UNIX Bash Shell Scripting

Week 2

Lecturer: Shahdad Shariatmadari

May 2020

Agenda

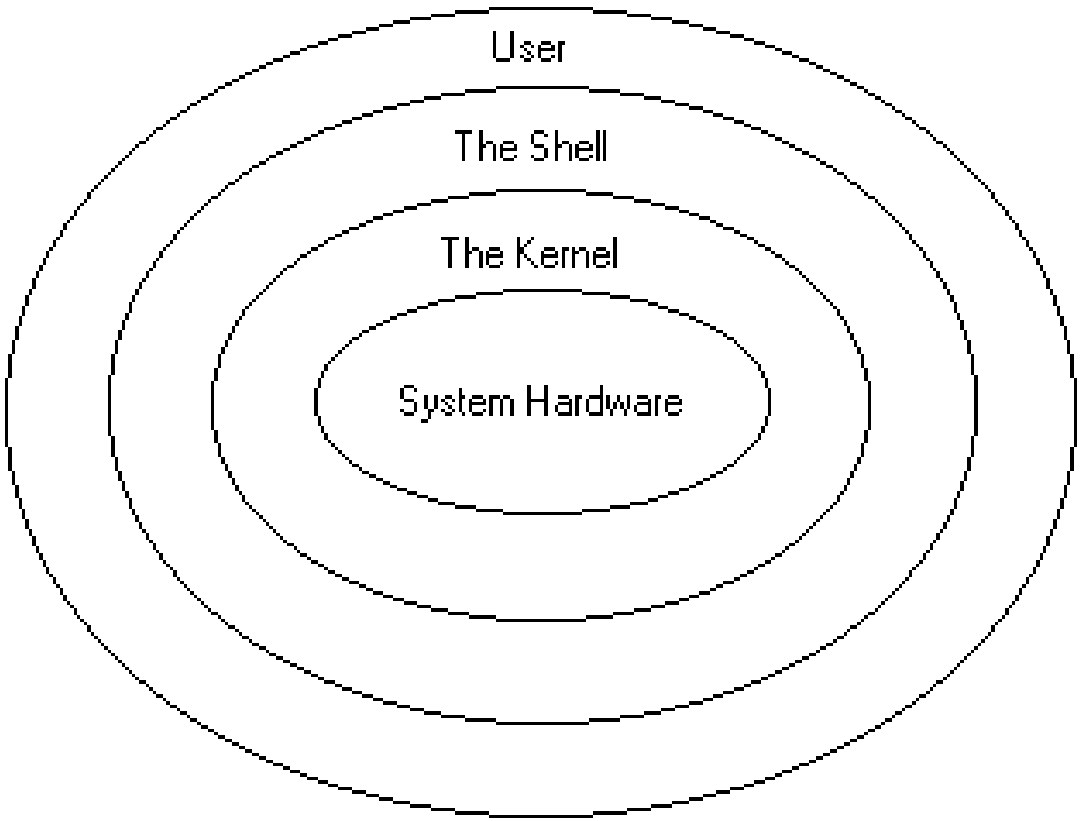
- Shell basics
- Command execution in detail
- Recalling and editing previous commands
  Quoting

Some open questions:

- What is Shell and What is Kernel?
- What does Shell do in Linux?
- Name some of the famous Unix Shell

Shell and Kernel

- **kernel** is the core of the operating system that controls all the tasks of the system
- **shell** is the interface that allows the users to communicate with the **kernel**



What is Shell

- The **shell** is the interface between the command prompt user and the operating system.
  - It interprets the user's commands and invokes the appropriate systems calls so that the commands are executed.
- Type of shells:
  - The first shell was the **Bourne shell (sh)** by Steven Bourne of AT&T Bell Labs.
  - The **C shell (csh)** was developed for BSD Unix, to offer better programming, but much of the syntax was changed. It has a few ancestor shells, which add additional features (tcsh).
  - The **Korn shell (ksh)**, by David Korn (AT&T Bell Labs) offered better programming like C shell, but followed the Bourne shell syntax. It is often called the K shell.
  - The **BASH shell (bash)** is quite similar to the Korn shell, but was released by the GNU project with an open source license.

Shell Script

- A **shell script** is a computer **program** designed to be run by the Unix **shell**, a command-line interpreter.
  - The various dialects of **shell** scripts are considered to be **scripting** languages. (.bash, .sh, .ksh, …)
- Typical operations performed by **shell** scripts include file manipulation, **program** execution, and printing text.

Shell Scripting

- **Shell scripting** is scripting in ***any*** **shell**
  - Bash scripting is scripting specifically for Bash.
- In practice, however, "shell script" and "bash script" are often used interchangeably, unless the shell in question is not Bash.

Shell Variables

- Variables can be read/write or read-only
- Name of a variable can be any sequence of letters and numbers, but it must not start with a number
- Shell variables are classified in 2 groups
  - System (shell) variables, describing the working environment
  - User-created variables, associated with scripts

Common (Environment) Shell Variables

- Shell environment variables shape the working environment whenever you are logged in
- Common shell variables include:
  - PS1 – primary prompt
  - PWD – present working directory
  - HOME – absolute path to user's home
  - PATH – list of directories where executables are
  - HOST – name of the host
  - USER – name of the user logged in
  - SHELL – current shell
- * use **env** to see the list of all environment variables!

The PATH variable

- PATH is an environment variable present in Unix/Linux operating systems, listing directories where executable programs are located
- Multiple entries are separated by a colon (:)
- Each user can customize a default PATH
- The shell searches these directories whenever a command is invoked in the sequence listed
- In case of multiple matches use the which utility to determine which match has a precedence
- On some systems the present working directory may not be included in the PATH by default
- Use ./ prefix or modify the PATH as needed

Assigning a Value

- Syntax: **`name=value`**
- For example: **`course=ULI101`**
- If variable values are to contain spaces or tabs they should be surrounded by quotes
- For example: phone="1 800 123-4567"

Read-Only Variables

- Syntax: `readonly variable = value`
- For example: *readonly phone="123-4567"*
- After a variable is set, it can be protected from changing by using the *readonly* command
- If no variable name is supplied a list of defined read only variables will be displayed

Removing Variables

- For example:
- **`course=`**
- OR
- **`unset course`**
- Read-only variables cannot be removed – you must log out for them to be cleared

Variable Substitution

- Whenever you wish to use the value of a variable (its contents), use the variable name preceded by a dollar sign ($)
- This is called variable substitution
  - Example:

```
name=Bob
echo $name
```

echo command

- Displays messages to the terminal followed by a newline
  - Use the **–n** option to suppress the default newline
- Output can be redirected or piped Arguments can be quoted to preserve spaces, double quotes to allow variable substitution or single quotes to disable variable substitution

echo command

- **echo** is a command that outputs the strings it is being passed as arguments
  - `echo hello world`
  - `echo 'hello world'`
  - `echo "hello world"`
- You may use echo to list the files!
  - `echo *.txt`
    - Show all files in current folder with .txt extension
  - `echo .*`
    - Show all hidden files in current folder

read commend

- The read command allows obtaining user input and storing it into a variable
  - Everything is captured until the Enter key is pressed
- Example:
- echo –n "What is your name? "
- read name
- echo Hello $name

Creating Shell Scripts

Here is an example of a simple shell script:

Contents of Shell Script

```
cat askAge.bash
# Start of Shell Script
# Prompt user for age and store result in a variable

echo -n "Please enter your age (in years): "
read age

# Print empty line, then print text using value
# of age stored in that variable...
echo
echo "You are $age years old"

# End of Shell Script
```

Execution of Shell Script

```
./askAge.bash
Please enter your age (in years): 44

You are 44 years old
```

Arithmetic expression

- A *Bash* and *Korn* shell built-in command for math is **let**.
  - let z=5
  - let z=$z+1
- With the *BASH* shell, whole arithmetic expressions may be placed inside double parenthesis.
  - ((e=5))
  - ((e = e + 3))

Activity

- Develop a new script : Name : **Script1.sh**
- The script will calculate the area of a rectangle, width is 100 and height is 250
- When you execute the script, it displays the following:
  - `The area of rectangle with the size (100 by 250) is 25000.`
- Modify the script in order to ask user to enter “width” and “height”