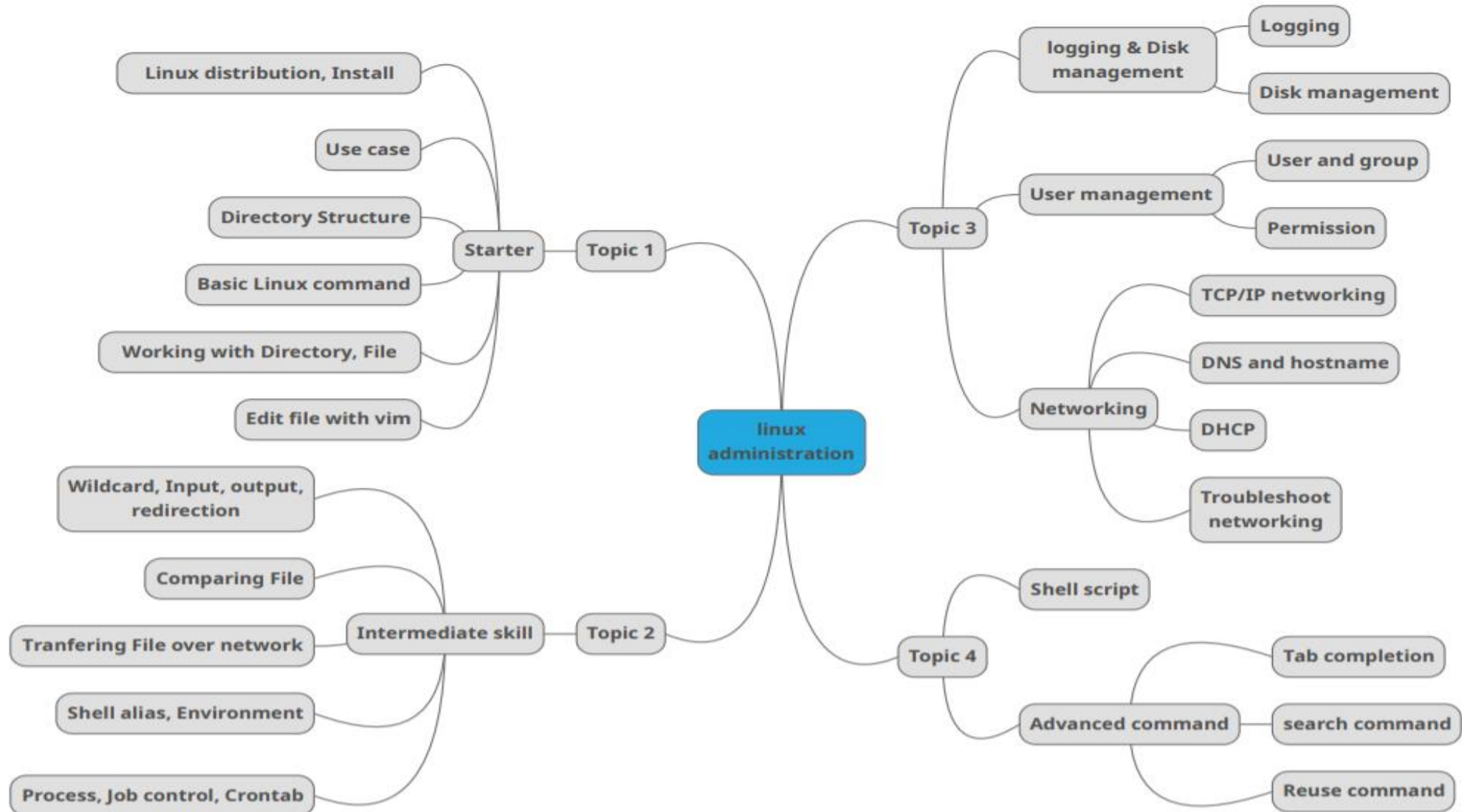




Linux administration

Techmaster

Nội dung



Deleting, Copying, Moving, and Renaming Files

What You Will Learn



- Deleting files
- Copying files
- Moving files
- Renaming files
- Compressing files
- Creating archives

Remove file



```
rm file
```

Remove file.

```
rm -r dir
```

Remove the directory and its contents recursively.

```
rm -f file
```

Force removal and never prompt for confirmation.

Copy file



```
cp source_file destination_file
```

Copy source_file to destination_file.

```
cp src_file1 [src_fileN ...] dest_dir
```

Copy source_files to destination_directory.

cp Option



```
cp -i
```

Run in interactive mode.

```
cp -r source_directory destination
```

Copy src_directory recursively to destination.

Moving and Renaming files



`mv` Move or rename files and directories.

```
mv source destination
```

```
mv -i source destination
```


Sort Options



- `k F` Sort by key. `F` is the field number,
- `r` Sort in reverse order.
- `u` Sort unique.

Creating a Collection of files



```
tar [-] c|x|t f tarfile [pattern]
```

Create, extract or list contents of a tar archive using pattern, if supplied.

tar Options



| | |
|--------|---------------------------------------|
| c | Create a tar archive. |
| x | Extract files from the archive. |
| t | Display the table of contents (list). |
| v | Be verbose. |
| z | Use compression. |
| f file | Use this file. |

Compressing Files to save space



`gzip`

Compress files.

`gunzip`

Uncompress files.

`gzcat`

Concatenates compressed files.

`zcat`

Concatenates compressed files.

Wildcards

What You Will Learn



- What wildcards are.
- When and where they can be used.
- The different types of wildcards.
- How to use wildcards with various commands.

Wildcards



- A character or string used for pattern matching.
- Globbing expands the wildcard pattern into a list of files and/or directories. (paths)
- Wildcards can be used with most commands.
 - ls
 - rm
 - cp

Wildcards



- * - matches zero or more characters.
 - *.txt
 - a*
 - a*.txt
- ? - matches exactly one character.
 - ?.txt
 - a?
 - a?.txt

Character Class



- `[]` - A character class.
 - Matches any of the characters included between the brackets. Matches exactly one character.
 - `[aeiou]`
 - `ca[nt]*`
 - can
 - cat
 - candy
 - catch

Ranges



- Use two characters separated by a hyphen to create a range in a character class.
- `[a-g]*`
 - Matches all files that start with a, b, c, d, e, f, or g.
- `[3-6]*`
 - Matches all files that start with 3, 4, 5 or 6.

Named Character Class



- `[[:alpha:]]`
- `[[:alnum:]]`
- `[[:digit:]]`
- `[[:lower:]]`
- `[[:space:]]`
- `[[:upper:]]`

IO Redirection

What You Will Learn



- Three I/O Type
- Redirection

Input/Output Types



| I/O Name | Abbreviation | File Descriptor |
|-----------------|--------------|-----------------|
| Standard Input | stdin | 0 |
| Standard Output | stdout | 1 |
| Standard Error | stderr | 2 |

Redirection



- > Redirects standard output to a file.
Overwrites (truncating) existing contents.
- >> Redirects standard output to a file.
Appends to any existing contents.
- < Redirects input from a file to a command.

Redirection



`&`

Used with redirection to signal that a file descriptor is being used.

`2>&1`

Combine stderr and standard output.

`2>file`

Redirect standard error to a file.

The null device



`>/dev/null` Redirect output to nowhere.

```
$ ls here not-here 2> /dev/null  
here
```

```
$ ls here not-here > /dev/null 2>&1  
$
```

Comparing Files

Comparing the contents of Files



```
diff file1 file2
```

Compare two files.

```
sdiff file1 file2
```

Side-by-side comparison.

```
vimdiff file1 file2
```

Highlight differences in vim.

Diff Output



```
$ diff file1 file2
```

```
3c3
```

```
...
```

LineNumFile1-Action-LineNumFile2

Action = (A)dd (C)hange (D)elete

vimdiff



Ctrl-w w

Go to next window

:q

Quit (close current window)

:qa

Quit all (close both files)

:qa!

Force quit all

Searching in files

Grep command



grep Display lines matching a pattern

```
grep pattern file
```

Grep options



- i Perform a search, ignoring case.
- c Count the number of occurrences in a file.
- n Precede output with line numbers.
- v Invert Match. Print lines that don't match.

file command



`file file_name` **Display the file type.**

```
$ file sales.data
```

```
sales.data: ASCII text
```

```
$ file *
```

```
bin: directory
```

```
jason.tar: POSIX tar archive
```

Transferring and Copying Files

Copying file over network



- SCP - Secure copy
- SFTP - SSH file transfer protocol

Command Line SCP Clients



- `scp`
- `sftp`
- PuTTY Secure Copy client - `pscp.exe`
- PuTTY Secure File Transfer client - `psftp.exe`

Scp/sftp command line



```
scp source destination
```

Copy source to destination.

```
sftp host
```

Start a secure file transfer session with host.

```
sftp jason@host
```

ftp command line



```
ftp host
```

Start a file transfer session with host.

Alias



- Shortcuts
- Use for long commands
- Use for commands you type often

Creating alias



```
alias [name[=value]]
```

List or create aliases.

Use name=value to create a new alias.

alias



- Fix Typos

```
$ alias grpe='grep'
```

- Make Linux behave like another OS.

```
$ alias cls='clear'
```

Removing Alias



```
unalias name
```

Remove the “name” alias.

```
unalias -a
```

Remove all aliases

Persisting Aliases



- Add aliases to your personal initialization files.
 - `.bash_profile`

Environment Variables

Common Environment Variables



| Variable | Description |
|----------|---|
| EDITOR | Program to run to perform edits |
| HOME | Home directory of the user. |
| LOGNAME | The login name of the user. |
| MAIL | The location of the user's local inbox. |

Common Environment Variables



| Variable | Description |
|----------|--|
| OLDPWD | The previous working directory. |
| PATH | A list of directions to search for commands. |
| PAGER | Program used to paginate through files. |

View environment variables



`printenv` - Print all or part of environment.

`echo $ENV_VAR` - Print the `ENV_VAR` variable.

Process and Job control



What you will learn



- List running processes.
- Foreground vs background processes.
- Launch background processes.
- Kill processes.

Listing Processes and Information



ps

Display process status.

Ps options



- e Everything, all processes.
- f Full format listing.
- u username Display username's processes.
- p pid Display information for PID.

Common Ps commands



| | |
|-----------------------------|------------------------------|
| <code>ps -e</code> | Display all processes. |
| <code>ps -ef</code> | Display all processes, full. |
| <code>ps -eH</code> | Display a process tree. |
| <code>ps -e --forest</code> | Display a process tree. |
| <code>ps -u username</code> | Display user's processes. |

Other ways to view processes



| | |
|----------------------|-------------------------------------|
| <code>ps tree</code> | Display processes in a tree format. |
| <code>top</code> | Interactive process viewer. |
| <code>htop</code> | Interactive process viewer. |

Background and Foreground processes



| | |
|----------------------------|---------------------------------|
| <code>command &</code> | Start command in background. |
| <code>Ctrl-c</code> | Kill the foreground process. |
| <code>Ctrl-z</code> | Suspend the foreground process. |

Background and Foreground processes



| | |
|--------------------------|--------------------------------------|
| <code>bg [%num]</code> | Background a suspended process. |
| <code>fg [%num]</code> | Foreground a background process. |
| <code>kill</code> | Kill a process by job number or PID. |
| <code>jobs [%num]</code> | List jobs. |

Background and Foreground processes



`Ctrl-c`

Kills the foreground proc.

`kill [-sig] pid`

Send a signal to a process.

`kill -l`

Display a list of signals.

`kill 123`

`kill -15 123`

`kill -TERM 123`

`kill -9 123`

Schduling Repeated Job with Cron

What you will learn



- Cron service
- Crontab format
- Crontab command

Cron



- `cron` - A time based job scheduling service.
- `crontab` - A program to create, read, update, and delete your job schedules.
- Use `cron` to schedule and automate tasks.

Cron Format



* * * * * command

| | | | |

| | | | +-- Day of the Week (0-6)

| | | +---- Month of the Year (1-12)

| | +----- Day of the Month (1-31)

| +----- Hour (0-23)

+----- Minute (0-59)

Example



```
# Run every Monday at 07:00.
```

```
0 7 * * 1 /opt/sales/bin/weekly-report
```

Example

```
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
| | | | |
| | | | +-- Day of the Week (0-6)
| | | +---- Month of the Year (1-12)
| | +----- Day of the Month (1-31)
| +----- Hour (0-23)
+----- Minute (0-59)
```

Redirecting Output



```
# Run at 02:00 every day and  
# send output to a log file.  
0 2 * * * /root/backupdb > /tmp/db.log 2>&1
```

Example



```
# Run every 30 minutes.
```

```
0,30 * * * * /opt/acme/bin/half-hour-check
```

```
# Another way to do the same thing.
```

```
*/2 * * * * /opt/acme/bin/half-hour-check
```

```
# Run for the first 5 minutes of the hour
```

```
0-4 * * * * /opt/acme/bin/first-five-mins
```