Linux, and BASH Scripting

Alexander Zerpa

Section 1

Linux, and BASH Scripting

Section 2

Filesystem

Filesystem

Everything is a file!

The main way of interacting with the Linux filesystem is through the cli.

Navigation

- pwd: get the current working directory.
- ls: list files and directories.
- cd: to change the current working directory.

Files

File information

- file: get file type.
- stat: list file information and metadata.

File content

- cat: output contents of file.
- head: output first 10 lines of file.
- tail: output last 10 lines of file.

Manipulation

- mkdir: make a new empty directory.
- touch: create empty files.
- rm: remove files and directories.
- mv: moving files and renaming.
- cp: coping files.

Links

symbolic

Symbolic links are relative and moving o deleting the original file will break them.

```
ln -s file link
```

hard

Hard links point the new file to the same location in memory so as long as one link exist the rest will continue to work.

```
ln file link
```

Editors

nano

Simple text editor for quick use on the cli.

Useful commands

^O	saves file
^X	exits the editor

VI (vim or neovim)

Focused on edition with different modes and commands.

Modes

- Normal: imputing commands for editing.
- Insert: actually writing text.
- Visual: selecting text with vi motions.

Useful commands

:q!	exit without saving
:wq	saves and exits
:e file	opens file for editing

Compression and Packaging

zip

Cross-platform compression and archival.

```
zip out.zip files      # zip files
unzip in.zip -d path   # unzip files on path
```

tar

Widely used on Linux.

```
tar -cf out.tar files       # packs files
tar -czf out.tar.gz files   # packs and compress
tar -xf in.tar -C path      # unpacks archive on path

# -a can be use instead of -z and tar will compress based
↪  on the file extension.
```

Finding Files

find

Find files starting at path.

Useful options

-type	only match specific file type
-name	name or pattern to match
-L	follow symbolic links

locate

Find files keeping track of filesystem on a database.
The database is updates daily by a cron job.

Useful options

-i	case insensitive search
-e	only show existing files
–regex	for use regex on search

Permissions and Ownership

Owner

- User
- Group
- Other

Permissions

- Read
- Write
- Execute

chown

Changes the user and group a file belongs to.

chmod

Changes permissions for a file (can use symbolic or octal representation).

- Symbolic representation: `-rwxrwxrwx`
- Octal representation: `0777`

Storage Management

Both accept the `-h` options to make size units human readable.

df

For viewing space and usage of different mount points on the filesystem.

du

For listing space use by files.

Section 3

Shell

Alias

Names that the shell translates into commands.

```
alias ll='ls -l'
alias la='ls -a'

# Use unalias to unset an alias while on the shell.
```

History

history

Shows history with numerated commands.

! (bang operator)

!!	repeats last commands
!*	last command arguments
!n	nth command in history
!-n	same as !n but in reverse order
!name	last command with same name

Streams and Redirection

Streams

name	Id
stdIn	0
stdOut	1
stdErr	2

Pipes

`|` to send stdOut to stdIn of next command.

Redirection

To redirect a stream to another use the file descriptors.

```
# redirects stdErr to errfile
command 2> errfile
# redirects stdIn and stdErr to
↪  allfile
command &> allfile
# pipes stdIn and stdErr to nextcmd
command |& nextcmd
```

To files

`>` to write `>>` to append

tee

Duplicates stdIn to file and stdOut

String manipulation

sort

Sorts lines on stdIn.

sed

Stream editor, mostly use to replace text.

awk

Powerful stream editor with it's own language.

```
# prints first field of every line on stdIn
command | awk '{print $1}'
```

Section 4

Processes

Shell Processes

A shell can execute commands in the background whether by adding a `&` after a command or by pressing `^Z`.

If using `^Z` the process will be suspended and you'll need to enter the `bg` command to continue it.

jobs

List the background processes of the shell.

fg

Bring a shell subprocess to the foreground.

bg

Resume background process of the shell.

watch

Reruns command on repeated intervals of time.

Processes

ps

List running processes with their ids.

pidof

Prints the process id matching a string.

kill

Send signals to processes, generally `SIGTERM` or `SIGINT`.

killall

Same as kill but to all processes that match name.

top

Terminal interface for process management.

There are some new implementations that build on top of `top`.

- htop
- btop
- gtop

All slightly differ in their interfaces but all allow to search processes, view info and consumption, and to send signals.

Section 5

System

User

useradd

Adds a new user.

-g	set user group
-G	assign multiple groups
-m	make home directory
-u	set specific UID
-s	assign user default shell

usermod

Modifies a user properties.

-g	set primary user group
-G	assign multiple groups
-a	append groups set by `-G`
-d	changes user home directory
-l	change name of user
-c	change full name of user
-s	change user default shell

userdel

Removes a user.

-r	remove home directory
-f	force delete

System Information

uname

List info about the OS.

hostname

View and set hostname and domainname.

lscpu

List cpu information.

/proc/cpuinfo

File containing more info about cpu and cores.

Scheduling

at

Execute a one time command at a set time.

batch

Same as `at` but when cpu load is bellow a threshold.

cron

For repeatable jobs, usable at user and root level.

Systemd

Init program mainstream on linux.

Services

`systemctl` to manage systemd units.

command	description
status	check service information
start	start service
stop	stop service
restart	same as stop and then start
enable	start at startup
disable	not start at startup
mask	makes service impossible to load
unmask	reverts mask action

root

The root user is superuser and have no restrictions.

su

Changes the current user to other, generaly to root.

sudo

Allows for the execution of commands as root

alternatives

As sudo is a SUID binary there are concerns about security.

- doas: FreeBSD alternative to sudo, smaller program
- run0: new systemd implementation that uses and isolated PTY to run commands

Section 6

Net

Configuration

`ifconifg` Manage interfaces, primarily ip-addresses, masks, and MACs.

```
# show info of all interfaces
ifconfig -a
# show info of interface
ifconfig interface
# set ip and mask of interface
ifconfig interface ip netmask mask
# set mac-address of interface
ifconfig interface hw class mac
# enables interface
ifconfig interface up
# disables interface
ifconfig interface down
```

Querying

ping

Use to test connection with remote.

dig

Queries DNS records of site.

nslookup

Same as dig but uses internal resolver libraries.

traceroute

Shows path a connection takes to remote.

File Sharing

`remote` is in the form `user@host:path`.

`file` and `remote` can be used interchangeably.

scp

Copies files over `ssh`.

```
# copies file to path
scp file remote
# port for ssh conection
scp -P port file remote
# conserve metadata
scp -p file remote
```

rsync

Transfers files as scp except that only sends the difference.

```
# sync file to path in host
rsync file remote
# transfer in archive mode
rsync -a file remote
# compress file data
rsync -z file remote
# remote shell command
rsync -e cmd file remote
```

Downloaders

curl

`curl` transfers data from a server.

```
# downloads contents of url
curl url
# saves contents to path
curl -o url path
# same name as in server
curl -O url
# follows redirects
crul -L url
# download using proxy
curl -x host:port url
```

wget

Downloads files from web.

```
# download file in url
wget url
# saves as file as name
wget -O name url
# saves file in path
wget -P path url
# downloads in background
wget -b url
# outputs file to stdOut
wget -q -O - url
```