Running Basic Text Handling Commands

Sadequl Hussain
INFORMATION TECHNOLOGIST, DIGITAL CONTENT PRODUCER

@sadequlhussain

Module Overview

Text processing examples

- Sorting
- Searching and replacing
- Extracting lines
- Comparing files

Module goal

- Basic Linux text handling commands
- Frequently used for text processing

Comparing Files with diff

Linux diff Command

“diff” stands for difference

- Compares two or more files line by line
- Reports the difference

Example 1: app configuration file

- File with latest timestamp not necessarily the correct one
- Use diff to find the difference

Example 2: shell script file

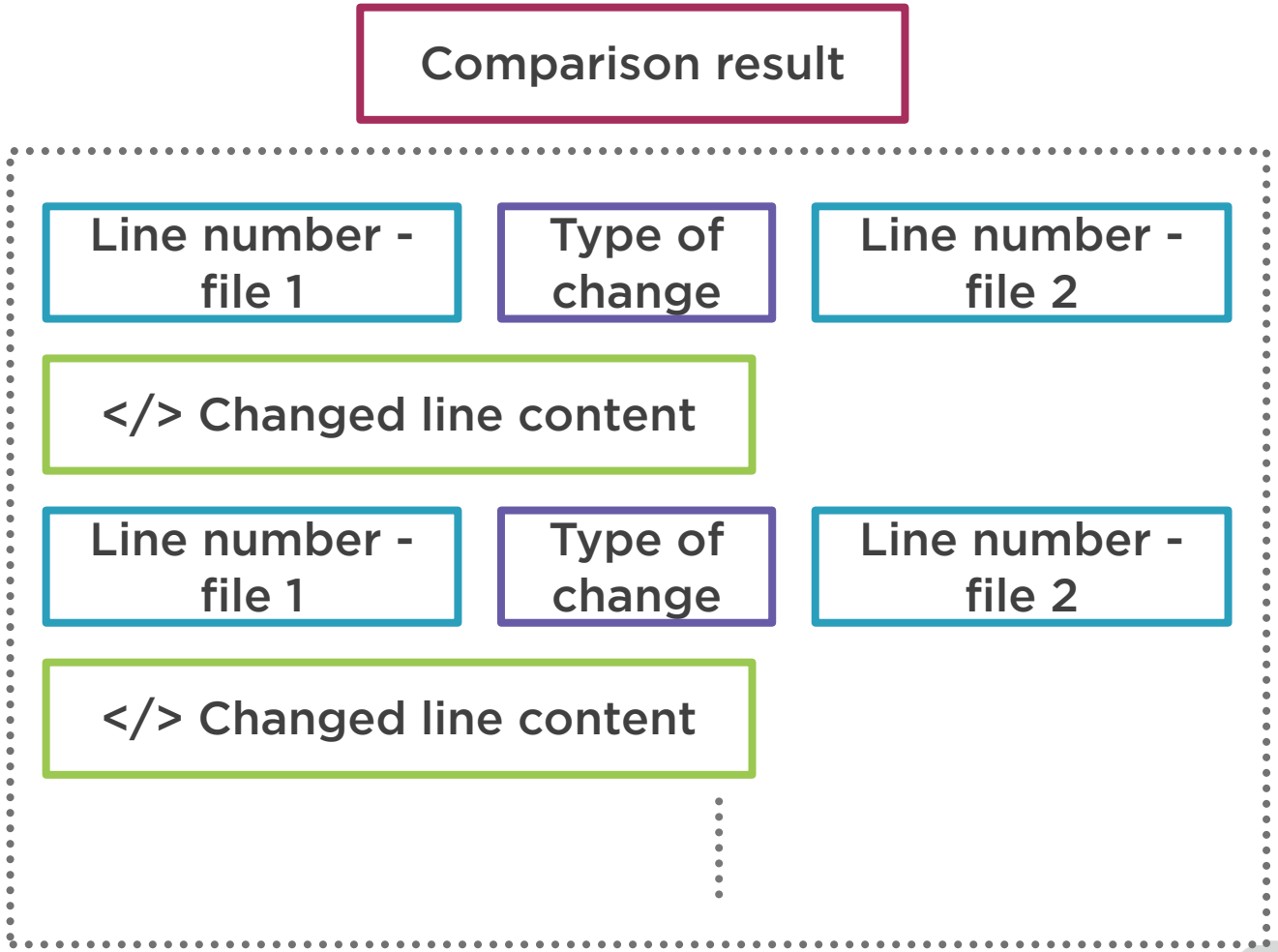
- Use diff to compare with previous version

```
diff <option_1> <option_2> … <option_n> <file_1> … <file_n>
```

Basic diff Syntax

Numerous options for diff. We will consider comparing two files

How diff Compares Files



Command:

```
diff file1 file2
```

Output:

```
file1 line# a/c/d file2 line#

</> Changed line's content
```

- Line number from first file, followed by “a”, “c”, or “d”, followed by line number from second file

- “<” means: this line needs to be removed or changed in the first file

  “>” means: this line needs to be added to the first file

```
7a8
> Contents of line 8 from second file
1d0
< Contents of line 1 from first file
7c7
< Contents of line 7 from first file
> Contents of line 7 from second file
```

- In the first file, *after* line 7, *add* the contents of line 8 from the second file.

  The contents of line 8 from the second file is shown after the “>” symbol

- In the first file, *delete* line 1, so both files can start with a common line

  The content of line 1 from the first file is shown after the “<” symbol

- *Change* line 7 of the first file to make it the same as line 7 of the second file

- The contents of line 7 from the first file is shown after the “<” symbol

- The contents of line 7 from the second file is shown after the “>” symbol

Demo

Mainly two files used for the demo

- stationery1
- stationery2

Other two files used

- stationery3
- stationery1_whitespaces

All four files included with exercise files

Demo run on Ubuntu server's Z shell

Counting with wc

wc is Word Count

Counts the number of words, lines and characters in a text file, or shows its size in bytes. Can be used to quickly analyze a file.

Sample Data File

Sample

Data file “rivers” included with exercise files

Data

Contains names of well-known rivers and their lengths in km

Location

Copy it under /tmp/data directory

Line Numbering with nl

Linux nl Command

Displays file content with line numbers

Without file name, displays standard input value with line number

Can help identify line breaks in files

Can be useful for checking specific lines in program code for errors

Changing Text with tr

Linux tr Command

“tr” stands for translate

- Used for translating or deleting characters
- Does not translate between languages
- Transforms or formats text characters

Ubuntu Zsh

- Sample file: “country_capitals.txt”

tr SETs

SETs are groups of characters that tell tr what to translate and how to translate it.

[‘upper:] and [:lower:] are two examples of SETs.

Ordering Text with sort

Linux Sort Command

Sort text files

Both ascending and descending order

Text and number

sort can order both types of data

Columnar text files

Can sort by a specific column

Sort Order

Line starting with number appears before line starting with letter

Line starting with lowercase letter precedes line starting with uppercase letter

sort concatenates multiple file inputs before sorting

Sample File

Commands shown in Z shell of Ubuntu server

Sample file: country_capitals.csv

File created in the section for “tr” command (under /tmp/data)

Power of sort

Basic

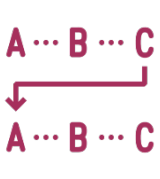
Can be used for basic text processing

Advanced

Can be used with other commands for advanced functionality

Finding Uniqueness with uniq

Linux uniq Command



Used to report or remove repeated lines of text

Useful for cleansing data files before processing

Comes with a number of options

Sample file “fruits.txt” included with exercise files

Remove Non-adjacent Repeats

Non-adjacent repeats

uniq cannot remove
non-adjacent repeated values

Workaround

Sort the text and remove
repeated values with uniq

Other uniq Options

Scenario: file with multiple fields

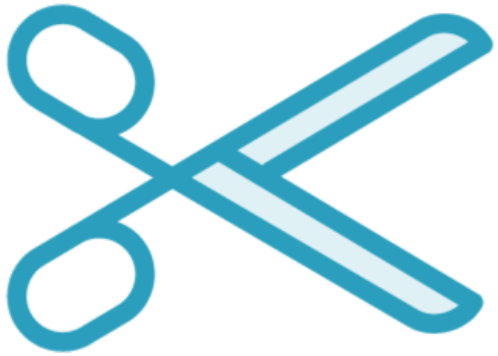
- Ignore number of columns from beginning
- Use –f switch

Scenario: Ignore number of characters from beginning of each line

- Use –s switch with number of characters
  - Useful for parsing time-stamped lines
- Use –c switch for number of repeats
- Pipe to “grep” for searching text

Extracting Text with cut

Linux cut Command



cut

Used to extract selected parts of lines from one or more files

CSV file

Created before: /tmp/data/country_capitals.csv

Files with non-columnar text content

- Example: header or footer rows

Default behavior of cut

- Extract text from all lines
- Result may not look nice

Exclude non-delimited text content

- Use “s” option with cut

Merging Lines with paste

Linux paste Command

paste

Used to merge lines from multiple files

Not *that* paste

Not like the “paste” we know in Word

Handy little tool

Can create new dataset from input files

cat

Can also add up multiple file contents

But paste

Merges lines vertically or horizontally

Sample Files

Dataset 1

states

Dataset 2

capitals

Joining Lines with join

Linux join Command

Special way of merging lines from two files

Similar to SQL join command

Lines from two files merged based on a common field

Both files must be sorted by the common field

Summary

Learning recap

- Finding file differences
- Counting characters, words, lines
- Numbering lines
- Transforming text
- Sorting text
- Finding unique values
- Extracting file fields
- Merging files
- Joining files

Summary

Processing text or input stream data

- Commands may need to be chained with pipes and redirects

Basic commands lack advanced features

- Conditional processing
- Formatting
- Looping
- Inserting lines

Text processing workflow may need these and other advanced features

- Next module covers sed and awk