

Bash Shell Scripting Introduction

Sankar S

The Shell

The shell is a program that runs automatically when the users logs into the system. This shell interprets commands typed in by the user on the terminal. There are a variety of shells in use – the C shell, the Bourne Shell, the Korn Shell.

The Bourne shell was written by Stephen Bourne at Bell Laboratories, where UNIX was originally developed. Because it is found on most UNIX systems, many software developers work under the assumption that the Bourne shell is available on a UNIX system. This use does not mean that it is the best shell, but simply that it is the most common. Other shells, most notably the Korn shell, were written to enhance the Bourne shell, so shell programs written for Bourne run under the Korn shell. In some literature, the Bourne shell is called the UNIX system Version 7 shell.

Functions of a shell

The shell provides users with a functional and flexible environment.

- It allows the user to assign values to variables.
- It controls input and output redirection on the command line.
- The output of one command can be redirected as the input of another command using the pipe operator (|).
- The shell allows the user to customize his environment.
- The built-in programming language is interpreted. Each line is interpreted and executed. Commands may span more than a line in length. The shell waits for a NEWLINE character before it starts interpreting the command.

Know these configurations

```
[ssankar@localhost ~]$ cd
[ssankar@localhost ~]$ pwd
/home/ssankar
[ssankar@localhost ~]$ ls -lrt .bashrc
-rw-r--r--. 1 ssankar ssankar 124 Jun 22 2010 .bashrc
[ssankar@localhost ~]
[ssankar@localhost ~]$ cat .bashrc
# .bashrc
```

```
# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```

```
# User specific aliases and functions[ssankar@localhost ~]$ ls -lrt .vimrc
ls: cannot access .vimrc: No such file or directory
[ssankar@localhost ~]$
```

.vimrc file : [VIM_RC.txt](#)

```
[ssankar@localhost ~]$ vi .bashrc
[ssankar@localhost ~]$ source .bashrc
[ssankar@localhost ~]$ . .bashrc
[ssankar@localhost ~]$ bash
[ssankar@localhost ~]$
```

.bash_rc : [bash_rc.txt](#)

Know these configurations

```
[ssankar@localhost ~]$ tail -1 /etc/passwd
ssankar:x:500:500:Sankar Suriyanarayanan:/home/ssankar:/bin/bash
[ssankar@localhost ~]$ which $SHELL
/bin/bash is /bin/bash
[ssankar@localhost ~]$
```

**** change default shell to C Shell for user ssankar ******

```
[ssankar@localhost ~]$ vi /etc/passwd
[ssankar@localhost ~]$ su -
Password:
[root@localhost ~]# vi /etc/passwd
[root@localhost ~]# su ssankar
[ssankar@localhost /root]$ which $SHELL
/bin/csh
[ssankar@localhost /root]$
```

***** PATH for your programs to be followed during executiong ****

```
[ssankar@localhost ~]$ echo $PATH
/tools/vendor/smartbear/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
[ssankar@localhost ~]$
```

Basic Commands

- **ls** - List the contents of a directory
- **mkdir** - Create a new directory
- **cd** - Change the current directory
- **cp** - Copy a file or directory
- **mv** - Move a file or directory
- **rm** - Remove (delete) a file or directory
- **cat** - Show contents of file(s)
- **du** - Estimate your current disk usage
- **Chown** - Change the ownership of files
- **chmod** - Changer the access levels
- **Df** - report file system disk space usage
- **grep** - Pattern matching
- **sed** - Search and Replace
- **cut** - Print specific columns
- **sort** - Sort alphabetically / numerically
- **uniq** - Remove duplicate lines from a file

#	Permission	rwx
7	read, write and execute	rwx
6	read and write	rw-
5	read and execute	r-x
4	read only	r--
3	write and execute	-wx
2	write only	-w-
1	execute only	--x
0	none	---

Examples

- Grep example:

```
prompt> cat example.txt
Hello World
Goodbye World
prompt> cat example.txt | grep Hello
Hello World
prompt> cat example.txt | grep -v Hello
Goodbye World
```

- Sed example:

```
prompt> cat example.txt
Hello World
Goodbye World
prompt> cat example.txt | sed "s/Hello/Goodbye/g"
Goodbye World
Goodbye World
```

- Cut example:

```
prompt> cat example.txt
1,Hello,World
2,Goodbye,World
prompt> cat example.txt | cut -d "," -f 2-
Hello,World
Goodbye,World
```

Examples

- Sort example:

```
prompt> cat example.txt
```

```
2 Goodbye
```

```
1 Hello
```

```
2 Goodbye
```

```
prompt> cat example.txt | sort -n
```

```
1 Hello
```

```
2 Goodbye
```

```
2 Goodbye
```

- Uniq example:

```
prompt> cat example.txt
```

```
2 Goodbye
```

```
1 Hello
```

```
2 Goodbye
```

```
prompt> cat example.txt | sort -n | uniq
```

```
1 Hello
```

```
2 Goodbye
```