Data Processing on the Command Line

Prof. Dr. Peter Braun

21. Oktober 2024

1 Titlepage

Data Processing on the Command Line

Prof. Dr. Peter Braun

thws Technical University of Applied Sciences
Würzburg-Schweinfurt

No Text – maybe some music :-)

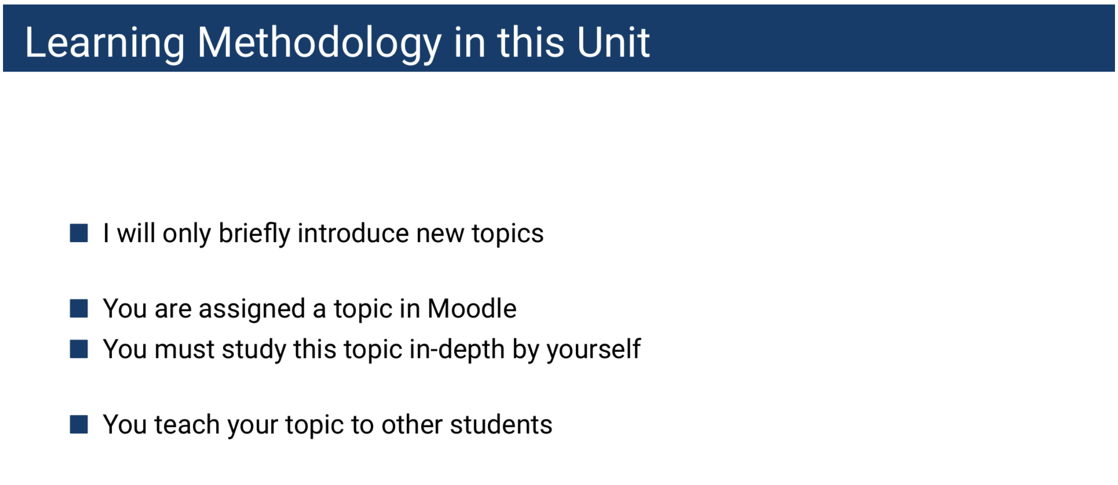
2 Learning Goals

Learning Goals

- You know basic commands to work with text files
- You know the Linux stream editor `sed`
- You know the Linux tool `awk`
- You know how to process JSON with `jq`
- You know how to modify audio/video with `ffmpeg`

Welcome to this unit, in which we will study more Linux command-line tools that are particularly applicable to data processing. In the last videos, you learned many Linux commands to organize your daily work on the console. We mentioned that Linux commands follow a basic philosophy: do one thing and do it right. To solve complex tasks, combining many Linux commands as a pipeline is necessary, passing data from one command to the next or using shell scripts, where you have all the freedom of a programming language. We also mentioned that text files are used very often because they are easy to read for humans, easy to edit, and easy to modify using Linux commands. It is possible to create generic tools if data is available as text. Today, we will look at some specific commands for text manipulation or modification. Well, to be honest, the last tool we will discuss is not for text but for audio and video data. Before we start, I want to explain today's learning methodology, because it's different compared to what you have done in the last videos.

3 Learning Methodology in this Unit

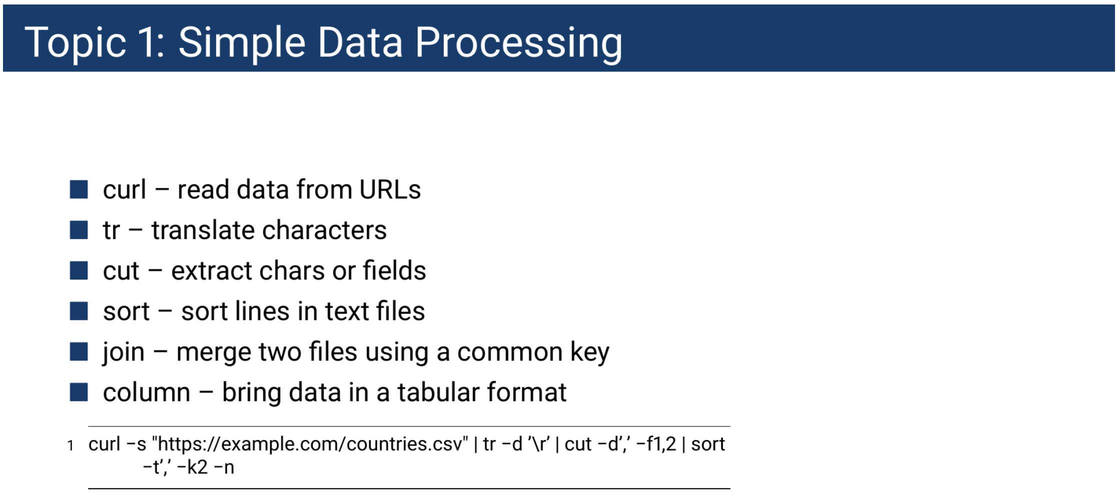


Learning Methodology in this Unit

- I will only briefly introduce new topics
- You are assigned a topic in Moodle
- You must study this topic in-depth by yourself
- You teach your topic to other students

In this unit, I will give only very brief input about the tools we will study. The main workload for studying this unit is on our side. In this short video, I will only talk about the motivation or the purpose of new commands, and your task is to study one of these commands or set of commands in detail. The unit covers five topics, and you are randomly assigned one of these five topics in the Moodle course. After you finish this video, go to Moodle and check your assigned topic. Then, use the literature that is provided at the end of this unit or in the Moodle course and study your subject in detail. You will also find a generic work instruction document in the Moodle course. It is your task to be prepared for the next meeting on campus. During the next meeting, you will then share your knowledge with your fellow students and learn from the other students about the other four topics. This peer teaching fortifies your understanding and aids in the collaborative learning process. By explaining the concept simplistically, you refine your understanding while helping others to understand the topic better.

4 Simple Data Processing



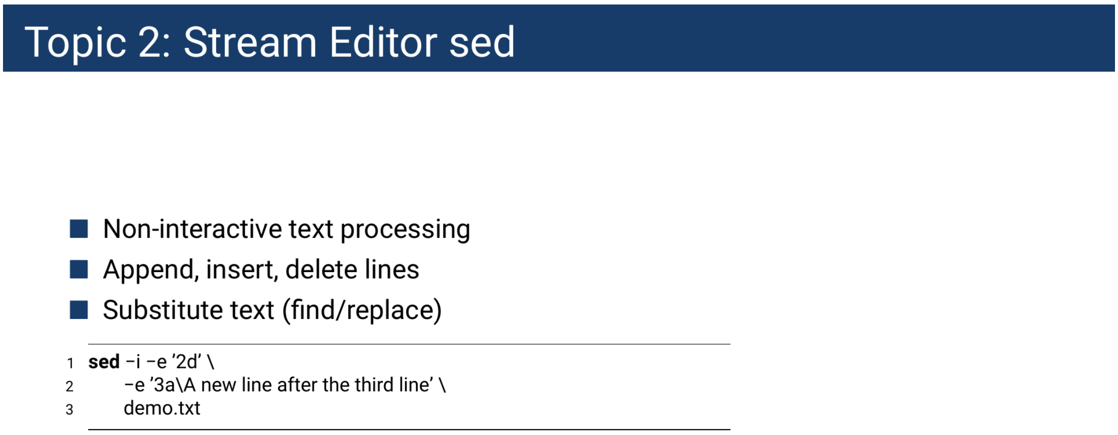
Topic 1: Simple Data Processing

- curl – read data from URLs
- tr – translate characters
- cut – extract chars or fields
- sort – sort lines in text files
- join – merge two files using a common key
- column – bring data in a tabular format

```
curl -s "https://example.com/countries.csv" | tr -d ’\r’ | cut -d’,’ -f1,2 | sort -t’,’ -k2 -n
```

The first topic is about a set of very simple tools for data processing. The tool “curl” is for loading data from a URL. The command ’tr’ stands for ’translate characters.’ This tool is used for translating or deleting characters. It allows us to change data formatting or remove unwanted characters, a common need when processing raw datasets. Next up, we have ’cut.’ The ’cut’ tool is particularly useful because it enables us to extract specific characters or fields. Fields in the meaning of columns of a CSV file. Moving on, ’sort.’ The ’sort’ command does exactly what it says and is particularly helpful when dealing with large text files. ’sort’ takes lines of text and orders them in a particular way – alphabetically, numerically, and so on. ’Join’ is a powerful command that merges two files using a common key. Let’s say you have two CSV files, and you want to combine them. Then we have ’column.’ This command tool can bring data into a tabular format, another frequent necessity when dealing with structured data. It ensures our raw data can be easily read and understood during analysis. You can see an example at the bottom of the slide beside me: Data is loaded from the given URL. It’s a CSV file with some country information like name, population, area, and so on. We delete unnecessary characters using “tr,” then cut out only the necessary columns, and finally sort the data numerically by the values in the second column.

5 Stream Editor sed



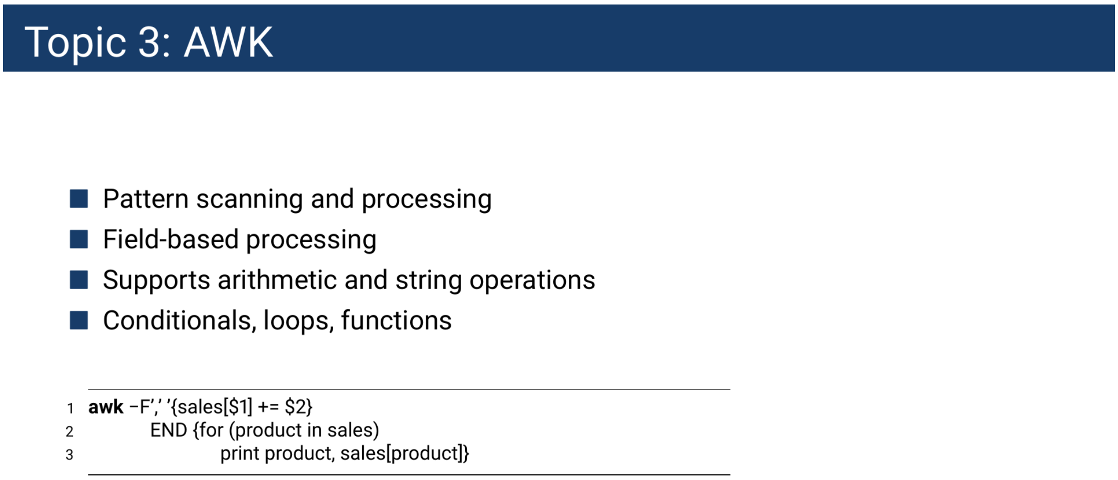
Topic 2: Stream Editor sed

- Non-interactive text processing
- Append, insert, delete lines
- Substitute text (find/replace)

```
sed −i −e '2d' \
    −e '3a\A new line after the third line' \
    demo.txt
```

The next topic is about the stream editor “sed.” To put it simply, “sed” is an editor for non-interactive text editing. This entails modifying text files without direct human intervention. A script or program will perform functions such as inserting, appending, or deleting lines, which can be invaluable for manipulating large datasets efficiently. Say you have a CSV file, and there are lines you’d like to append, insert, or delete. This is completely feasible using commands on the command line. For instance, to append a line is to add one at the end of the file. Meanwhile, to insert a line means to add a line at a specific location within the file. This position is found by a regular expression, matching the line before or after. This editor provides functions to substitute text, often called ’find and replace.’ These functions can search a text file for a specific string and replace it with a different string. Imagine having to manually search a document with thousands of lines for a specific text and replace it—an incredibly tedious task. But with the power of the command line, you can automate this process, saving you time and ensuring accuracy. Sed does not provide more commands than a normal editor such as VIM, but using sed, you can write scripts to execute the same text modifications again and again. It helps you to automate repetitive tasks. Look at the short example beside me: sed is called with two commands. The first one deletes the first two lines of the document. The second command inserts a new line after the third line. All commands are executed on a text file named “demo.txt.”

6 AWK



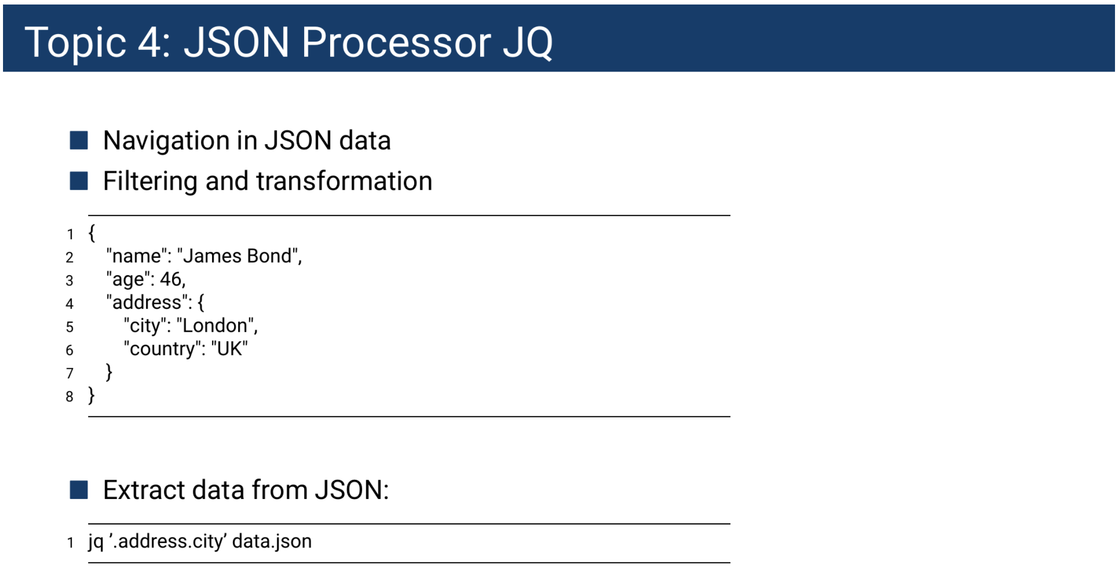
Topic 3: AWK

- Pattern scanning and processing
- Field-based processing
- Supports arithmetic and string operations
- Conditionals, loops, functions

```
1 awk −F’,’ ’{sales[$1] += $2}
2         END {for (product in sales)
3                   print product, sales[product]}
```

The next tool, “awk, ïs similar to the last tool, “sed,” but has a different focus. AWK is a powerful tool for pattern scanning and processing. This tool allows you to work through vast amounts of data, find matching patterns, and then manage or manipulate those matches according to your specifications. AWK is particularly useful in field-based processing, such as parsing and manipulating tabular data. AWK can understand and operate on it, whether comma-separated data from a CSV file or log-file details separated by spaces. For instance, you may use AWK to sum all numbers in a specific column of a multi-column file, which would be time-consuming if done manually. Additionally, AWK supports arithmetic and string operations, making it flexible and adaptable to your data processing needs. Use AWK to perform arithmetic transformations, concatenate strings, or convert strings to numbers. These operations are applicable at the field, record, or file level, evidencing AWK’s flexibility. More complex operations are also possible with AWK through conditionals, loops, and functions, just like in a regular programming language. You can use AWK for complex tasks beyond simple scanning and transformations. For example, you could use AWK’s conditional structures to filter and process specific data entries based on sophisticated criteria. Look at the example beside me: Input to this program is a CSV file with products and sales information. In the first line, an array named “sales” is created where the product names are used as a key, and we sum up sales information per product. In lines 2 and 3, the result of this analysis is printed to the console.

7 JSON Processor JQ



Topic 4: JSON Processor JQ

- Navigation in JSON data
- Filtering and transformation

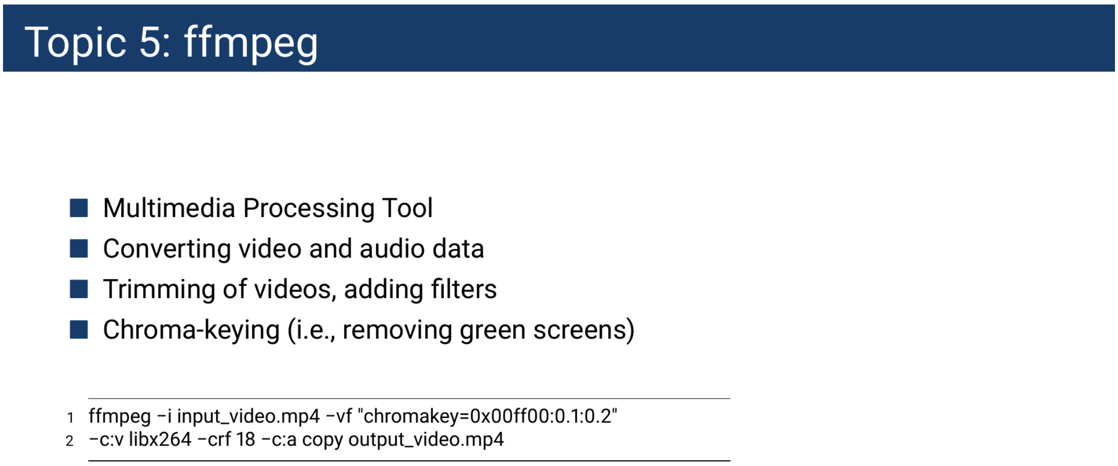
```
{
  "name": "James Bond",
  "age": 46,
  "address": {
    "city": "London",
    "country": "UK"
  }
}
```

- Extract data from JSON:

```
jq ’.address.city’ data.json
```

The fourth tool is for JSON processing. The tool is named “JQ.” Handling JSON data effectively is key in data processing on the command line, and JQ is a flexible assistant. You can compare JQ to SQL for JSON data. The query language is easy to understand for easy queries, as you can see in the example beside me. The JSON data contains data about a person with an embedded object for the address. The query below describes the path from the root to the data element “city,” which would result from this query. The JQ query language becomes a bit cryptic when queries become more complex. JQ has an extensive library of functions and operators, which lets you filter by conditions, reducing the JSON to the data you need. Moreover, JQ is not limited to just querying the data; it can transform it as required. This includes performing operations, changing formats, and modifying value – transforming the JSON data into a more convenient or functional shape for subsequent processing.

8 ffmpeg



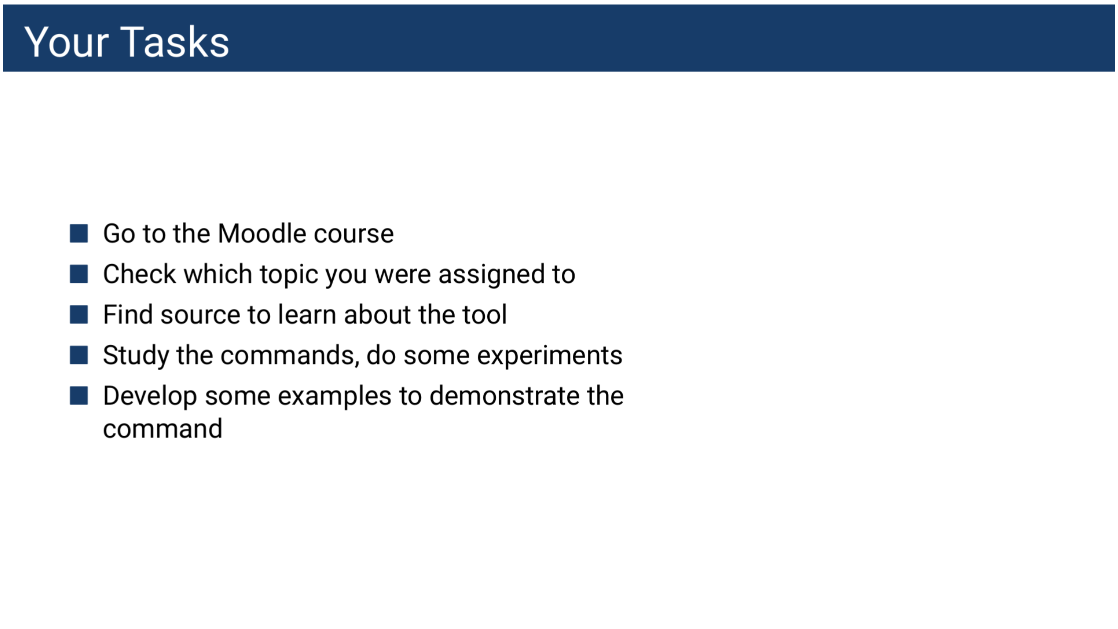
Topic 5: ffmpeg

- Multimedia Processing Tool
- Converting video and audio data
- Trimming of videos, adding filters
- Chroma-keying (i.e., removing green screens)

```
ffmpeg -i input_video.mp4 -vf "chromakey=0x00ff00:0.1:0.2"
-c:v libx264 -crf 18 -c:a copy output_video.mp4
```

The last tool we want to study in this unit is used for media data, in particular audio and video data. The name of the tool is “ffmpeg.” As its name subtly suggests, it is a fast-forward solution to manipulate video and audio data. The simplest task would be to convert videos from one format to another. Or to change the resolution or the number of frames per second. A more complex task is, for example, to chroma-key a video, which means removing a green screen from a recording and replacing it with a background image. The command you can see on the slide beside me is for removing the green screen, but it does not show how to put an image in the background. With ffmpeg, you can add text to your video and do all kinds of video filtering tasks. All the videos in my learning videos are produced by “ffmpeg.” I don’t edit my videos manually, but I have a set of scripts for all video editing tasks.

9 Your Tasks

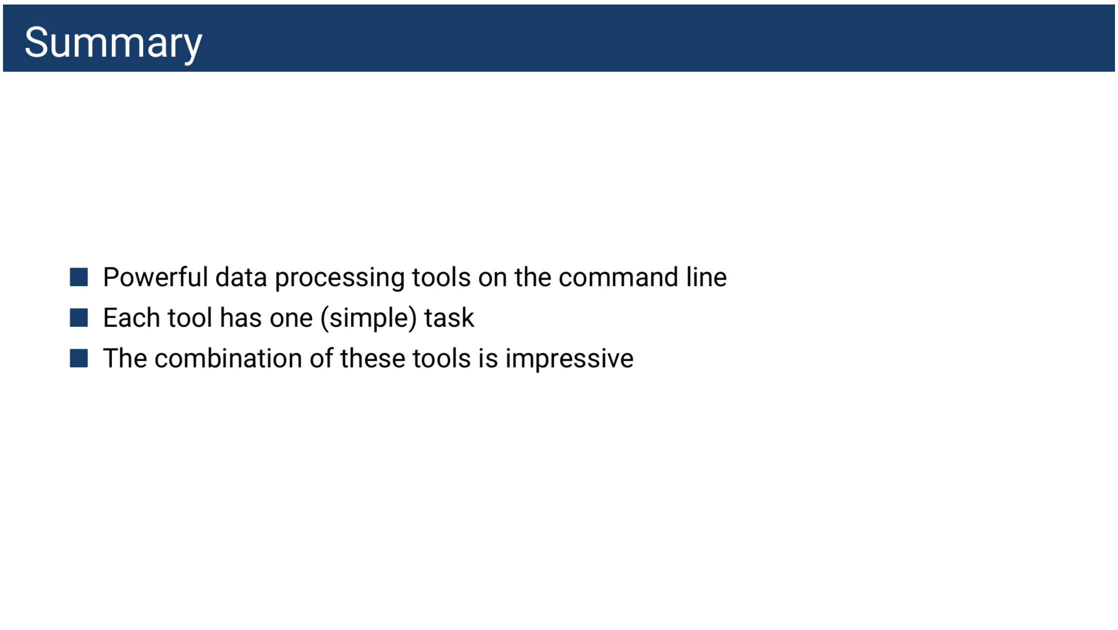


Your Tasks

- Go to the Moodle course
- Check which topic you were assigned to
- Find source to learn about the tool
- Study the commands, do some experiments
- Develop some examples to demonstrate the command

Now that you have the first idea about all topics, it's your task to study one of the topics in detail. Here's how you can approach this task effectively. Begin by visiting the Moodle course. Here, you'll find detailed information about the tool you must focus on, including your assigned topic. Remember, each student may get a different topic depending on the distribution done by Moodle. Once you know your specific topic, your next step is to identify and gather sources that will help you grasp the functionality of your assigned tool. Make sure the source is credible. You can use websites, online courses, tutorials, or even books—select a medium that suits your learning style better. With the right study material in place, delve into the specifics. Start by studying the commands. Understand what each command does and how it influences the data processing. It's advised also to do some experiments. Execution helps in understanding. Run commands, generate output, and check errors. This hands-on experience is invaluable in making you comfortable with the tools. After familiarizing yourself with the command set, think about how to demonstrate these commands. You need to develop a few examples. These examples should clearly illustrate the use and effect of each command. Framework a story around it: Data processing is not just about executing commands; it's more about solving problems. Please note that you do not need to prepare slides for the next class. You might want to connect to other students working on the same topic.

10 Summary



Summary

- Powerful data processing tools on the command line
- Each tool has one (simple) task
- The combination of these tools is impressive

That’s it for today. That was a short unit because the main workload this week is up to you. I introduced you to some very powerful command line tools for text and media processing. Again, a Linux command has one job, and only the combination of many tools models a complex workflow. But, honestly, tools like “sed” or “awk” are more like a Swiss army knife and useful for many tasks. Using the tools is quite cryptic in the beginning. But remember, Linux command line tools are used to automate your workflow. It’s worth spending more time learning how to automate than doing manual work over and over again.