis@LAPTOP-H3N6JCNE:~$ history 5
 1996  chmod ugo+rwx CalEx2.sh
 1997  ls -l
 1998  pwd
 1999  tty
 2000  history 5
```