Scripting Languages – Tasks for the 1st Cycle of Laboratory Exercises

March 2024

---

1. Introduction

In the first cycle of laboratory exercises, students will reinforce and practically apply knowledge of basic Unix tools, regular expressions, and writing simple bash shell scripts. Students are required to prepare for the lab by independently solving a series of simple tasks. For tasks involving file operations (searching contents, renaming, etc.), students must prepare files suitable for testing and demonstrating their programs.

A prerequisite for participating in the lab exercise is uploading the solution files through the Ferko system (https://ferko.fer.hr/ferko/). When uploading files, students must follow the naming conventions and script execution instructions (such as passing parameters).

During the lab session, students will complete a short knowledge check, solve one programming task independently on a computer, and defend their task solutions in front of the teaching assistant.

1.1 Laboratory Session Protocol

In short, the lab protocol is as follows:

1. The student must solve the assigned tasks and submit the solution files through Ferko (mind the submission deadline!).
2. The student must attend the lab session at their scheduled time.
3. At the start of the lab session, students take a short quiz.
4. During the lab session, students receive a task to be solved on the spot (exit test). The task is solved on the computer and submitted via Ferko. Submissions **must be locked!**
5. During the session, students must present their prep-task solutions to the teaching assistant, who evaluates them.

1.2 Laboratory Resources

The exercises are conducted in the university laboratories. The computers are equipped with the Cygwin environment and required tools (bash, sed, grep, etc.).
Students may complete the exercises on their own computers but must ensure network connectivity (for submission through Ferko).

Note: If using a Windows text editor, be sure to use Unix-style line endings (EOL = Line Feed), otherwise bash scripts may fail to execute properly.

Task 1

Getting familiar with shell variables and command line expansion.
Upload instructions: Write all commands for the sub-tasks into a single file named `zadatak1.sh`.

- Set a shell variable `proba` to the value `"Ovo je proba"`.
- Print the value of the set variable.
- Store the list of all files in the current directory into a variable `lista_datoteka` using filename expansion. Print the result.
- Create a new variable `proba3` that consists of the value of `proba` repeated three times, each sentence ending with a period and a space.
- Set variables `a=4`, `b=3`, and `c=7`. Then calculate and store into variable `d` the result of the expression `(a + 4) * b % c` using arithmetic expansion. Print the values of `a`, `b`, `c`, and `d`.
- Store into `broj_rijeci` the total number of words in `.txt` files in the current directory using command substitution and `wc`.
- Print the contents of your home directory using tilde expansion.

Task 2

The `grep` command, regular expressions, `find` command, programming loops.
Upload instructions: Write all commands for the sub-tasks into a single file named `zadatak2.sh`.

- Write a `grep` command to find and print all lines in `namirnice.txt` that contain names of fruits (banana, apple, strawberry, melon, watermelon), case-insensitively.
- Modify the previous command so that it prints only lines **not** containing the specified words.
- Write a `grep` command to search within the `~/projekti/` directory and all subdirectories for lines in files that contain a code in the format of **three uppercase letters followed by a six-digit number**, with a space separating the code from surrounding text.
- Write a command to list the names and detailed info of all files in the current directory and its subdirectories that were modified between 7 and 14 days ago.
- Write a one-liner `for` loop that prints numbers from 1 to 15, using either a sequence expression or the `seq` command.
- Modify the previous loop so that the upper limit is defined by a variable `kraj`. Check if it makes a difference whether you use a sequence expression or `seq`.

Task 3

The repository on the course page contains two log files from a web server. These are text files generated daily, and the date is part of the filename. Each line corresponds to a single access to the server.

Write a script that loops through all files generated during **February** and prints data about the actions performed.
For each file, print the date, then for each action appearing in the logs, print how many times it occurred **that day**.
Sort actions by descending frequency, and print the count before the action itself.

The script should take one argument: the name of the directory containing the log files.
The script must check whether the given directory exists; if not, print usage instructions and exit.

Sample script execution:

```
$ ./zadatak3.sh ../TestPrimjeri/

datum: 24-02-2008
--------------------------------------------------
693 : GET /burza/b/Main.action HTTP/1.1
603 : GET /favicon.ico HTTP/1.1
567 : GET /burza/css/default.css HTTP/1.1
...
```

Upload instructions: Name the script `zadatak3.sh`, and pass the log directory name as an argument. Include all required checks in the script.

Task 4

A directory contains photos transferred from a camera. Each filename includes the photo date and time in the format: `YYYYMMDD_HHMM.jpg`.

Write a script that lists photo filenames from a directory (given as a command-line argument), **grouped by month**.
Photos in each group should be sorted by time and numbered. At the end of each group, print the total number of photos in that group.

The script must verify the given directory exists; otherwise, print usage instructions and exit.

For testing, a tar archive with example "photos" is available on the course page. Extract it using:

```
tar -xvf testovi.tar
```

Sample script execution:

```
$ ./zadatak4.sh ../Testovi/Slike/
01-2020 :
----------
```

```
1. 20200102_1025.jpg
2. 20200102_1915.jpg
...
--- Ukupno: 9 slika -----

02-2020 :
----------
1. 20200207_1706.jpg
2. 20200212_0954.jpg
3. 20200214_1234.jpg
4. 20200219_1541.jpg
5. 20200221_1134.jpg
6. 20200222_1609.jpg
--- Ukupno: 6 slika -----
...
```

Upload instructions: Name the script `zadatak4.sh`, pass the photo directory as an argument, and include necessary checks.

Task 5

Write a script that, starting from a given directory (argument 1), traverses all subdirectories and counts the total number of lines in files matching a pattern (argument 2, e.g., `'*.c'`).

At the beginning of the script, print the arguments passed to it.

Upload instructions: Name the script `zadatak5.sh`, and pass both the directory and file pattern as arguments.

Task 6

Write a script for **two-way synchronization** of two directories given as command-line arguments.
To simplify testing, the script should **simulate** synchronization – instead of actually copying files, it should **print messages** indicating the copy direction.

Two-way synchronization means copying newer or new files from one directory to the other, with both directories being equal peers.
Assume only files are in the directories (no subdirectories).
Use the `test` command with `-nt` (newer than) to compare file modification times.

Example: `[ file1 -nt file2 ]`

The script takes two directories as arguments and it must check that both directories exist, and if not, print usage instructions and exit.

For testing, use `dir1` and `dir2` from the tar archive described in Task 4.

Sample script execution:

```
bash
CopyEdit
$ ./zadatak6.sh ../Testovi/dir1 ../Testovi/dir2
../Testovi/dir1/aaa --> ../Testovi/dir2
../Testovi/dir2/ccc --> ../Testovi/dir1
...
```

Upload instructions: Name the script `zadatak6.sh`, pass the two directories as arguments, and include all necessary checks.