

# grep Command in Linux

**Spoken Tutorial Project**  
**National Mission on Education through ICT**  
**<http://spoken-tutorial.org>**

**Script**  
**Anirban Choudhury & Sachin Patil**

**Narration**  
**Ashwini Patil**

**8 Dec 2014**



# Learning Objectives



# Learning Objectives

- **grep command**



# Learning Objectives

- ▶ **grep command**
- ▶ **with the help of some examples**



# System Requirements



# System Requirements

## ► GNU/Linux OS



# System Requirements

- ▶ **GNU/Linux OS**
- ▶ **GNU BASH v. 4.2.24**



# System Requirements

- ▶ **GNU/Linux OS**
- ▶ **GNU BASH v. 4.2.24**
- ▶ **GNU BASH version 4 or above is recommended for practise**





# Prerequisites



# Prerequisites

- ▶ You should know basics of Linux Terminal



# Prerequisites

- ▶ You should know basics of Linux Terminal
- ▶ For relevant tutorials please visit our website:



# Prerequisites

- ▶ You should know basics of Linux Terminal
- ▶ For relevant tutorials please visit our website:
- ▶ <http://spoken-tutorial.org>



# Regular Expressions



# Regular Expressions

- ▶ Regular expressions are pattern matching techniques



# Regular Expressions

- ▶ Regular expressions are pattern matching techniques
- ▶ When we have to find out whether a pattern exists in a line, paragraph or a file



# Regular Expressions

- For eg. If you want to search a phone number in the telephone directory





# Regular Expressions

- ▶ For eg. If you want to search a phone number in the telephone directory
- ▶ Or find a keyword in a paragraph or a line, we use **grep** command



# Introduction to grep



# Introduction to grep

- ▶ **grep** searches for one or more patterns in one or more line, paragraph or a file



# Introduction to grep

- ▶ **grep** searches for one or more patterns in one or more line, paragraph or a file
- ▶ If filename is not mentioned, **grep** searches for the patterns in the standard input



# Introduction to grep

- ▶ If filename is missing, **grep** searches for the patterns in the standard input



# Demonstration file

- ▶ I will demonstrate usage of **grep** using a demo file called **grepdemo.txt**



# Reason for missing records

- ▶ This is because **grep** searched for the pattern "computers" with small **c**



# Reason for missing records

- ▶ This is because **grep** searched for the pattern "computers" with small **c**
- ▶ **Whereas for "Zubin" the stream is "Computers" with capital C**





# Reason for missing records

- ▶ This is because **grep** searched for the pattern "computers" with small **c**
- ▶ Whereas for "Zubin" the stream is "Computers" with capital **C**
- ▶ The pattern matching is case-sensitive



# What grep can do

- ▶ **grep** lists only those lines that match a given pattern



# What grep can do

- ▶ **grep** lists only those lines that match a given pattern
- ▶ **We may do the reverse**



# What grep can do

- ▶ **grep** lists only those lines that match a given pattern
- ▶ We may do the reverse
- ▶ **grep** can list those lines that do not match a given pattern



# What grep can do

- ▶ **grep** lists only those lines that match a given pattern
- ▶ We may do the reverse
- ▶ **grep** can list those lines that do not match a given pattern
- ▶ Use the **-v** option



# Single and Multi-word patterns

- ▶ The patterns so far have been of **single words**



# Single and Multi-word patterns

- ▶ The patterns so far have been of **single words**
- ▶ We may have **multi word** patterns, as well



# Single and Multi-word patterns

- ▶ The patterns so far have been of **single words**
- ▶ We may have **multi word** patterns, as well
- ▶ **But the entire pattern must be within quotes**





# Summary

- ▶ To see the content of a file  
eg. `cat filename`
- ▶ To list the entries of a particular stream  
eg. `grep "computers" grepdemo.txt`
- ▶ To ignore cases  
eg. `grep -i "computers" grepdemo.txt`



# Summary

- ▶ Lines that do not match the pattern  
eg. `grep -iv "pass" grepdemo.txt`
- ▶ To list the line numbers with the entries  
eg. `grep -in "fail" grepdemo.txt`



# Summary

- ▶ To store the result in another file  
eg. `grep -iv "pass" grepdemo.txt > notpass.txt`
- ▶ To know the count  
eg. `grep -c "Fail" grepdemo.txt`



# Assignment

Explore some other commands like

- ▶ -E
- ▶ +
- ▶ ?



# About the Spoken Tutorial Project

- ▶ Watch the video available at [http://spoken-tutorial.org/What\\_is\\_a\\_Spoken\\_Tutorial](http://spoken-tutorial.org/What_is_a_Spoken_Tutorial)
- ▶ It summarises the Spoken Tutorial project



# About the Spoken Tutorial Project

- ▶ Watch the video available at [http://spoken-tutorial.org/What\\_is\\_a\\_Spoken\\_Tutorial](http://spoken-tutorial.org/What_is_a_Spoken_Tutorial)
- ▶ It summarises the Spoken Tutorial project
- ▶ If you do not have good bandwidth, you can download and watch it



# Spoken Tutorial Workshops

## The Spoken Tutorial Project Team

- ▶ Conducts workshops using spoken tutorials
- ▶ Gives certificates to those who pass an online test
- ▶ For more details, please write to [contact@spoken-tutorial.org](mailto:contact@spoken-tutorial.org)



# Acknowledgements

- ▶ Spoken Tutorial Project is a part of the Talk to a Teacher project
- ▶ It is supported by the National Mission on Education through ICT, MHRD, Government of India
- ▶ More information on this Mission is available at:

<http://spoken-tutorial.org/NMEICT-Intro>

