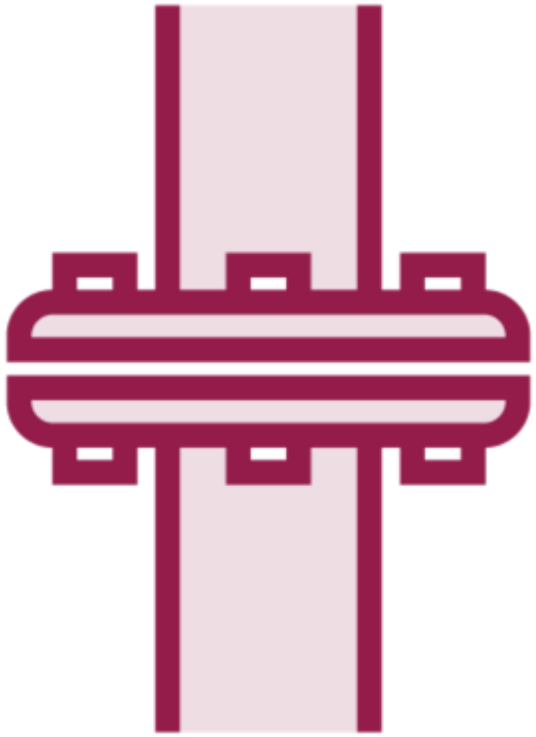
Reading and Writing to Text Files

Sadequl Hussain
INFORMATION TECHNOLOGIST, DIGITAL CONTENT PRODUCER

@sadequlhussain



Running text processing commands manually

- Pipe one command output to next
- Some advanced processing possible

Advanced text processing needs

- Dynamic in nature
- Piping commands together not enough
- Input text will change for each case

Automation necessary

- Saves from running same series of commands

Shell Scripts

Text file with a series of Linux commands

Together, commands achieve a workflow

An executable file, like other Linux commands

Can accept one or more parameters

Similar to batch files or PowerShell scripts

Problem Statement

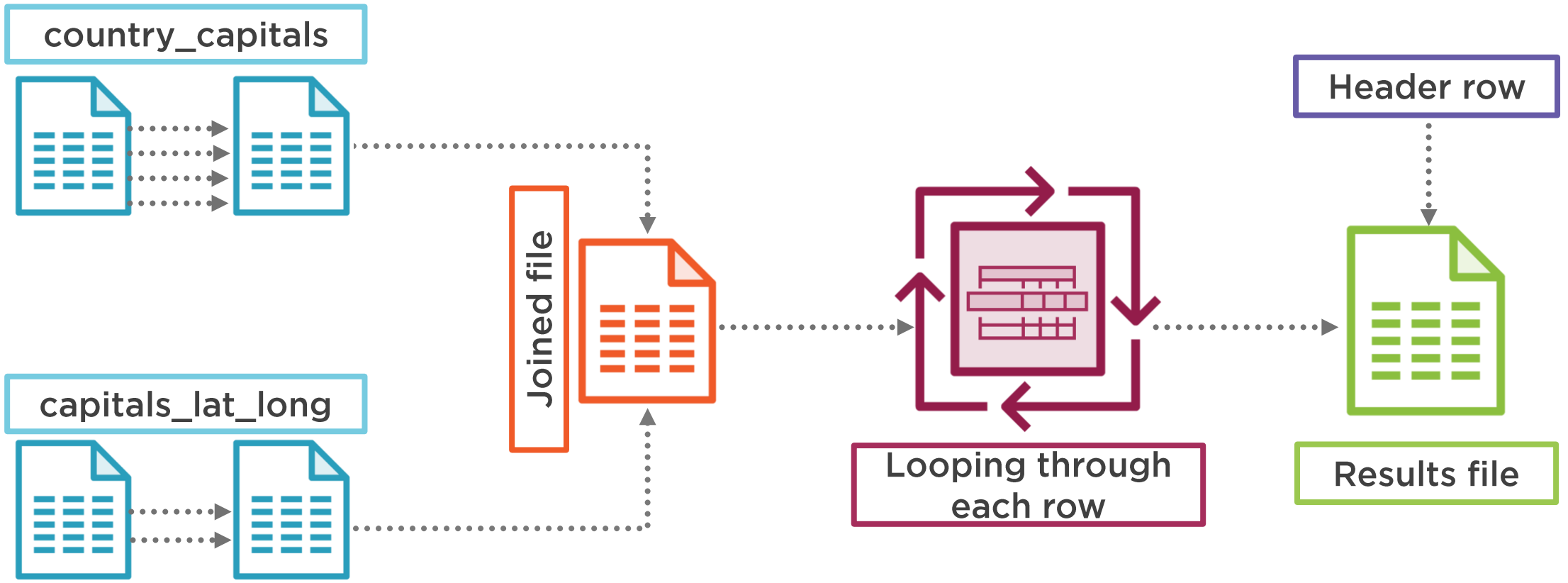
For a given continent, extract the names of all countries listed under that continent, their capital cities, and the latitude, longitude of those capital cities.

Example

When querying for Europe, create a file with five columns: continent name (Europe), countries listed under Europe, capital city of each country and the latitude, longitude of each capital city.

File should have a header row, showing column names.

High-level Workflow



A Quick Introduction to Shell Scripts

A Linux Shell Script

A file with a series of Linux commands

No structure, but needs to start with a pointer to the shell executable

Example script uses Bash shell, so need to point to the bash interpreter

First executable line of script starts with a #!. Example: #!/bin/sh

For Z shell, it can be pointing to the Z shell interpreter: #!/bin/zsh

```
#!/bin/sh

# This is a comment, it does not run

variable_name=<some_value>


echo $varaiable_name


if [[ <condition(s)> ]]; then
    <code block>
fi

while <condition>
do
    <code block>
done

echo “<Some message>”

some_command >> <some_file>

touch  <file_name>
```

- Comments start with hash sign (pound)
- A variable stores value in memory. Use equal sign to assign a value to a variable
- Refer to variable names starting with $
- An “if” block runs a block of commands based on one or more conditions
- A “while” loop runs a block of commands repeatedly based on a condition
- The “echo” command displays messages
- “>” overwrites a file, “>>” appends to it
- “touch” creates an empty file

A Text Processing Shell Script Walkthrough

Sample Shell Script

File name

process_data.sh
(name can be anything)

File extension

Typically based on shell
(e.g. .sh, .zsh, .csh)

```
#!/bin/sh

# Get the continent name from the argument passed t
# If the continent name is not provided, throw an e
continent="$@"
if [[ $continent == "" ]]; then
    echo "Continent name not provided!"
    exit
fi

# Check the existence of the source files in currer
# If the files don't exist, throw an error
country_capitals_filename="country_capitals.txt"
capitals_lat_long_filename="capitals_lat_long.txt"

if [[ ! -f ./"$country_capitals_filename" ]]; then
    echo "Source file for countries and their capit
    exit
fi

if [[ ! -f ./"$capitals_lat_long_filename" ]]; ther
    echo "Source file for capitals and their locati
    exit
fi

#Set some intermediary file names
country_capitals_filename_csv="country_capitals.csv
capitals_lat_long_filename_csv="capitals_lat_long.c
joined_filename="country_capitals_lat_long.csv"

# Process the source files into CSV format
# # Get rid of empty lines, capitalize all entries
# # Use comma separator
cat \
  "$country_capitals_filename" | \
  tr [:lower:] [:upper:] | \
  tr \\t \, | \
```

Script written for RHEL Bash shell

- Tested in RHEL Bash
- May need tweaking for running in Z shell

Creates a data file based on two input files

- country_capitals.txt
- capitals_lat_long.txt

Accepts one argument

- Continent name
- Can have spaces in it
- Can be in single or double quotes
- Can be in upper or lowercase

Course Summary

Congratulations!

If you have come this far, you have finished the last module and completed the course.

Well done!

Course Summary

Skills achieved

- Searching for text files
- Reading text files and command outputs in different ways
- Searching for text in files and input streams
- Comparing text files and directories
- Transforming text in different ways
- Filtering text from files and command outputs
- Reading from and writing to text files programmatically

Next Steps

Try the learning checks

Practice the commands learnt

Check out the “man” pages

Think how to apply the skills

Further Learning

Try advanced Linux courses from Pluralsight

- Linux system administration
- Advanced shell scripting
- Linux security
- Linux system programming
- Linux performance troubleshooting

This course will provide the basics for all

Hope you had a great experience

Feel free to rate it or leave feedback

Ask questions or make suggestions

Thank you and see you again!