

basic Linux commands part 2

Nguyen Quang Khai

23/05/2024

Contents

I. Working with text file.

1. Search and filter.
2. Input/output redirection.

II. Some other commands

3. Download file.
4. File permission.
5. Compress and decompress.

1. Search and filter

Search and filter

find command: search for files or directories in a directory

Search for files and directories named "vui", with -name option:

```
khainguyen@khai:~$ find -name vui
./Documents/new/vui
./Documents/vui
khainguyen@khai:~$ _
```

Search for directories named "vui", with -type option, d means directory:

```
khainguyen@khai:~$ find -type d -name vui
./Documents/vui
khainguyen@khai:~$ _
```

Search for files named "vui", with -type option, f means file:

```
khainguyen@khai:~$ find -type f -name vui
./Documents/new/vui
khainguyen@khai:~$ _
```

don't remember the file name clearly, but there is "vu" at the beginning of the name:

```
khainguyen@khai:~$ find -type f -name vu*
./var/app/com.usebottles.bottles/data/bottles/templates/ea3c7b87-f330-4eca-88df-296521338473/drive_c/windows/system32/vulkan-1.dll
./var/app/com.usebottles.bottles/data/bottles/templates/ea3c7b87-f330-4eca-88df-296521338473/drive_c/windows/syswow64/vulkan-1.dll
./var/app/com.usebottles.bottles/data/bottles/bottles/Chat/drive_c/users/khainguyen/AppData/Local/Programs/Zalo/Zalo-23.10.1/vulkan-1.dll
```

* means zero or any characters

Search and filter

head command: prints the first few lines in the file

Default: prints the first 10 lines in the file:

```
khainguyen@khai:~$ cat view.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
khainguyen@khai:~$ head view.txt
1
2
3
4
5
6
7
8
9
10
khainguyen@khai:~$ _
```

print the first #number lines instead of the first 10

```
khainguyen@khai:~$ cat view.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
khainguyen@khai:~$ head -3 view.txt
1
2
3
khainguyen@khai:~$ _
```

Search and filter

tail command: prints the last few lines in the file

Default: prints the last 10 lines in the file:

```
khainguyen@khai:~$ cat view.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
khainguyen@khai:~$ tail view.txt
4
5
6
7
8
9
10
11
12
13
khainguyen@khai:~$ _
```

print the last #number lines instead of the last 10

```
khainguyen@khai:~$ cat view.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
khainguyen@khai:~$ tail -5 view.txt
9
10
11
12
13
khainguyen@khai:~$ _
```

Search and filter

wc command: print newline, word, and byte counts for each given file

Default:

```
khainguyen@khai:~$ cat view.txt
1 a
2
3
4
5
6
7
8
9
10
11
12
13
khainguyen@khai:~$ wc view.txt
13 14 32 view.txt
khainguyen@khai:~$ _
```

13 lines, 14 words, 32 bytes

select which counts are printed:

```
khainguyen@khai:~$ wc --lines view.txt
13 view.txt
khainguyen@khai:~$ _
```

```
khainguyen@khai:~$ wc --words view.txt
14 view.txt
khainguyen@khai:~$ _
```

```
khainguyen@khai:~$ wc --bytes view.txt
32 view.txt
khainguyen@khai:~$ _
```

Search and filter

cut command: Print selected parts of lines from each FILE

```
khainguyen@khai:~$ cat state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
khainguyen@khai:~$ _
```

Print the 3rd character, 5th to 7th characters of the file:

```
khainguyen@khai:~$ cut --characters 3,5-7 state.txt
dra
uach
sm
hr
hthi
khainguyen@khai:~$ _
```

Specify fields separated by a space, printing the 2nd field:

```
khainguyen@khai:~$ cut --delimiter " " --fields 2 state.txt
Pradesh
Pradesh
Assam
Bihar
Chhattisgarh
khainguyen@khai:~$ _
```

use **--only-delimited** option: do not print lines not containing delimiters

```
khainguyen@khai:~$ cut --delimiter " " --fields 2 --only-delimited state.txt
Pradesh
Pradesh
khainguyen@khai:~$ _
```

<i>cut --help:</i>	N	N'th byte, character or field, counted from 1	N-M	from N'th to M'th (included) byte, character or field
	N-	from N'th byte, character or field, to end of line	-M	from first to M'th (included) byte, character or field

Search and filter

grep command: print lines that match patterns

Default: case-sensitive

```
khainguyen@khai:~$ cat state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
khainguyen@khai:~$ grep "pra" state.txt
khainguyen@khai:~$ grep "Pra" state.txt
Andhra Pradesh
Arunachal Pradesh
khainguyen@khai:~$ _
```

--ignore-case option: case-insensitive

```
khainguyen@khai:~$ grep --ignore-case "pra" state.txt
Andhra Pradesh
Arunachal Pradesh
khainguyen@khai:~$ _
```

--line-number option: Display the matched lines and their line numbers.

```
khainguyen@khai:~$ grep --line-number --ignore-case "pra" state.txt
1:Andhra Pradesh
2:Arunachal Pradesh
khainguyen@khai:~$ _
```

--count option: print a count of matching lines

```
khainguyen@khai:~$ grep --count --ignore-case "pra" state.txt
2
khainguyen@khai:~$ _
```

--invert-match option: print non-matching lines

```
khainguyen@khai:~$ grep --count --invert-match --ignore-case "pra" state.txt
3
khainguyen@khai:~$
```

case-sensitive
abc \neq ABC

case-insensitive
abc = ABC

2. input/output redirection

input/output redirection

Piping

Pipe is formed using the vertical dash |

A | B | output of command A will be input of B command.

A | B | C output of command A will be input of B command, output of command B will be input of C command.

The pipe is used to combine two or more commands.

Example:

ls | grep "Do"

ls's output

grep's input

grep's output

```
khainguyen@khai:~$ ls
230Sam      ngs2amr      R
bin         Miniconda3-py312_24.1.2-0-Linux-x86_64.sh  r_ngs2amr
card        Music        shotgun
Desktop     ooo          snap
Documents   opt          Templates
Downloads   Pictures     Videos
ENTER       Public       view.txt
Exercise_1  qiime2-amplicon-2024.2-py38-linux-conda.yml
localDB     qiime2-import
khainguyen@khai:~$
```

```
khainguyen@khai:~$ ls | grep "Do"
Documents
Downloads
khainguyen@khai:~$
```

input/output redirection

Output redirection: The ">" character

To redirect output to a file

```
khainguyen@khai:~$ ls | grep "Do" > view.txt
khainguyen@khai:~$ cat view.txt
Documents
Downloads
khainguyen@khai:~$ _
```

If the view.txt file already exists, its content will be overwritten

```
khainguyen@khai:~$ cat view.txt
Picture
Videos
khainguyen@khai:~$ ls | grep "Do" > view.txt
khainguyen@khai:~$ cat view.txt
Documents
Downloads
khainguyen@khai:~$ _
```

input/output redirection

If you do not want the content to be overwritten, use the ">>" character

```
khainguyen@khai:~$ cat view.txt
Picture
Videos
khainguyen@khai:~$ ls | grep "Do" >> view.txt
khainguyen@khai:~$ cat view.txt
Picture
Videos
Documents
Downloads
khainguyen@khai:~$ _
```

Input redirection: The "<" character

```
khainguyen@khai:~$ grep -n "8" < view.txt
8:8
```

```
khainguyen@khai:~$ cat view.txt
1 a
2
3
4
5
6
7
8
9
10
11
12
13
```

3. Download file

Download file

wget command: Downloader

Default: Download file from a URL

```
khainguyen@khai:~/Documents$ wget http://linuxcommand.org/index.php
--2024-05-15 19:08:44-- http://linuxcommand.org/index.php
Resolving linuxcommand.org (linuxcommand.org)... 216.105.38.11
Connecting to linuxcommand.org (linuxcommand.org)|216.105.38.11|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3929 (3,8K) [text/html]
Saving to: 'index.php'

index.php                               100%[=====>]    3,84K  --.-KB/s   in 0s

2024-05-15 19:08:45 (542 MB/s) - 'index.php' saved [3929/3929]

khainguyen@khai:~/Documents$ ls
index.php
khainguyen@khai:~/Documents$ _
```

--output-document option: download file from URL, with a different name

```
khainguyen@khai:~/Documents$ wget --output-document "edit.php" http://linuxcommand.org/index.php
--2024-05-15 19:11:22-- http://linuxcommand.org/index.php
Resolving linuxcommand.org (linuxcommand.org)... 216.105.38.11
Connecting to linuxcommand.org (linuxcommand.org)|216.105.38.11|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3929 (3,8K) [text/html]
Saving to: 'edit.php'

edit.php                               100%[=====>]    3,84K  --.-KB/s   in 0s

2024-05-15 19:11:23 (492 MB/s) - 'edit.php' saved [3929/3929]

khainguyen@khai:~/Documents$ ls
edit.php
khainguyen@khai:~/Documents$ _
```

Download file

--input-file option: download from multiple URL

1. Create a text file containing URLs, each line includes a URL, use: nano.

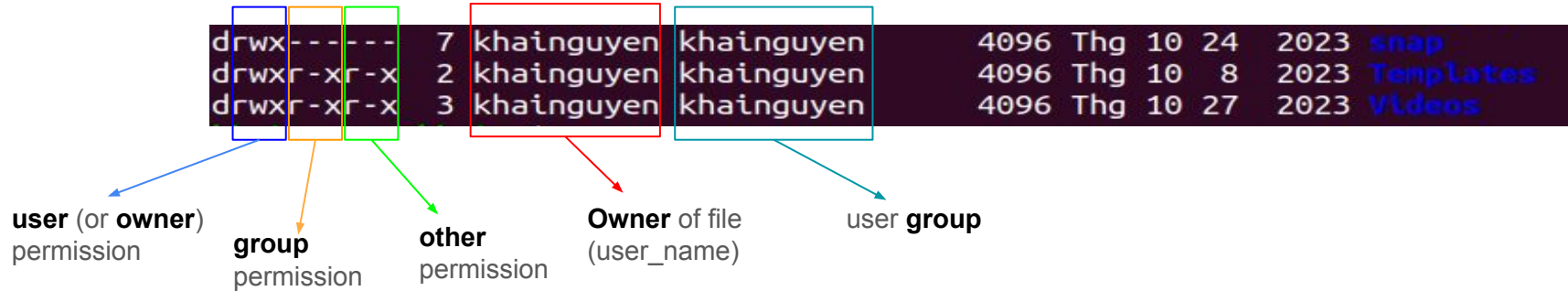
```
khainguyen@khai:~/Documents$ cat link_down.txt
http://linuxcommand.org/index.php
http://linuxcommand.org/lc3_resources.php
http://linuxcommand.org/tlcl.php
khainguyen@khai:~/Documents$ _
```

2. wget --input-file link_down.txt

```
khainguyen@khai:~/Documents$ wget --input-file link_down.txt
```


4. File permission

When run: ls -l



Meaning of r, w, x, -:

r = readable

w = writeable

x = executable

- = denied

snap directory

- User permission: read (yes), write (yes), execute (yes).
- Group permission: read (no), write (no), execute (no).
- Other permission: read (no), write (no), execute (no).

Templates directory

- User permission: read (yes), write (yes), execute (yes).
- Group permission: read (yes), write (no), execute (yes).
- Other permission: read (yes), write (no), execute (yes).

Permission Attributes

Attribute	Files	Directories
r	Allows a file to be opened and read.	Allows a directory's contents to be listed if the execute attribute is also set.
w	Allows a file to be written to or truncated; however, this attribute does not allow files to be renamed or deleted. The ability to delete or rename files is determined by directory attributes.	Allows files within a directory to be created, deleted, and renamed if the execute attribute is also set.
x	Allows a file to be treated as a program and executed. Program files written in scripting languages must also be set as readable to be executed.	Allows a directory to be entered, e.g., <i>cd directory</i> .

chmod: Change File Mode

To change the mode (permissions) of a file or directory, use the **chmod** command.

chmod supports **two distinct ways** of specifying mode changes:

1. Octal number representation

Octal	Binary	File mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwX

File Modes in Binary and Octal

chmod 755 z.txt

Diagram illustrating the breakdown of the octal mode 755:

- 7 (user):** rwx (read, write, execute)
- 5 (group):** r-x (read, execute)
- 5 (other):** r-x (read, execute)

```
khainguyen@khai:~/Documents/new$ ls -l
total 0
----- 1 khainguyen khainguyen 0 Thg 5 16 21:44 z.txt
khainguyen@khai:~/Documents/new$ chmod 755 z.txt
khainguyen@khai:~/Documents/new$ ls -l
total 0
-rwxr-xr-x 1 khainguyen khainguyen 0 Thg 5 16 21:44 z.txt
khainguyen@khai:~/Documents/new$ _
```

chmod: Change File Mode

2. Symbolic representation

Symbol	Meaning
u	Short for "user" but means the file or directory owner.
g	Group owner.
o	Short for "others" but means world.
a	Short for "all." This is a combination of u, g, and o.

chmod Symbolic Notation

If no character is specified, "all" will be assumed.

- + a permission is to be added.
- a permission is to be taken away.
- = only the specified permissions are to be applied and that all others are to be removed.

Permission:

- r readable
- w writeable
- x executable

```
khainguyen@khai:~/Documents/new$ ls -l
total 0
----- 1 khainguyen khainguyen 0 Thg 5  16 21:44 z.txt
khainguyen@khai:~/Documents/new$ chmod u+rx,o+xw z.txt
khainguyen@khai:~/Documents/new$ ls -l
total 0
-r-x---wx 1 khainguyen khainguyen 0 Thg 5  16 21:44 z.txt
```

```
-r-x---wx 1 khainguyen khainguyen 0 Thg 5  16 21:44 z.txt
khainguyen@khai:~/Documents/new$ chmod a=r z.txt
khainguyen@khai:~/Documents/new$ ls -l
total 0
-r--r--r-- 1 khainguyen khainguyen 0 Thg 5  16 21:44 z.txt
```

```
-r--r--r-- 1 khainguyen khainguyen 0 Thg 5  16 21:44 z.txt
khainguyen@khai:~/Documents/new$ chmod ug+wx z.txt
khainguyen@khai:~/Documents/new$ ls -l
total 0
-rwxrwxr-- 1 khainguyen khainguyen 0 Thg 5  16 21:44 z.txt
```

5. Compress and decompress

Compress and decompress

tar command: compress and decompress

Compress all files and directories in the "Documents" directory to a compressed file named "doc.tar.gz"

-c --create: create a new archive file

-z, --gzip: Using gzip to compress

```
khainguyen@khai:~/Documents$ ls
index.php  lc3_resources.php  link_down.txt  new  tlcl.php  vui
khainguyen@khai:~/Documents$ tar cvzf doc.tar.gz ../Documents/
tar: Removing leading `../' from member names
../Documents/
../Documents/link_down.txt
../Documents/lc3_resources.php
../Documents/index.php
../Documents/new/
../Documents/new/Ani.txt
../Documents/new/vui
../Documents/new/z.txt
../Documents/tlcl.php
../Documents/vui/
tar: ../Documents: file changed as we read it
khainguyen@khai:~/Documents$ ls
doc.tar.gz  index.php  lc3_resources.php  link_down.txt  new  tlcl.php  vui
khainguyen@khai:~/Documents$ _
```

-f or --file: compressed file name


-v or --verbose: Verbosely list files processed

The four options "-c -v -z -f" can be written as "cvzf"

Compress and decompress

Decompress a compressed file named "doc.tar.gz"

```
khainguyen@khai:~/Documents$ ls
doc.tar.gz index.php lc3_resources.php link_down.txt new tlcl.php vui
khainguyen@khai:~/Documents$ tar xvf doc.tar.gz
Documents/
Documents/link_down.txt
Documents/lc3_resources.php
Documents/index.php
Documents/new/
Documents/new/Ani.txt
Documents/new/vui
Documents/new/z.txt
Documents/tlcl.php
Documents/vui/
khainguyen@khai:~/Documents$ ls
doc.tar.gz index.php link_down.txt tlcl.php
Documents lc3_resources.php new vui
khainguyen@khai:~/Documents$ ls ./Documents/
index.php lc3_resources.php link_down.txt new tlcl.php vui
khainguyen@khai:~/Documents$
```



-x, --extract: Extract files from an archive.

Summary

find	search for files or directories in a directory
head	prints the first few lines in the file
tail	prints the last few lines in the file
wc	print newline, word, and byte counts for each given file
cut	Print selected parts of lines from each FILE
grep	print lines that match patterns

Summary

	pipe
">", ">>"	output redirection
"<"	input redirection
wget	download file from <link>
chmod	Change File Mode
tar	compress and decompress

Homeworks

1. Create a directory named "hw2".
2. Move to "hw2" directory, and download this file (SRR25653406.fasta.tar.gz) using wget command:
https://github.com/UeenHuynh/MGMA_2024/raw/main/lecture2/2.%20basic%20Linux%20commands%20part%202/SRR25653406.fasta.tar.gz
3. Decompressing this file SRR25653406.fasta.tar.gz using tar command.

Additional information about FASTA format (The file has the extension .fasta, .faa, .fna, etc.):

FASTA format is a text-based format, contains two lines:

- First line: is the comment (description) line.
 - + **Always starts with ">" sign (This information is useful for the exercise).**
 - + Give basic information about the sequence (nucleotide or amino acid).
- Second line: The actual sequence of the first line description, using a standard one-letter character string.

Example:

```
>M35309.1 E.coli 16S rRNA fragment
GGCATGAAGACACACTGCTAACTCCGAATACGCACAAGCCCGTAATGGAGCGACGGTGGGCCTTGTTCCC
GTGCCCCGATGTGGGGTGGAGGTGACTGTGGGTTGTGATATTCGGGGAGGCAAAAGAAGTAGCGAGTCTA
ACCTTGCTTACCACTTTGCCTAATACGGGAAACG
```

(<https://www.ncbi.nlm.nih.gov/nuccore/M35309.1?report=fasta>)

=> Let's call this single-sequence FASTA format (https://en.wikipedia.org/wiki/FASTA_format), so if a text file contains a single-sequence FASTA format, this file is called a single-sequence FASTA file.

If a text file contains two or more single-sequence FASTA format, this file is called a multiple-sequence FASTA file or multi-FASTA file.

Homeworks

An example of multi-FASTA format :



(https://www.researchgate.net/figure/A-sample-of-the-Multi-FASTA-file_fig1_309134977)

Additional information for this file: SRR25653406.fasta (after decompressing using tar command).

- This is a multi-FASTA file. It is converted from fastq to fasta format file (Don't worry about fastq format in this exercise), the original fastq format file is obtained from the sequencing machine.
- Each single-sequence FASTA in this file is information about **a read** - a piece of DNA sequence obtained from the sequencing machine:
 - + First line: some information about the sequencing platform of this file, that's all, don't worry about the details.
 - + Second line: DNA sequence of that **read**, a character string consisting of 4 types of letters: A, T, G, C representing 4 types of nucleotides. In addition, there is the letter N, which means 1 of 4 possible types of nucleotides, because the sequencing machine cannot identify which type of nucleotide it is.

Homeworks

A single-sequence FASTA in this file:

```
>SRR25653406.lite.3.8 M02606:41:000000000-J3RLF:1:1101:8131:2344 length=200  
TTTGGTCCAGCAGGCTATTGCTGGGAATTGTTAGCTTTTCGGGATCCATTTTATTGGGCGGTGTGCTCATCTCTACCATTTC AACATTATAGAGCGGCGTGTGG  
ATGTGGTAAACACGGGGCGAATGACTTACCGCAACATAACTCAACACTATGTATTGATAGGCTTCAATGA ACTAACTATCAATATGATACGCGAA
```

(The DNA sequence has 2 lines because the screen is not enough to display 1 line, so it is split into 2 lines)

4. How many lines does this file have? Using: wc.
5. How many reads does this file have? Using: grep for the search “^>” means the line starts with the character ">", and -c option.
6. How many reads **do not have any N**? Using: grep with -v, -c option; pipe.
7. Create a file named "id_read.txt" containing only the first line (the line contains information) of all reads. Using: grep with “^v”; output redirection.

Homeworks

8. Create a file named "part_of_id_read.txt" containing **a part of the first line** (the line contains information) of **all reads**. Using: grep; pipe; cut; output redirection.

Two lines of 1 read:

```
>SRR25653406.lite.3.8 M02606:41:000000000-J3RLF:1:1101:8131:2344 length=200
TTTGGTCCAGCAGGCTATTGCTGGGAATTGTTAGCTTTTCGGGATCCATTTTATTGGGCGGTGTGCTCATCTCTACCATTTC AACATTATAGAGCGGCGTGTGG
ATGTGGTAAACACGGGGCGAATGACTTACCGCAACATAACTCAACACTATGTATTGATAGGCTTCAATGAACTAACTATCAATATGATACGCGAA
```

(The line containing the DNA sequence has 2 lines because the screen is not enough to display 1 line, so it is split into 2 lines).

a part of the first line of 1 reads:

```
M02606:41:000000000-J3RLF:1:1101:8131:2344
```