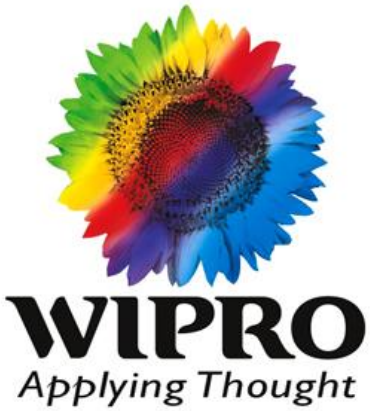


Unix Shell Scripting Basics

Module 2

Agenda

1 **Unix Utilities**

Unix Utilities

Objectives

In this session, you will learn how to:

- use the Unix utilities such as
  - echo, touch, more, file, wc, find, diff
- employ redirection operators
- use filters such as
  - sort, grep, cut, head, tail, tr, paste, uniq
- use backup commands
  - tar

cat

- cat command takes the input from the keyboard, and sends the output to the monitor
- We can redirect the input and output using the redirection operators

$ cat > file1
Type the content here
press <ctrl d>
$ cat file1
Displays the content of the file
$cat >> file1
This will append standard input to the content of file1

touch

- touch is used to change the time stamp of the file

Syntax:
touch [options] file

- Options:
  - -a to change the access time
  - -m to change the modification time
  - -c no create if not exists

- touch <file> will change the time of change of the file if the file exists

- If the file does not exist, it will create a file of zero byte size.

echo & read

- echo command is used to print output to the screen

```
echo “This is an example”
This is an example

x=10
echo $x
10
```

- read command allows to read input from user and assign it to the variable specified.

```
read x
```

The diff command

- **To compare contents of two files we can use the diff command**
- **Usage**

diff [options] file1 file2

- Options

  -i : ignores case

General Purpose Utilities

- more
  - Allows user to view one page-full of information at a time.
- file
  - Used to display the type of the file
- tty
  - Prints the terminal’s name

General Purpose Utilities (Contd.).

- wc
  - A filter used to count the number of lines, words, and characters in a disk file or from the standard input.
  - -l - displays the number of lines
  - -w - displays the number of words
  - -c - displays the number of characters

find

- Lets user to search set of files and directories based on various criteria
- Syntax: find [path...] [expression]
- [path]
  - where to search
- [expression]
  - What type of file to search (specified with –type option)
  - What action to be applied (–exec, –print, etc.)
  - Name of the files (specified as part of –name option, enclosed in “ “)
- Example

  find . –name “*.c” -print

  lists all files with .c extension from the current dir & its subdirectories

Find (Contd.).

- Finding files on the basis of file size
  - – size [+ –]n[bc]

n represents size in bytes (c) or blocks (b) of 512 bytes

find . –size 1000c	**Lists all files that are exactly 1000 bytes in size**
find . –size +1000c	Lists all files that are more than 1000 bytes in size
Find . –size -1000c	Lists all files that are less than 1000 bytes in size

Find (Contd.).

- Finding files on the basis of access time (atime) or modified time (mtime)
  - – atime [+-]n
  - – mtime [+-]n

    n represents number of days (actually 24 * n hours)

find . –atime 2	Lists files accessed exactly 2 days ago
find . –atime +2	Lists files accessed more than 2 days ago
find / -mtime -2	Lists files modified less than 2 days ago

Find (Contd.).

- Applying a command on files matching the criteria with –exec and –ok options
  - – exec *command* {} \;

    command *is command* to be applied on the matching files (does not prompt user)

    find . -name “*.dat” –exec ls –l {} \;

    Long listing of all files with .dat extension in the current and its subdirectories
  - -ok *command* {} \;

    Functionality is similar to –exec, but prompts user before applying the command on the file matching the criteria.

Standard Files

- Standard Input file
  - Keyboard, file descriptor is 0
- Standard Output file
  - Monitor, file descriptor is 1
- Standard Error file
  - Monitor, file descriptor is 2

I/ O Redirection

< file	Redirect standard input from file
> file	Redirect standard output to file
2>file	Redirect standard error to file
2>&1	Merge standard error with standard output
$ cat > abc	To enter contents in a file abc
$ cat xyz abc > outfile 2	Read files xyz and abc and redirect the output to outfile2
$ls –l > outfile	Redirect the output of command ls –l to file outfile
$ cat xyz abc > outfile2>&1	Read files xyz and abc and redirect the output to outfile2 with error output

Filters

- Filters are programs that takes its input from the standard input file, process it, and sends it to the standard output file.

- Commonly used filter commands
  - sort
  - grep
  - cut
  - head
  - tail
  - paste

sort

Sorts the contents of the given file based on the first char of each line.

-n numeric sort (comparison made according to strings numeric value)

-r reverse sort

-t specify delimiter for fields

+num specify sorting field numbers

+num [-num] to specify the range

$ sort +1 -3 emp.dat, will sort the emp.dat file on the 2nd field & 3rd field.

grep

- grep -Global Regular Expression Printer is used for searching regular expressions
- Syntax
  - grep <options> <pattern> <filename(s)>

grep options

-c	Displays count of line where the pattern occurs
-n	Displays line numbers along with the lines
-v	Displays all lines except lines matching pattern
-i	Ignore case for matching

Patterns

*	matches 0 or more characters
^pqr	Matches pqr at the beginning of the line
pqr$	Matches pqr at the end of the line
“.”	Matches any one character

Filter Command - head

Displays the first n lines of the file

$ head -3 file1

Filter Command - tail

Displays the last n lines of a file

$ tail -3 file1

Can also specify the line number from which the data has to be displayed till the end of file

$ tail +5 file1

Filter command - tr

tr - translate filter used to translate a given set of characters

Example :

tr [a-z] [A-Z] < filename

This converts standard input read from lower case to upper case.

option -s can be used to squeeze the repeated characters.

cat lcasefile| tr ‘[a-z]’ ‘[A-Z]’>ucasefile

Filter command – tr (Contd.).

Useful options for tr

- -s char
  Squeeze multiple contiguous occurrences of the character into single char

- -d char
  Remove the character

Command Piping

- Allows the output (only the standard output) of a command to be sent as input to another command.
- Multiple pipes may appear in one command line.

Example:

```
$ cat file1 | head | wc -l
```

Filter Command – tee

- tee command allows the normal output to the standard output, as well as to a file
- Useful to capture intermediate output of a long command pipeline for further processing, or debugging purpose.
- Example
  - who | tee userlist
  - cat - | tee file1 | wc -l

Filter Command – cut

Used to extract specified columns of a text

Optionremark
-c used to extract characters
-d Delimiter for fields
-f Field no.
Examples
$ cut -c2-5 file1
$ cut -d “|” -f2,3 file1

The uniq command

- To remove/eliminate duplicate lines from a file or input data stream we can use uniq command
- Usage

Syntax:

uniq [-c | -d | -u] [-f fields] [-s char] [-n] [+m] [input_file [output_file]]

Options:

-c: precede each output line with a count of the number of times the line occurred in the input.

Example:

uniq file1 > file2

Tape Archive - tar

- Tar is an utility to store and retrieve files from an archive, known as tarfile.
- Though archives are created on a tape, it is common to have them as disk files as well.
  - tar c|t|x [vf destination] source...
  - $ tar -cf tar1 emp
  - $ tar –tf tar1

Tape Archive – tar (Contd.).

Examples:

Create a new tar file containing all .dat files (assuming a.dat, b.dat and c.dat exist)

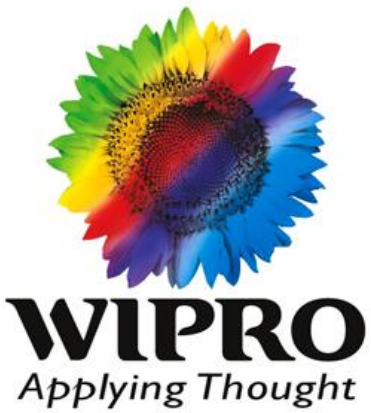
$ tar –cf mytar *.dat

- $ tar –xf mytar

Summary

In this session, you have learned to:

- use the Unix Utilities like
  - cat, echo, touch, more, file, wc, find, diff
- employ redirection operators
- use backup commands
  - tar
- use filters like
  - sort, grep, cut, head, tail, tr, uniq



Thank You