Linux Operating System

Viettel Digital Talent 2024 - Software & Data Engineering

Huan Phan

Lesson Outline

1. Linux Kernel deep dive.
    - Introduction
    - Process
    - Kernel synchronization
    - Virtual Filesystem
2. **Linux command lines.**

Prerequisite

1. Install docker
2. Download VT folder
3. Run Ubuntu by docker with VT folder mounted, access bash terminal and install vim

```
docker run -it -v $(pwd)/VT:/VT  mattrayner/lamp:latest-1804 bash
```

```
apt-get update
```

```
unminimize
```

Basic commands: ls, pwd, cd, clear

`ls`

- `a` show all files including hidden one.
- `l` show files and metadata (size, permission, time…).
- `lh` show sizes in a human-readable format.
- `lS` show file size in decreasing order.

`pwd` print the current folder

`cd` navigate to a folder

`cd ..` navigate to previous folder

`clear` clear the terminal

Tip: create alias

Create alias in ~/.bashrc file

```
vim ~/.bashrc
    alias l=”ls -lSh”
    :wq
source ~/.bashrc
```

Basic commands: wc, head, cut, grep, sort, uniq

`wc`

`wc -l`

`wc -c`

`wc -w`

```
$ wc -l animals.txt
7 animals.txt
$ wc -w animals.txt
51 animals.txt
$ wc -c animals.txt
325 animals.txt
```

Pipe

use | to send output of a command to another command.

1. Count number of file in my current directory

```
ls -l | wc -l
```

Basic commands: wc, head, cut, grep, sort, uniq

head - prints the first lines of a file

```
head -n3
```

```
head -n3 animals.txt | wc -w
```

```
ls -l /bin | head -n5
```

```
tail -n1
```

Practice

- https://www.hackerrank.com/challenges/text-processing-in-linux---the-middle-of-a-text-file
-

Basic commands: wc, head, cut, grep, sort, uniq

cut - prints one or more columns from a file

`cut -f2` prints 2nd field from each line.

`cut -f1,3` prints 1st and 3rd field from each line.

`cut -f2-4` prints 2nd, 3rd, 4rd field from each line.

`cut -c2` prints 2nd character from each line.

`cut -c1,3`

`cut -d ' ' -f1` prints 1st field, delimiter is space

Practice

In Linux, username and password (encrypted) are stored in /etc/passwd.

Print all username and sort them

Practice

- https://www.hackerrank.com/challenges/text-processing-cut-8?isFullScreen=true
- https://www.hackerrank.com/challenges/text-processing-cut-9/problem?isFullScreen=true
-

Basic commands: wc, head, cut, grep, sort, uniq

grep - prints lines that match a given string

```
grep Nutshell animals.txt

grep -v Nutshell animals.txt

grep Perl *.txt

grep -w the star.txt

grep -i the star.txt
```

```
grep his frost

grep -w his frost

grep -i his frost

grep '^[A-Z]' frost

grep "house\|snow" frost

grep -v '^$'
```

Practice

Count number of directory under current directory

Practice

- https://www.hackerrank.com/challenges/text-processing-in-linux-the-grep-command-3?isFullScreen=true
-

Basic commands: wc, head, cut, grep, sort, uniq

Command #5: sort - reorders the lines of a file into order

```
sort animals.txt

sort -r animals.txt

sort -n

sort -n -r
```

Practice

Get the year of the most recent book in animals.txt

Basic commands: wc, head, cut, grep, sort, uniq

uniq: detects repeated, adjacent lines in a file, removes the repeats line by default.

`uniq`

`uniq -c` count occurrences

`uniq -u` print unique lines only

Basic commands: wc, head, cut, grep, sort, uniq

print the grade with the most occurrences from grade.txt file

Basic commands: wc, head, cut, grep, sort, uniq

print the grade with the most occurrences from grade.txt file

```
cut -f1 grades | sort | uniq -c | sort -nr | head -n1 | cut -c9
```

md5sum

compute md5 hash of a file content

```
cd VT/Images

md5sum image1.jpg

md5sum *.jpg
```

Practice

Find all duplicate files in VT/Images

Basic command: history

Show the command history

`history`

`history 3` show 3 most recent commands in history

`history | grep cd` show commands in history that contains word “cd”

Advanced commands

Producing text commands:

1. date - print the current date and/or time in various formats

```
date Default format
date +%Y-%m-%d Year-Month-Day format
date +%H:%M:%S Hour:Minute:Seconds format
date +"Hello Viettel, today is %A!"
```

Advanced commands

Producing text commands:

2. seq - print a sequence of numbers in a range

`seq 1 5` print numbers from 1 to 5
`seq 1 2 10` print numbers with increment is 2, i.e. 1 3 5 7 9
`seq 3 -1 0` print 3 2 1
`seq -s/ 1 5` print 1/2/3/4/5

Advanced commands

Producing text commands:

3. find - list files in a directory **recursively**

`find /etc` print all files and directory under /etc folder.
`find /etc -type f` print all files under /etc folder.

`find /etc -type d` print all directory under /etc folder.

`find /etc -type f -name "*.py"` print all Python files under /etc folder.

apply a command to the output of find using -exec and add ";" at the end.

```
find /etc -type f -name "*.conf" -exec ls -l {} ";"
```

Practice

delete all jpg files under VT/Images folder

Advanced commands

tr - Translates characters into other characters

```
echo $PATH | tr : "\n"

echo efficient | tr a-z A-Z

echo Efficient | tr A-Z a-z

echo Hello Linux | tr " " "\n"
```

Advanced commands

sed: find & replace string using regex

```
sed ‘s/REGEXP/REPLACEMENT/FLAGS’ filename
```

FLAGS:

- g: Replace all
- n: replace nth instance
- p: prints the new pattern space if substitution was made
- i: match in a case-insensitive manner
- w: if substitution was made, write out the result to given file

```
sed 's/[pP]ython/PYTHON/gpw output.txt' VT/animals.txt
```

Verify regex: regexr.com

Bash script

A bash script is a series of commands written in a file.

file extension: .sh

start with #! /bin/bash

More tips

Bash Script

Hello world bash script

`touch hello_world.sh` create a bash script file

`vim hello_world.sh`

```
    #! /usr/bin/bash
    echo "Hello World"
```

`chmod u+x hello_world.sh` allow execution

`./hello_world.sh` execute the script

`bash hello_world.sh` another way to execute

Bash Script

Basic syntax:

1. define variables

```
#!/bin/bash
name=Quang
age=19
echo “Hello world, my name is $name. I am $age”
```

Bash Script

2. arithmetic Expressions

```
+
-
*
/
**
%
var=$((expression))
```

Bash Script

3. read input and print output

```
read variable_name

read -p "Enter your name" name

for i in {1..10}; do echo "$i"; done

printf "%.2f\n" 3.14158
```

Bash Script

```
#! /usr/bin/bash
read -p "enter your name: " name
echo "Hello World, my name is $name."
```

Bash Script

```
#! /usr/bin/bash
read -p "enter your name: " name
echo "Hello World, my name is $name."
```

Practice

Write a bash script that receive 2 integers and output the sum.

Bash Script

4. condition and comparison

```
num1 -eq num2
num1 -ge num2
num1 -gt num2
num1 -le num2
num1 -lt num2
num1 -ne num2
-a
-o
```

```
if [[ conditions ]]
then
   commands
elif [[ condition ]]
    commands
else
    commands
fi
```

Bash Script

```
read x
read y

if [ $x -gt $y ] then
    echo "$x is greater than $y"
elif [ $x -lt $y ]then
    echo "$x is less than $y"
elif [ $x -eq $y ] then
    echo "$x is equal to $y"
fi
```

Practice

- https://www.hackerrank.com/challenges/bash-tutorials---more-on-conditionals/problem?isFullScreen=true

Bash Script

5. loop

```
for i in {1..10}
do
    commands
done
```

```
while [[ condition ]];
do
    commands
done
```

Bash Script

```
#!/bin/bash
i=1
while [[ $i -le 10 ]] ; do
  (( i += 1 ))
done
echo i
```

Practice

- https://www.hackerrank.com/challenges/bash-tutorials---looping-with-numbers/problem?isFullScreen=true
-

Practice

Use for loops to display only odd natural numbers from 1 to 99

Practice

Given N integers, read them from console and print the sum.

e.g.
4
1
2
9
8
output: 20

Process management: ps, kill

ps: displays the currently running processes

- PID: process ID
- TIME: total time process has been running
- CMD: command that launches the process
- STAT: state of process

State of process

- Running or Runnable (R)
- Uninterruptible Sleep (D)
- Interruptible Sleep (S)
- Stopped (T)
- Zombie (Z)
- More

Process management: ps, kill

`ps -ef (=ps aux)`: all processes of all users

`ps -u $USER`: list processes owned by current user

`ps -C <process_name> -o pid=`: find PID of <process_name>

`ps -fp <PID>`: list processes by PID

`ps -efL`: show threads

`ps -e -o pid,uname,pcpu,pmem,comm`: show processes with only selected columns

`ps -ejHf`: display process tree

Practice

Run `VT/zombie` executable file then use `ps -efjH a` to observe its behavior

Run command in the background

`<command> &`: run command in current session, get killed if logout

`nohup ./my-shell-script.sh &`: session independent

`screen`: support detach and resume a session

```
nohup /run.sh > nohup.log 2>&1 &
```

Practice

Count number of threads belonging to mysqld process

Process management: ps, kill

`kill`: send signal to a process

`kill -L`: list all the signals (ref)

`kill [pid]`: kill process by SIGTERM (gracefully kill)

`kill -9 [pid]`: kill process by SIGKILL (immediately kill)

Practice

Use ps, grep & kill to kill this background task below

```
$ while true; do echo "Writing to file" && date '+At: %H:%M:%S, %m/%d/%y'
>> /var/log/crontab.log && sleep 10; done > /var/log/run.log 2>&1 &
```

crontab

`crontab`: schedule task

`crontab -e`: edit the crontab

`crontab -l`: list crontab entries

`crontab -r`: remove the crontab

`0 2 * * *`: run at 02:00 every day

`30 22 * * 6`: run at 23:30 every Sat

Verify cron: crontab.guru

Format

```
# ┌─ minute (0-59)
# │ ┌─ hour (0-23)
# │ │ ┌─ day of the month (1-31)
# │ │ │ ┌─ month (1-12)
# │ │ │ │ ┌─ day of the week (0-6)
(Sun->Sat)
# * * * * * <command>
```

Special

```
*: any value
,: list of values
-: range of values
/: step values
```

Practice

Run command E every 30th minute of hour 3,4,5 from Monday to Friday?

User management: useradd, groupadd

`adduser <username>` : add new user with prompt

`useradd -s <shell> -m -d <homedir> -c <description> -g <group> -m <username>` : add new user with parameters

`useradd -N <username>` : add new user to default group

`passwd <username>` : change user password

`cat /etc/passwd` : see user information

`cat /etc/group` : see group

`id <user>` : see user ID

`userdel <user>` : delete user

User management: useradd, groupadd

`groupadd <group>`: add a new group

`cat /etc/group`: see user group

`usermod -g <group> <user>` : add an user to an existing group

`groupdel <group>`: delete a group

User management: useradd, groupadd

`su - <username>`: switch user

`sudo <command>`: run command as root

`visudo -g <group> <user>` : grant a group the root permissions

`groupdel <group>`: delete a group

Monitoring: top, netstat

`top`: displays real time information about various performance metrics of the system

- `k: kill`
- `c: full command`
- `M: sort by %MEM`
- `N: sort by PID`
- `P: sort by %CPU`
- `T: sort by TIME+`
- `R: reverse sort ordering`
- `V: tree mode`

`htop:` better UI/UX

Monitoring: top, netstat

`netstat`: show network connections, routing tables, interface statistics, etc.

`apt-get install net-tools`

`netstat -an:` active connections & domain sockets

`netstat -ap:` active connections with PID

`netstat -l:` connections in LISTEN state

`netstat -tu:` tcp & udp connections

`netstat --route:` routing table

`netstat -ap | grep mysql:` connection used by mysql

`/etc/services:` store port-service mappings

Practice

Find the command listening on address 0.0.0.0:80 (port 80)

Additional resources

- Linux 101 Hacks