

A person wearing a dark jacket is pulling a thick white rope with blue stripes through a pulley system on a boat. The background shows a blue ocean under a blue sky with white clouds. The title text is overlaid on the image.

Text Processing & Backup Utilities in Linux

Dr. Vimal Baghel, Assistant Professor
SCSET, BU

Outline

- Basic Text Editing Commands: **cat, tac, echo, vi/vim/nano/gedit**
- Text Processing Utilities: **sort, uniq, comm, cmp, diff, tr, cut, paste, grep, sed, awk**
- Pipes and redirection: **|, >, >>, <, <<**
- Backup Utilities: **rsync**
- Q & A

Basic Text Editing Commands

- Navigate, edit, and format your text efficiently
 - \$ cat filename: display the contents of a file
 - \$ tac filename: display the contents of a file in reverse lines
 - \$ vi/vim filename ✓
 - \$ nano filename
 - \$ gedit filename
- Display a value of a variable
 - \$ var="Today is Friday!" ✓
 - \$ echo \$var ✓

sed nano

Sort: sort lines of text files

- sort alphabetically, numerically, or based on custom criteria having numbers, strings, and alphanumeric combinations.
 - `$ sort dir_list.txt`
 - `$ sort -n numbers.txt`
- Options
 - `-R` random sort
 - `-r` reverse the sort order
 - `-o` redirect sorted result to specified filename
 - `-n` sort numerically
 - `-V` version sort, aware of numbers within text
 - `-h` sort human readable numbers like 4K, 3M, etc
 - `-k` sort via key
 - `-u` sort uniquely
 - `-b` ignore leading white-spaces of a line while sorting
 - `-t` use SEP instead of non-blank to blank transition

Handwritten red notes:
Sort unique
-o output file
-b ignore leading white spaces

uniq: report or omit repeated lines

- This command is more specific to recognizing duplicates.
- Usually requires a sorted input as the comparison is made on adjacent lines only
- **Options**
 - -d print only duplicate lines
 - -c prefix count to occurrences
 - -u print only unique lines
- **\$ sort test_list.txt**
- **\$ uniq test_list.txt**
- **\$ uniq -d sorted_list.txt** print only duplicate lines
- **\$ uniq -cd sorted_list.txt** print only duplicate lines and prefix the line with number of times it is repeated
- **\$ uniq -u sorted_list.txt** print only unique lines, repeated lines are ignored

comm: compare two sorted files line by line



Without any options, it prints output in three columns - *lines unique to file1, line unique to file2 and lines common to both files*



Options

- 1 suppress lines unique to file1
- 2 suppress lines unique to file2
- 3 suppress lines common to both files



`comm -12 sorted_file1.txt sorted_file2.txt` print lines common to both files

cmp: compare two files byte by byte



Useful to compare binary files.



If the two files are same, no output is displayed (exit status 0)



If there is a difference, *it prints the first difference - line number and byte location (exit status 1)*



Option -s allows to suppress the output, useful in scripts



```
$ cmp /bin/grep /bin/fgrep
```



```
/bin/grep /bin/fgrep    differ: byte 25, line 1
```

diff: compare files line by line

```
$ diff -s test1.txt test2.txt
```

Useful to compare old and new versions of text files

All the differences are printed, which might not be desirable if files are too long

Options

- -s convey message when two files are same
- -y two column output
- -i ignore case while comparing
- -w ignore white-spaces
- -r recursively compare files between the two directories specified
- -q report if files differ, not the details of difference

tr: translate or delete characters

Options

- -d delete the specified characters
- -c complement set of characters to be replaced

tr 'a-z' 'A-Z' < file.txt

- tr a-z A-Z < test_list.txt
 - convert lowercase to uppercase
- tr -d . _ < test_list.txt
 - delete the dot and underscore characters
- tr a-z n-za-m < test_list.txt >
 - encrypted_test_list.txt Encrypt by replacing every lowercase alphabet with 13th alphabet after it
 - Same command on encrypted text will decrypt it

cut: remove sections from each line of files

For columns operations with well defined delimiters, cut command is handy

Examples

- `ls -l | cut -d' ' -f1` first column of `ls -l`
 - `-d` option specifies delimiter character; in this case it is single space character (Default delimiter is TAB character)
 - `-f` option specifies which fields to print separated by commas, in this case field 1
- `cut -d':' -f1 /etc/passwd`
 - prints first column of `/etc/passwd` file
- `cut -d':' -f1,7 /etc/passwd`
 - prints 1st and 7th column of `/etc/passwd` file with character in between
- `cut -d':' --output-delimiter=' ' -f1,7 /etc/passwd`
 - use space as delimiter between 1st and 7th column while printing

paste:
merge
lines of
files

Examples

- `paste list1.txt list2.txt list3.txt > combined_list.txt`
 - combines the three files column-wise into single file, the entries separated by TAB character
- `paste -d':' list1.txt list2.txt list3.txt > combined_list.txt`
 - the entries are separated by : character instead of TAB

Summary of Text Processing Utilities

Command	Description	Examples
cat	Concatenate and display files	cat file1.txt file2.txt
sort	Sort lines of text files	sort file.txt
uniq	Remove duplicate lines from a sorted file	sort file.txt uniq
grep	Search for patterns in files	grep "pattern" file.txt
cut	Extract columns of text from files	cut -f1,3 file.txt
sed	Stream editor for filtering and transforming text	sed 's/old/new/' file.txt
awk	Pattern scanning and processing language	awk '{print \$1, \$3}' file.txt
tr	Translate or delete characters	tr 'a-z' 'A-Z' < file.txt
wc	Count lines, words, and characters in a file	wc file.txt
diff	Compare two files and show differences	diff file1.txt file2.txt
patch	Apply a diff file to a file or directory	patch file.txt patch.diff
nl	Number lines in a file	nl file.txt
head	Display the first few lines of a file	head file.txt
tail	Display the last few lines of a file	tail file.txt
tee	Redirect output to a file and to the terminal	ls tee output.txt
fmt	Format text files for printing	fmt file.txt
pr	Convert text files for printing	pr file.txt
iconv	Convert character encoding of a file	iconv -f utf-8 -t iso-8859-1 file.txt
dos2unix	Convert DOS line endings to UNIX line endings	dos2unix file.txt
rev	Reverse lines of a file	rev file.txt
fold	Wrap lines of text to a specified width	fold -w 80 file.txt
join	Join lines from two files based on a common field	join file1.txt file2.txt

Pipes & Redirection

Piping in Linux:

- Linking one command's output to another's input is known as “piping”.
- This enables you to execute complicated activities by chaining together commands.
- The vertical bar (|) is the pipe operator.
- `$ ls | wc -l`

```
$ echo "Hello, World" | tr \[a-z\] \[A-Z\]
```

HELLO, WORLD

Redirection in Linux:

- Redirection is a way of changing a command's default input or output.
- This allows you to save the output of a command to a file or to read input from a file instead of the keyboard.
- The following operators are used for redirection purposes.

The output of a command can be redirected to a file using the `>` operator:

```
$ curl -L https://github.com/kubernetes/kubernetes/blob/master/README.md > README.md
```

The `>>` operator is used to write to a new file or append a command's output to the end of an already-existing file.

```
$ cat numbers.txt  
one  
two
```

```
$ echo "three" >> numbers.txt
```

```
$ cat numbers.txt  
one  
two  
three
```

The `<` operator reads input from a file and then acts upon it.

```
$ wc -l < numbers.txt > lines.txt
```

The operator called `>>` redirects the errors to your desired file.

```
$ docker ps >> error.txt
```

Why to backup?

- Important to prevent permanent data loss on PCs/servers.
- To know different backup tools is very important especially for **System Administrators**.
- Backup can either be done manually or configured to work automatically.
- Many backup utilities have different features that allow users to configure the
 - type of backup, ✓
 - time of backup, ✓
 - what to backup, ✓
 - logging backup activities and many more ✓

rsync

a command-line backup tool popular among Linux users especially System Administrators.

It performs incremental backups, update whole directory tree and file system, both local and remote backups, preserves file permissions, ownership, links and many more.

It also has a GUI called **Grsync** but one advantage with the rsync is that backups can be automated using scripts and cron jobs

rsync(1)

rsync(1)

NAME

rsync - a fast, versatile, remote (and local) file-copying tool

SYNOPSIS

Local: rsync [OPTION...] SRC... [DEST]

Access via remote shell:

Pull: rsync [OPTION...] [USER@]HOST:SRC... [DEST]

Push: rsync [OPTION...] SRC... [USER@]HOST:DEST

Access via rsync daemon:

Pull: rsync [OPTION...] [USER@]HOST::SRC... [DEST]

rsync [OPTION...] rsync://[USER@]HOST[:PORT]/SRC... [DEST]

Push: rsync [OPTION...] SRC... [USER@]HOST::DEST

rsync [OPTION...] SRC... rsync://[USER@]HOST[:PORT]/DEST

Usages with just one SRC arg and no DEST arg will list the source files instead of copying.

DESCRIPTION

Rsync is a fast and extraordinarily versatile file copying tool. It
Manual page rsync(1) line 1 (press h for help or q to quit)

rsync

```
# rsync options source destination
```



-v : verbose



-r : copies data recursively (don't preserve timestamps and permission while transferring data).



-a : archive mode allows copying files recursively and preserves symbolic links, file permissions, user & group ownerships, and timestamps.



-z : compress file data.



-h : human-readable, output numbers in a human-readable format.

```
$ sudo apt-get install rsync [On Debian/Ubuntu & Mint]
$ pacman -S rsync [On Arch Linux] $ emerge sys-apps/rsync [On Gentoo]
$ sudo yum install rsync [On Fedora/CentOS/RHEL and Rocky Linux/AlmaLinux]
$ sudo zypper install rsync [On openSUSE]
```

Copy/Sync Files and Directory Locally

Copy/Sync a File on a Local Computer

```
[root@tecmint:/home/tecmint]# rsync -zvh backup.tar.gz /tmp/backups/  
created directory /tmp/backups  
backup.tar.gz  
  
sent 224.54K bytes  received 70 bytes  449.21K bytes/sec  
total size is 224.40K  speedup is 1.00  
[root@tecmint:/home/tecmint]# _
```

Copy/Sync a Directory on Local Computer

```
[root@tecmint:~]# rsync -avzh /root/rpmpkgs /tmp/backups/  
sending incremental file list  
rpmpkgs/  
rpmpkgs/httpd-2.4.37-40.module_el8.5.0+852+0aaafc63b.x86_64.rpm  
rpmpkgs/mod_ssl-2.4.37-40.module_el8.5.0+852+0aaafc63b.x86_64.rpm  
rpmpkgs/nagios-4.4.6-4.el8.x86_64.rpm  
rpmpkgs/nagios-plugins-2.3.3-5.el8.x86_64.rpm  
  
sent 3.47M bytes  received 96 bytes  2.32M bytes/sec  
total size is 3.74M  speedup is 1.08  
[root@tecmint:~]# _
```

Copy a Directory from Local Server to a Remote Server

```
[root@tecmint:~]# rsync -avzh /root/rpmpkgs root@192.168.0.141:/root/
The authenticity of host '192.168.0.141 (192.168.0.141)' can't be established.
ED25519 key fingerprint is SHA256:bH2tiWQn4S5o6qmZhmtXcBR0V5TU5H4t2C42QDEMx1c.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.141' (ED25519) to the list of known hosts.
root@192.168.0.141's password:
sending incremental file list
rpmpkgs/
rpmpkgs/httpd-2.4.37-40.module_el8.5.0+852+0aaafc63b.x86_64.rpm
rpmpkgs/mod_ssl-2.4.37-40.module_el8.5.0+852+0aaafc63b.x86_64.rpm
rpmpkgs/nagios-4.4.6-4.el8.x86_64.rpm
rpmpkgs/nagios-plugins-2.3.3-5.el8.x86_64.rpm

sent 3.74M bytes  received 96 bytes  439.88K bytes/sec
total size is 3.74M  speedup is 1.00
[root@tecmint:~]# _
```

Copy/Sync a Remote Directory to a Local Machine

```
[root@tecmint:~]# rsync -avzh root@192.168.0.141:/root/rpmpkgs /tmp/myrpms
root@192.168.0.141's password:
receiving incremental file list
created directory /tmp/myrpms
rpmpkgs/
rpmpkgs/httpd-2.4.37-40.module_el8.5.0+852+0aaafc63b.x86_64.rpm
rpmpkgs/mod_ssl-2.4.37-40.module_el8.5.0+852+0aaafc63b.x86_64.rpm
rpmpkgs/nagios-4.4.6-4.el8.x86_64.rpm
rpmpkgs/nagios-plugins-2.3.3-5.el8.x86_64.rpm

sent 104 bytes  received 3.49M bytes  997.68K bytes/sec
total size is 3.74M  speedup is 1.07
[root@tecmint:~]# _
```



Thanks

Q & A