Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Chapter 5

Unix Shell

School of Computer Science,
MIT-World Peace University,
Pune, Maharashtra, India.

Amol D. Vibhute (PhD)
Assistant Professor

Email ID:- amol.vibhute@mitwpu.edu.in

Roadmap of Chapter

- Introduction
- Shell Prompt
- Shell Types
- Shell Scripts
- Comments
- Extended Shell Scripts

Introduction:

- How can we interact with OS:
  - GUI (Graphical User Interface)
  - CLI (Command Line Interface)

- **Unix Shell**:
  - Interpreter-execute command line by line,
  - Compiler-execute the whole program at once.

- **Shell**:
  - A term used for the interactive user interface of an operating system.
  - It is the layer of programming that understands and executes the commands.

Cont.…

- A Shell provides an interface to the Unix OS system.
- It gathers input from user and executes programs based on that input. When a program finishes executing, it displays that program's output.
- Shell is an environment in which we can run our commands, programs, and shell scripts.
- There are different flavors of a shell, just as there are different flavors of operating systems.
- Each flavor of shell has its own set of recognized commands and functions.

Cont.…

- **Uses of shells**:
  - Customizing our work environment,
  - Automating our daily tasks,
  - Automotive repeating tasks,
  - Executing important system procedure like shutting down the system.
  - Performing same operating on many files.
  - When a single Unix command unable to satisfy client requirement will make together some commands run at a time.

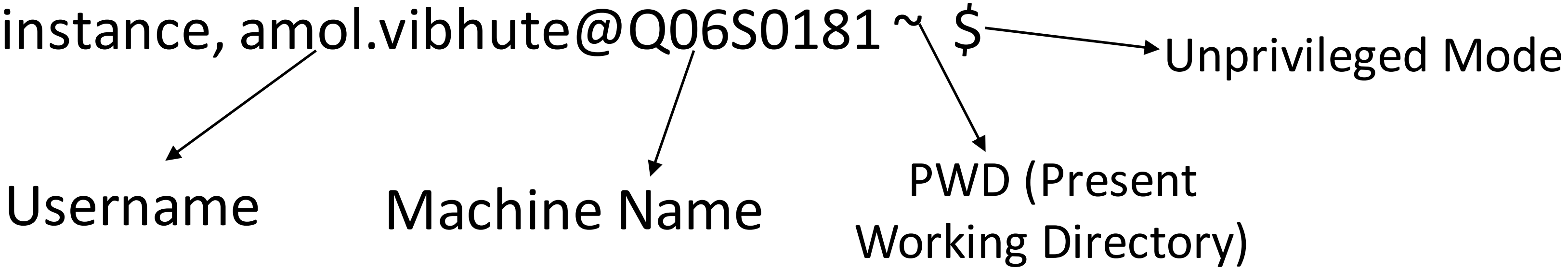
Shell Prompt:

- The prompt, **$**, which is called the **command prompt**, is issued by the shell. While the prompt is displayed, we can type a command.
- Shell reads input after press Enter.
- It determines the command we want executed by looking at the first word of our input.
- A word is an unbroken set of characters.
- Spaces and tabs separate words.
- Following is a simple example of the date command, which displays the current date and time −
  - $date
  - Mon Mar 22 07:31:59 IST 2021

Cont.…

- When a shell is used interactively, it displays a string where it is waiting for a command from the user, this is known as shell prompt.
- When a normal or regular user starts a shell, the default prompt ends with a **$** character.
- When a super (root) user starts a shell, the default prompt ends with a # character.

- For instance, amol.vibhute@Q06S0181 ~ $



Unprivileged Mode

Username

Machine Name

PWD (Present Working Directory)

Shell Types:

- In Unix, there are two major types of shells −
  - **Bourne shell (**predefined**)−** If you are using a Bourne-type shell, the $ character is the default prompt.
  - **C shell** − If you are using a C-type shell, the % character is the default prompt.
- The Bourne Shell has the following subcategories −
  - Bourne shell (sh)-standard and secure shell operated by root user.
  - Korn shell (ksh)-Created by David Corn which supports multiple type of programming.
  - Bourne Again shell (bash)-can run any of the programming lang or used for any programming lang. Default shell comes with **$** sign.
  - POSIX shell (sh)
- The different C-type shells follow −
  - C shell (csh)-Designed by Bill Joy at university of California and used for c programming.
  - TENEX/TOPS C shell (tcsh)

- The original Unix shell was written in the mid-1970s by Stephen R. Bourne while he was at the AT&T Bell Labs in New Jersey.
- Bourne shell was the first shell to appear on Unix systems, thus it is referred to as "the shell".
- Bourne shell is usually installed as **/bin/sh** on most versions of Unix. For this reason, it is the shell of choice for writing scripts that can be used on different versions of Unix.

Shell Scripts:

- Collection of Unix commands is known as **Unix shell scripting**.
- Shell scripting is writing a series of command for the shell to execute.
- It can combine lengthy and repetitive sequences of commands into a single and simple script, which can be stored and executed anytime. This reduces the effort required by the end user.
- The basic concept of a shell script is a list of commands, which are listed in the order of execution. A good shell script will have comments, preceded by # sign, describing the steps.
- There are conditional tests, such as value A is greater than value B, loops allowing us to go through massive amounts of data, files to read and store data, and variables to read and store data, and the script may include functions.
- We are going to write many scripts in the next sections. It would be a simple text file in which we would put all our commands and several other required constructs that tell the shell environment what to do and when to do it.
- Shell scripts and functions are both interpreted. This means they are not compiled.

Cont….

- **Difference between Scripting and Programming**:

Scripting	Programming
Weakly or loosely typed programs	strictly typed, and syntax based
Interpreter based	compiler based
Less libraries are available while installing	More libraries are available
No header files required	Header files are compulsory.
Special environment is not required	Special environment and setup is required.

Cont.…

- **Example Script:**
  - Assume we create a **test.sh** script. Note all the scripts would have the **.sh** extension. Before you add anything else to your script, you need to alert the system that a shell script is being started. This is done using the **shebang** construct.
  - For example −#!/bin/sh
  - This tells the system that the commands that follow are to be executed by the Bourne shell. It's called a shebang because the # symbol is called a hash, and the **!** symbol is called a bang.

  - To create a script containing these commands, you put the shebang line first and then add the commands −

```
#!/bin/bash
pwd
ls
```

Shell Comments:

- You can put your comments in your script as follows −
  - #!/bin/bash
  - # User : MIT
  - # Demo Example
  - # Script follows here:
  - pwd
  - ls
- Save the above content and make the script executable −$chmod +x test.sh
- The shell script is now ready to be executed −
- $./test.sh
- Upon execution, you will receive the following result −
- /home/amol.vibhute
  - index.htm unix-basic_utilities.htm unix-directories.htm
  - test.sh unix-communication.htm unix-environment.htm

Extended Shell Scripts:

- Shell scripts have several required constructs that tell the shell environment what to do and when to do it. Most scripts are more complex.
- The shell is, after all, a real programming language, complete with variables, control structures, and so forth. No matter how complicated a script gets, it is still just a list of commands executed sequentially.
- The following script uses the read command which takes the input from the keyboard and assigns it as the value of the variable PERSON and finally prints it on STDOUT.
- #!/bin/sh
  - # User : MIT
  - # Demo Example
  - # Script follows here:
  - echo "What is your name?"
  - read PERSON
  - echo "Hello, $PERSON“
- Here is a sample run of the script −
  - $./test.sh
  - What is your name?
  - Amol
  - Hello, Amol
  - $

References:

- Unix Shell Programming: Yashwant Kanitkar, BPB Publications, New Delhi.

Thank You !!!