MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NTU "Dnipro Polytechnic"

Institute of Electric Power Engineering

Faculty of Information Technology

Department of PZKS

REPORT

from laboratory work #2

disciplines "Development of microservice systems in the Golang language"

Performed by: Art. c. 122-20-04

Shcherbatyuk Artem Oleksiyovych

Checked by: Assoc. Reuta O.V

Topic: Go routines and channels

Goal: 1) Learn how to use goroutines

2) Learn to work with channels

Task 2

"CSV Concurrent Sorter" is a CLI application that allows sorting of its input presented as CSV-text.

Technical details

Using the "CSV Sorter" from the Task 1, extend it with the following required features:

1. The application has additional option **-d dir-name** that specifies a directory where it must read input files from. All files in the directory must have the same format. The output stays the same, it is a one file with sorted content from all input files.
2. Processing must be implemented concurrently based on pipeline.

The pipeline includes two stages:

- Reading - read input and sent it line by line further.
- Sorting - add received lines into the Tree.

3. The application outputs the result when the input ends up.
4. The project includes Unit tests covering the unit that builds the Tree.

Optional features (not required but appreciated):

1. Add signal processing that allows to gracefully stop the application when the user interrupts it pressing Ctrl-C. The interrupted application must write the current result.
2. If your application supports two types of algorithms, include to the project benchmarks comparing usages of these algorithms.

Program code:

Link to the GitHub repository:

https://github.com/kaaamich/GoLang/tree/main/2_Laba/main.go

The results:

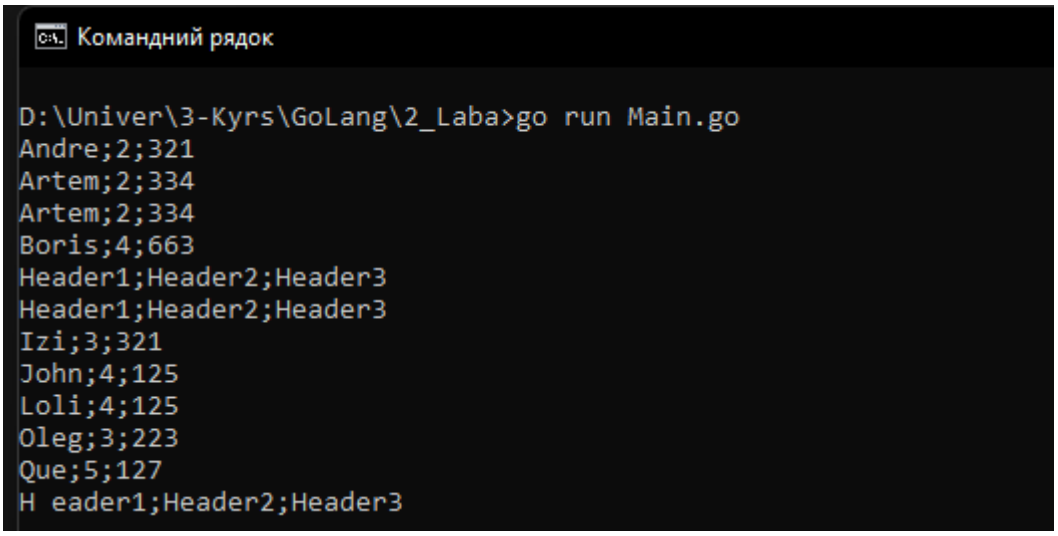

Fig. 1 – Input and output data

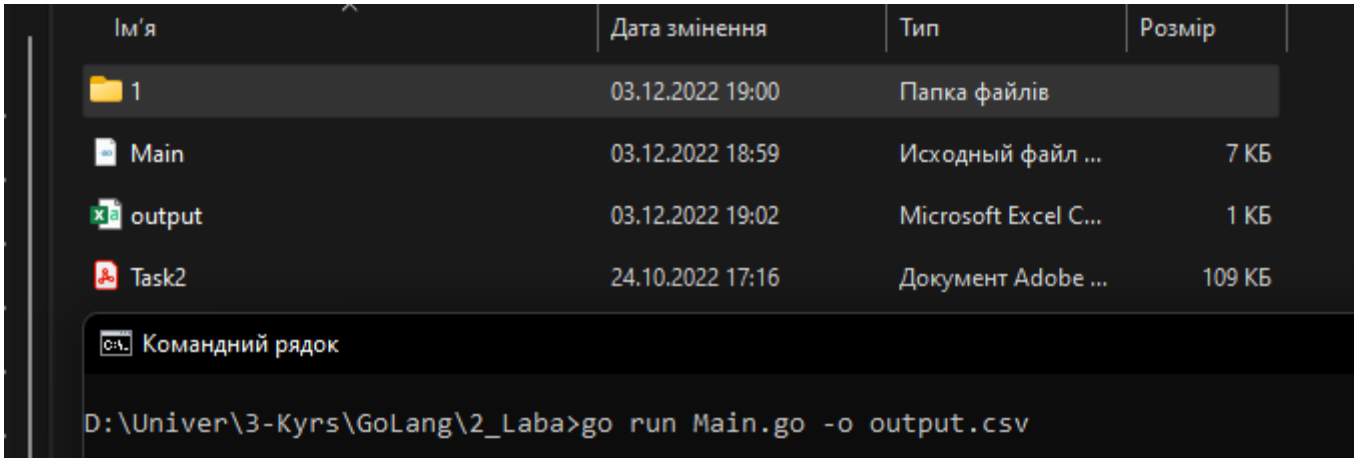

Fig. 2 – The input is output to a file

```
D:\Univer\3-Kyrs\GoLang\2_Laba>go run Main.go -i inputput.csv -d dir_path
2022/12/03 19:03:34 You can't use -i and -d options at the same time
exit status 1

D:\Univer\3-Kyrs\GoLang\2_Laba>
```

Fig. 3 – Using options simultaneously

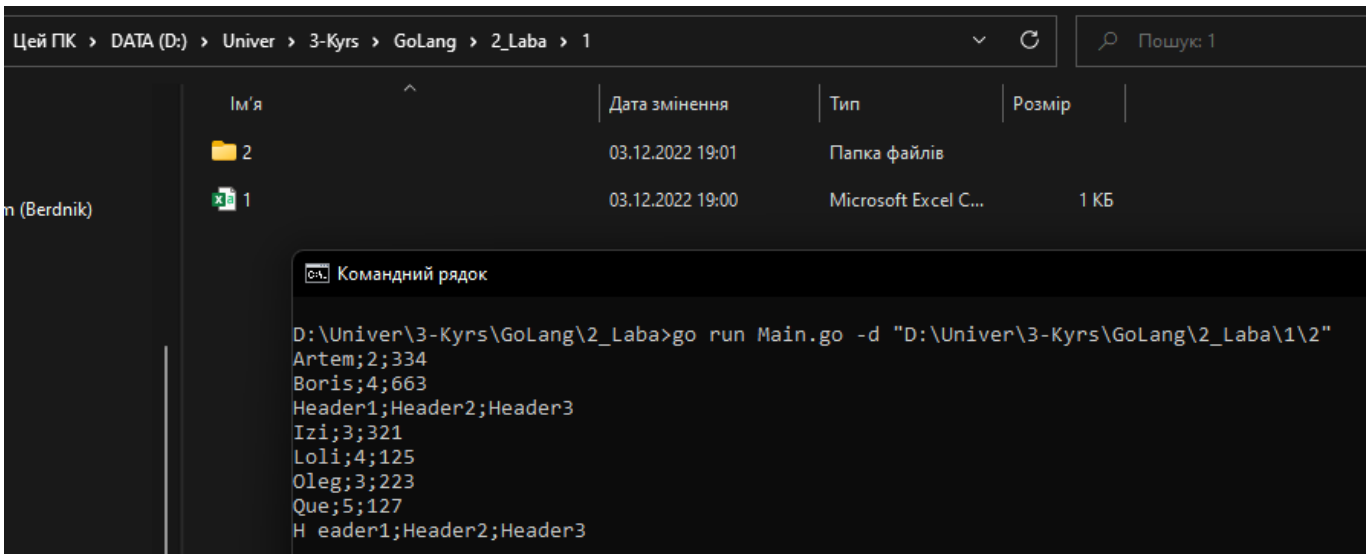

Fig. 4 – Recursive search for files according to the specified path

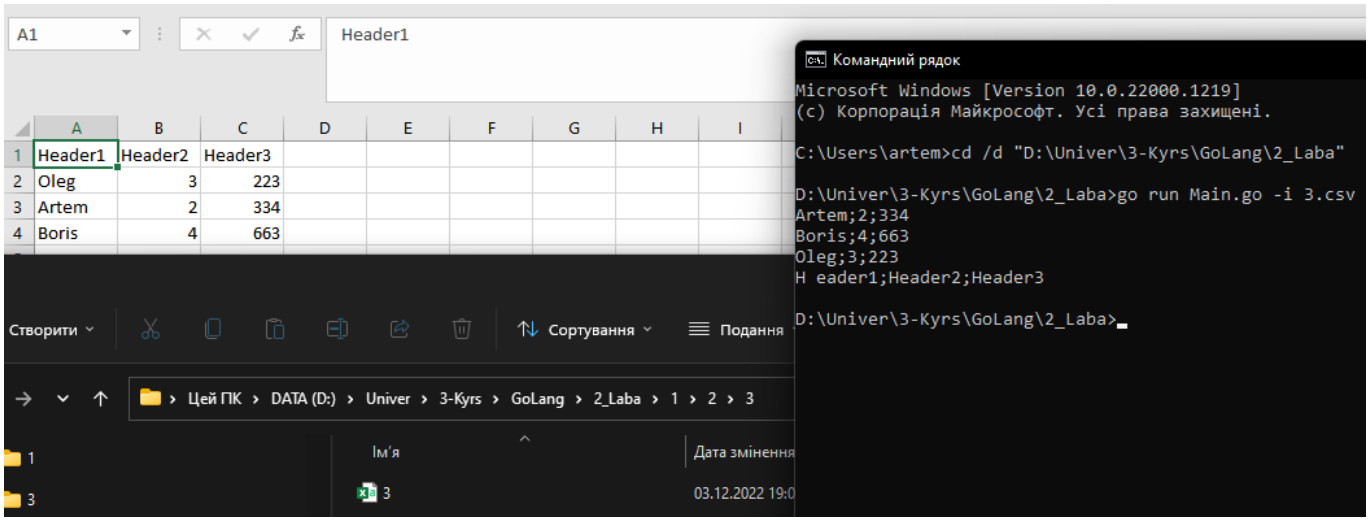

Fig. 5 – Sorting a specific file

Conclusion:

Significantly improved Go language skills, including general knowledge of the language's syntax. Using channels and goroutines, a pipeline is built that performs the operations specified by the condition. Received practical skills of working with channels; learned how to synchronize goroutine work (correct sending/receiving data from channels). I got the experience of breaking one task (conveyor link) into parts. Having chosen the tree sorting algorithm, I significantly optimized the operation of the program.