Basic Commands and Combinations

Diptesh Kanojia
IITB-Monash Research Academy

e-Yantra Team
ERTS Lab, IIT Bombay

IIT Bombay
March 29, 2021

Agenda for Discussion

1. Commands
   - Basic Commands
   - Con'CAT'enating!
   - Word Count Dracula
   - 'Cut' it out!
2. Combinations
   - Global Regular Expression Print (grep)
   - Sorting Data
   - Output Redirection(s)

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

- **cat**: Used to display the contents of a file.

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

- **cat**: Used to display the contents of a file.
- **wc**: Used to count the words, lines, and character.

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

- **cat**: Used to display the contents of a file.
- **wc**: Used to count the words, lines, and character.
- **cut**: Used to manipulate content from a structured content file.

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

- **cat**: Used to display the contents of a file.
- **wc**: Used to count the words, lines, and character.
- **cut**: Used to manipulate content from a structured content file.
- **sort**: Used to display sorted content from file or any output.

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

- **cat**: Used to display the contents of a file.
- **wc**: Used to count the words, lines, and character.
- **cut**: Used to manipulate content from a structured content file.
- **sort**: Used to display sorted content from file or any output.
- **grep**: Used to find specific content from the displayed content.

Basic Commands

We discuss the basic commands which can be used on the Linux terminal. These commands will help you find content in files, edit them with the use of a single command *etc.* In this tutorial, we learn the following:

- **cat**: Used to display the contents of a file.
- **wc**: Used to count the words, lines, and character.
- **cut**: Used to manipulate content from a structured content file.
- **sort**: Used to display sorted content from file or any output.
- **grep**: Used to find specific content from the displayed content.
- **redirection**: Redirecting the file output from display to a file.

Shall we bell the 'CAT'?

- This command helps you display the data in a file: `cat filename`.
- You can use it to concatenate the output of two or more files on display (stdout): `cat file1 file2`.
- If you want line numbers prepended: `cat -n filename`.
- Basically, the general syntax is: cat [OPTIONS] [FILENAME].
- Also, try and find out what 'TAC' does!
- **Try it out on a text file!**

Bonus: A head and a tail

- The **head** command displays the top ten lines of the file on the terminal.
- Similarly, can you guess how many lines does the **tail** command display?
- But they can surely be customized!
- Try to execute:
  head -n4 /etc/passwd
- You can change the value of '**n**' and display lines.
- Use '**man**' and *find out what else* 'head' *and* 'tail' *can do!*

How many words does it take to light a bulb?

This command shall help you count words, lines and characters in a file. The most simple usage of 'wc' is: `wc` [FILENAME]. It should show you the number of lines, number of words, and number of characters in the file.

How many words does it take to light a bulb?

This command shall help you count words, lines and characters in a file. The most simple usage of 'wc' is: `wc` [FILENAME]. It should show you the number of lines, number of words, and number of characters in the file.

- It can count the number of lines in a file and show you using: `wc -l` [FILENAME]
- You can use it to count words by executing: `wc -w` [FILENAME]
- Similarly, for characters, execute: `wc -c` [FILENAME]
- Basically, the general syntax is: `wc` [OPTIONS] [FILENAME]

'Cut'ting data from files

From any output or file, which contains structured data, this command can be used to select and extract data based on fields/bytes/characters.

'Cut'ting data from files

From any output or file, which contains structured data, this command can be used to select and extract data based on fields/bytes/characters.

- A simple execution to list all the system users would be:

```
cut -d“:” -f1 /etc/passwd
```

'Cut'ting data from files

From any output or file, which contains structured data, this command can be used to select and extract data based on fields/bytes/characters.

- A simple execution to list all the system users would be:
  `cut -d":" -f1 /etc/passwd`
- The above command specifies a delimiter (-d":") ":" and then specifies that we want the data from field number one (-f1) from the file named "passwd" located in the "/etc/" location.

'Cut'ting data from files

From any output or file, which contains structured data, this command can be used to select and extract data based on fields/bytes/characters.

- A simple execution to list all the system users would be:
  `cut -d“:” -f1 /etc/passwd`
- The above command specifies a delimiter (-d“:”) “:” and then specifies that we want the data from field number one (-f1) from the file named “passwd” located in the “/etc/” location.
- For usernames, and group ids:
  `cut -d“:” -f1,4 /etc/passwd`

Have you met 'grep'?

'grep' can be used to select lines from the output, based on a string or a pattern.

Have you met 'grep'?

'grep' can be used to select lines from the output, based on a string or a pattern.

- The general syntax is:
  `grep` [STRING] [FILENAME].

Have you met ‘grep’?

‘grep’ can be used to select lines from the output, based on a string or a pattern.

- The general syntax is:
  `grep` [STRING] [FILENAME].
- To grab the complete line which contains your USERNAME from the ‘passwd’ file, use:
  `grep ‘USERNAME’ /etc/passwd`

Have you met ‘grep’?

‘grep’ can be used to select lines from the output, based on a string or a pattern.

- The general syntax is:
  `grep` [STRING] [FILENAME].
- To grab the complete line which contains your USERNAME from the ‘passwd’ file, use:
  `grep ‘USERNAME’ /etc/passwd`
- grep is used most effectively when combined with other commands like ‘cat’ or ‘ls’ with the use of a pipe (|). *E.g.*,
  `cat` [FILENAME] | `grep` [STRING]

Have you met ‘grep’?

‘grep’ can be used to select lines from the output, based on a string or a pattern.

- The general syntax is:
  `grep` [STRING] [FILENAME].
- To grab the complete line which contains your USERNAME from the ‘passwd’ file, use:
  `grep ‘USERNAME’ /etc/passwd`
- grep is used most effectively when combined with other commands like ‘cat’ or ‘ls’ with the use of a pipe (|). *E.g.*,
  `cat` [FILENAME] | `grep` [STRING]
- The above should print the lines from FILENAME which contain the STRING.

Have you met ‘grep’?

‘grep’ can be used to select lines from the output, based on a string or a pattern.

- The general syntax is:
  `grep` [STRING] [FILENAME].
- To grab the complete line which contains your USERNAME from the ‘passwd’ file, use:
  `grep ‘USERNAME’ /etc/passwd`
- grep is used most effectively when combined with other commands like ‘cat’ or ‘ls’ with the use of a pipe (|). *E.g.,*
  `cat` [FILENAME] | `grep` [STRING]
- The above should print the lines from FILENAME which contain the STRING.
- Try: `ls -l` | `grep` “Aug”.

Don your ‘sort’ing hat!

It sounds obvious what ‘`sort`’ command might be useful for, but in combination with other commands like ‘`cat`’ or ‘`ls`’ with the options ‘`sort`’ provides, it is a very powerful tool. Try the following:-

Don your ‘sort’ing hat!

It sounds obvious what ‘`sort`’ command might be useful for, but in combination with other commands like ‘`cat`’ or ‘`ls`’ with the options ‘`sort`’ provides, it is a very powerful tool. Try the following:-

- To sort folder and file wise: `ls -l | sort`

Don your 'sort'ing hat!

It sounds obvious what 'sort' command might be useful for, but in combination with other commands like 'cat' or 'ls' with the options 'sort' provides, it is a very powerful tool. Try the following:-

- To sort folder and file wise: `ls -l | sort`
- To sort the file list, based on size of the files: `ls -l | sort -k5n`

Don your ‘sort’ing hat!

It sounds obvious what ‘sort’ command might be useful for, but in combination with other commands like ‘cat’ or ‘ls’ with the options ‘sort’ provides, it is a very powerful tool. Try the following:-

- To sort folder and file wise: `ls -l | sort`
- To sort the file list, based on size of the files: `ls -l | sort -k5n`
- To sort the file output, use:
  `cat [FILENAME] | sort`

Don your 'sort'ing hat!

It sounds obvious what 'sort' command might be useful for, but in combination with other commands like 'cat' or 'ls' with the options 'sort' provides, it is a very powerful tool. Try the following:-

- To sort folder and file wise: `ls -l | sort`
- To sort the file list, based on size of the files: `ls -l | sort -k5n`
- To sort the file output, use:
  `cat [FILENAME] | sort`
- Can you use 'man' and find how to sort in inverse alphabetical order?

One Direction or Many!

A very simple use of output redirection is to combine two text files into one single file. Suppose you have FILE1 and FILE2, and want to create FILE3 with the contents of the files above, you can simply use ‘cat’ but with redirection.

- `cat FILE1 FILE2 > FILE3`

One Direction or Many!

A very simple use of output redirection is to combine two text files into one single file. Suppose you have FILE1 and FILE2, and want to create FILE3 with the contents of the files above, you can simply use ‘cat’ but with redirection.

- `cat FILE1 FILE2 > FILE3`
- This will create a new file FILE3 on the disk, and add the contents of FILE1 and FILE2 to it; but if a file already exists and you only want to append the content:
  `cat FILE1 FILE2 >> FILE3`

One Direction or Many!

A very simple use of output redirection is to combine two text files into one single file. Suppose you have FILE1 and FILE2, and want to create FILE3 with the contents of the files above, you can simply use 'cat' but with redirection.

- `cat FILE1 FILE2 > FILE3`
- This will create a new file FILE3 on the disk, and add the contents of FILE1 and FILE2 to it; but if a file already exists and you only want to append the content:
  `cat FILE1 FILE2 >> FILE3`
- You can redirect any command output to a file, not just 'cat'.

One Direction or Many!

A very simple use of output redirection is to combine two text files into one single file. Suppose you have FILE1 and FILE2, and want to create FILE3 with the contents of the files above, you can simply use 'cat' but with redirection.

- `cat FILE1 FILE2 > FILE3`
- This will create a new file FILE3 on the disk, and add the contents of FILE1 and FILE2 to it; but if a file already exists and you only want to append the content:
  `cat FILE1 FILE2 >> FILE3`
- You can redirect any command output to a file, not just 'cat'.
- Try creating a file which contains the list of all files in this directory. *Hint*: Redirect the output of '`ls`' command in a new file.

References

- TLDP: Basic Commands
- e-Yantra Homepage

Thank You!

Author: Diptesh Kanojia
Contributor: Prashant Sharma

Post your queries at: resources@e-yantra.org