Shell Scripting

Miss. Apurva Patil.
Miss. Vaishnavi Katgaonkar

SHELL SCRIPTING

WLUG
COMMUNITY | KNOWLEDGE | SHARE

What is Shell ?

- It is a program or an environment provided for interaction between user and system.
- It is a command-line interpreter.

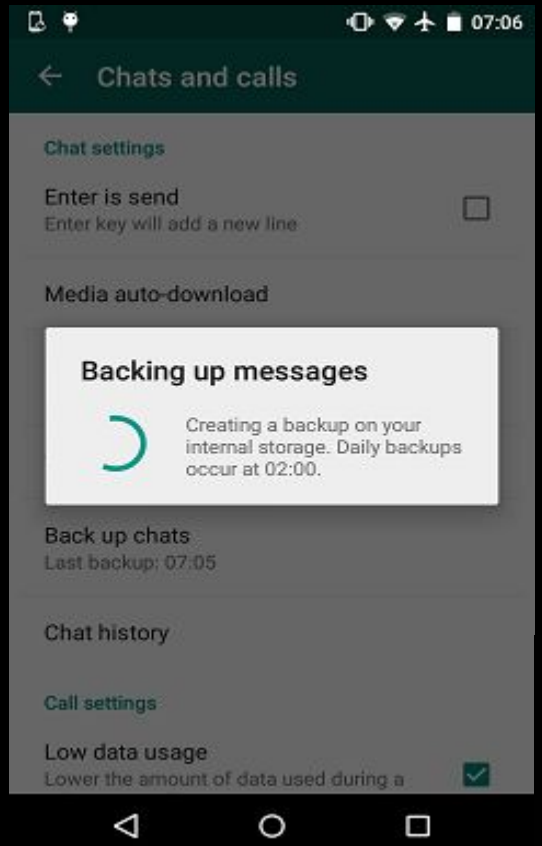
Why Shell Scripting ?

- Automation of tasks - Saves lot of time

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Why Shell Scripting ?

- Executing routine backups



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Why Shell Scripting ?

- Manipulating files

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Why Shell Scripting ?

- System administration

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Types of **Linux Shells**

Types of Shell in Linux:

- sh - Bourne Shell
- csh - C Shell
- ksh - Korn Shell
- tcsh - Tenex Shell
- zsh - Z Shell
- bash - Bourne Again Shell

Commands in Linux:

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Commands	Description
pwd	Prints current working directory
ls	Lists all the files/ directory of a current directory
cd	Change directory
mkdir	Creates new directory
touch	Creates file if not exists and updates access/ modification date
rm	Removes files or directories

Commands	Description
cat	Concatenate and print the contents of file
apropos	It is useful for searching commands without knowing their exact names
man	Opens manual of given command
sudo	Allows a permitted user to execute command as the superuser

Setting File permissions in Linux :

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Ownership in Linux files

→ User :

- Unique account of a system
- Owner of the file

→ Groups :

- Collection of users
- Allow to set permission on group level

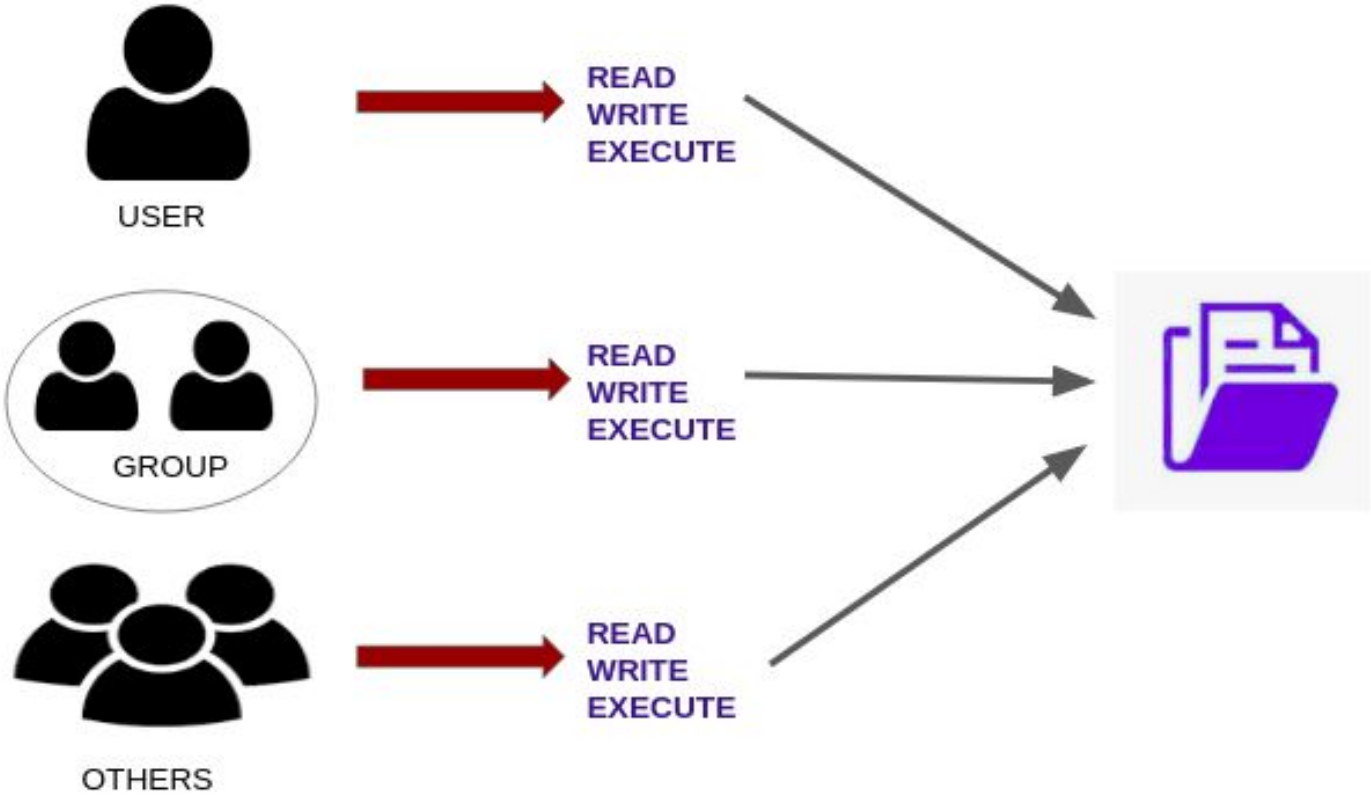
→ Others :

- It can be any other user who has access to the file
- Neither creates a file nor belongs to any usergroup

How does Linux distinguishes between three user-types?

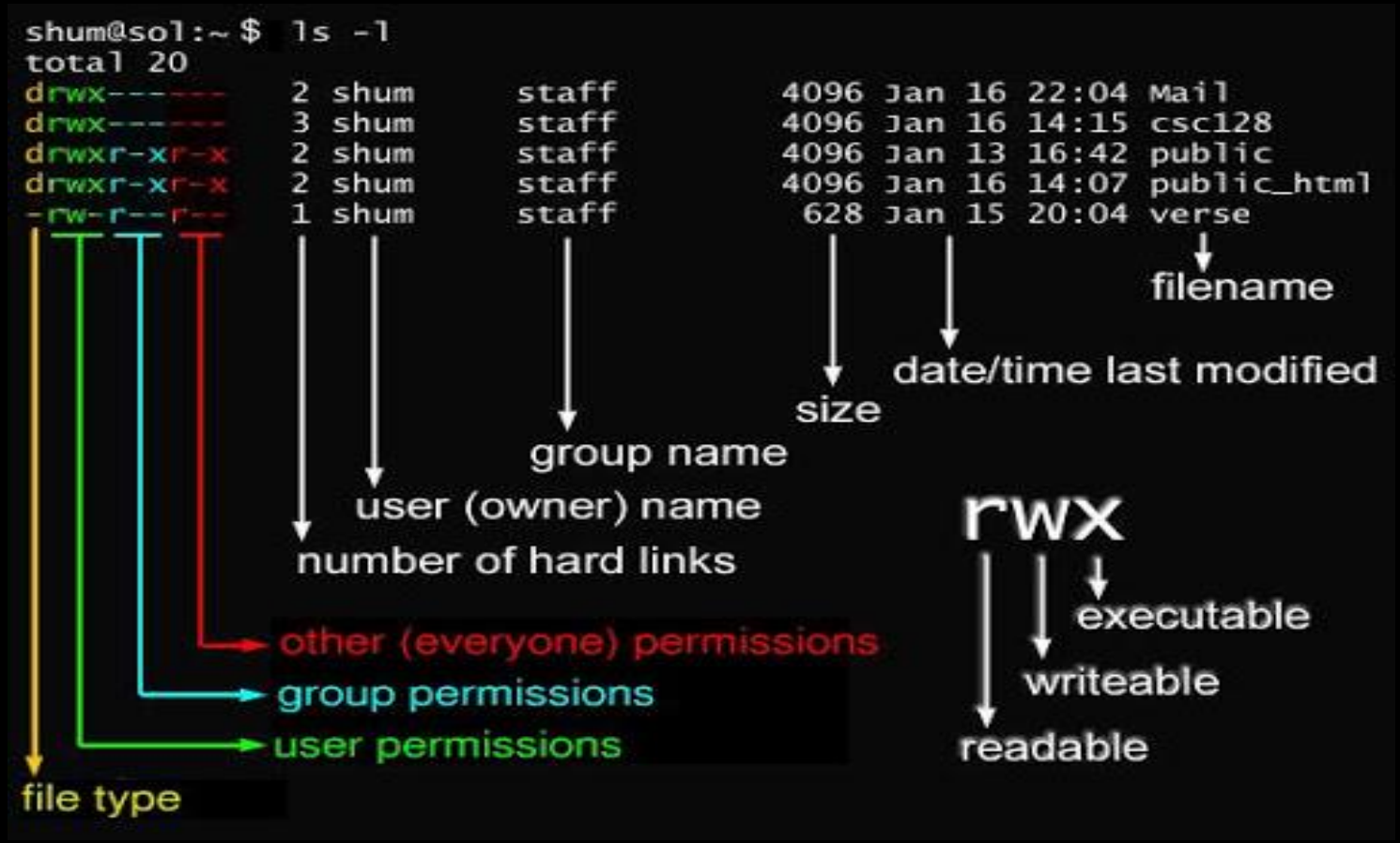
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Permission system in Linux :



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Permission system in Linux :



WLUG
COMMUNITY | KNOWLEDGE | SHARE

‘chmod’ command :

- ‘chmod’ stands for ‘change mode’ using this command we can set permissions on a file for owner / group / other.

Permission system in Linux :

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

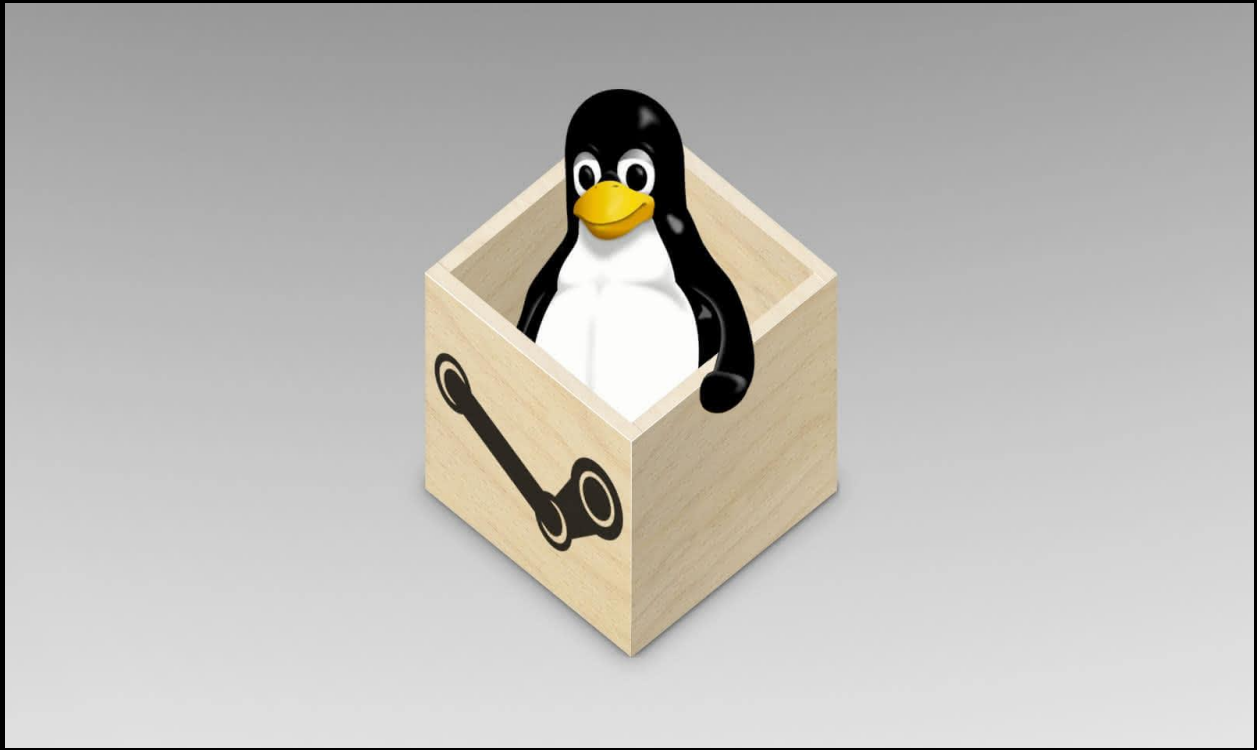
7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

chmod 777

rwx | rwx | rwx
Owner | Group | Others

WLUG
COMMUNITY | KNOWLEDGE | SHARE

VARIABLES



WLUG
COMMUNITY | KNOWLEDGE | SHARE

VARIABLES

- Variables are the containers used to store data inside them.
- These are symbolic names that represent values stored in memory.

Operations on variables:

- Create
- Access
- Unset

Creating Variables:

- Syntax :

```
#! /bin/bash
variable=value
```

- What are the rules while creating a variable?

➔ Don't put spaces on either side of equal to sign when assigning value to the variable otherwise you will get following type of error.

```
expert@vaishnavi:~/Desktop$ gedit second.sh
expert@vaishnavi:~/Desktop$ ./second.sh
./second.sh: line 3: kali: command not found
linux
ubuntu

expert@vaishnavi:~/Desktop$
```

Accessing and Unsetting variables

- For accessing variable, prefix its name with the dollar sign ($).
  Eg. echo $var_name

- Tells the shell to remove the variable from the list of variables.
  Syntax : unset var_name

Readonly variables

- Variable whose value cannot be changed after it is defined.

- Its value persist until the shell exists.

  Syntax : readonly var_name

Types of variables :

- Environment variables
- User defined variables :
    1. Scalar variables
    2. Array variables

Environment variables

- These variables are globally available to all programs in shell.

- Environmental variables govern behavior of programs in your Operating System.

- Created by using :

➔ Syntax :

export ENV_VAR_NAME=value

- For printing all env. variables :

type env

- For printing single variable :

printenv ENV_VAR_NAME

Variable	Description
$PWD	Shows the name of current working directory
$PATH	Search path for command
$USER	Current user who is logged in
$LANG	Shows the value of language that user understood
env	Displays all environmental variables

User defined variables :

1. Scalar variables
2. Array variables

- **Scalar variable** :

  A scalar variable can hold only one value at a time.

  Eg: a=”WLUG”

- **Array variable** :

  Arrays provide a method of grouping a set of variables. Following is the simplest method of creating an array variable.

  Syntax: array_name[index]=value

Printing array variable's values :

- For printing value at specific index:

```
echo ${arr_name[index]}
```

- For printing all values :

```
echo ${arr_name[$*]}
echo ${arr_name[$@]}
```

SPECIAL VARIABLES

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Variable	Meaning
$0	The filename of current script
$n	The positional argument
$#	The no. of arguments supplied to a script
$*	All the command line argument double quoted
$@	All the command line argument double quoted
$?	Exit status of the last command executed
$$	PID of current shell
$!	PID of last background process

WLUG

COMMUNITY | KNOWLEDGE | SHARE

Difference between “$*” and “$@”

- ➔ $* : It will consider all the positional arguments that we have passed as a single string.
- ➔ $@ : It will treat each argument as a seperate string.

Thank You !