

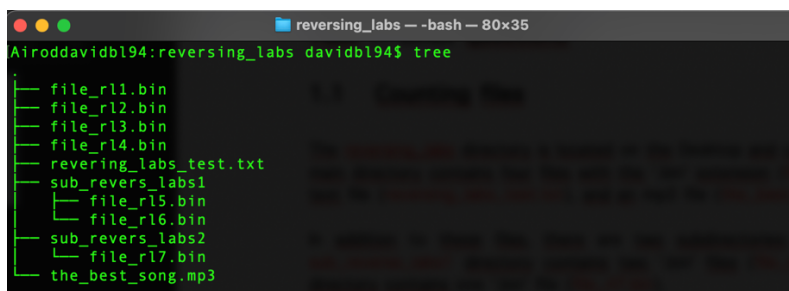
Task 1: Linux questions

1.1 Counting files

The **reversing_labs** directory is located on the Desktop and contains a variety of files and subdirectories. The main directory contains four files with the '.bin' extension (**file_rl1.bin**, **file_rl2.bin**, **file_rl3.bin**, **file_rl4.bin**), a text file (**reversing_labs_test.txt**), and an mp3 file (**the_best_song.mp3**).

In addition to these files, there are two subdirectories: **sub_revers_labs1** and **sub_revers_labs2**. The **sub_revers_labs1** directory contains two '.bin' files (**file_rl5.bin**, **file_rl6.bin**), and the **sub_revers_labs2** directory contains one '.bin' file (**file_rl7.bin**).

Directory structure:



```
reversing_labs -- -bash -- 80x35
Airoddavidbl94:reversing_labs davidbl94$ tree
.
├── file_rl1.bin
├── file_rl2.bin
├── file_rl3.bin
├── file_rl4.bin
├── reversing_labs_test.txt
├── sub_revers_labs1
│   ├── file_rl5.bin
│   └── file_rl6.bin
├── sub_revers_labs2
│   └── file_rl7.bin
└── the_best_song.mp3
```

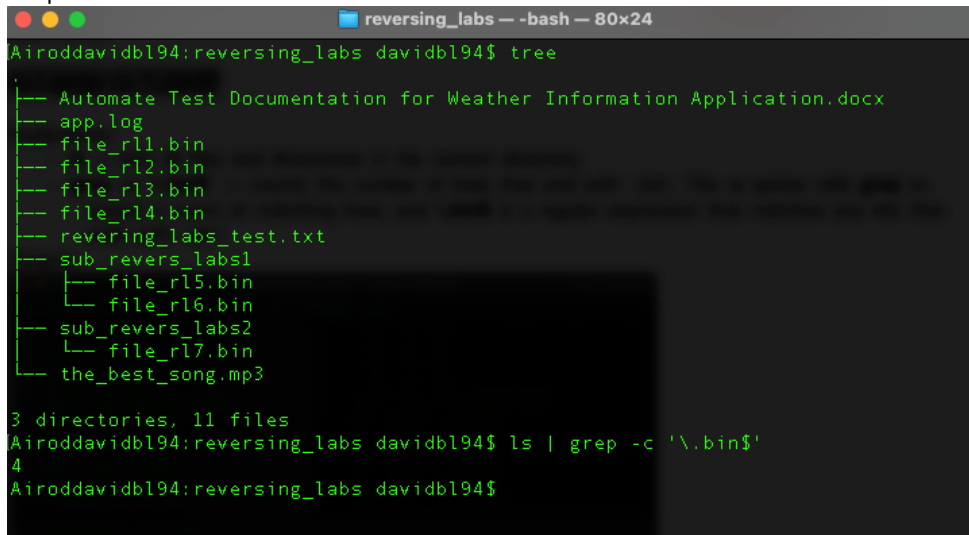
To count the number of files with the extension '.bin' only in that folder (reversing_labs/) (not in subfolders):

```
ls | grep -c '\.bin$'
```

Explanation:

- **ls** -> lists all files and directories in the current directory
- **grep -c '\.bin\$'** -> counts the number of lines that end with '.bin'. The **-c** option tells **grep** to output a count of matching lines, and **\.bin\$** is a regular expression that matches any line that ends with '.bin'.

Output:



```
reversing_labs -- -bash -- 80x24
Airoddavidbl94:reversing_labs davidbl94$ tree
.
├── Automate Test Documentation for Weather Information Application.docx
├── app.log
├── file_rl1.bin
├── file_rl2.bin
├── file_rl3.bin
├── file_rl4.bin
├── reversing_labs_test.txt
├── sub_revers_labs1
│   ├── file_rl5.bin
│   └── file_rl6.bin
├── sub_revers_labs2
│   └── file_rl7.bin
└── the_best_song.mp3

3 directories, 11 files
Airoddavidbl94:reversing_labs davidbl94$ ls | grep -c '\.bin$'
4
Airoddavidbl94:reversing_labs davidbl94$
```

1.2 Comparing files

To verify if the files are identical, I can use a **checksum** -> **sha256sum** command.

1. Generate checksums:

- To generate checksum value of the 100 GB file I should use tool **sha256sum** that is available on most Linux distros.
- Using the command **sha256sum** in Linux first I'll generate a SHA-256 checksum of my 100 GB file:

```
sha256sum 100_gb_file
```

- The output should be a long string of numbers and letters. And that is the SHA-256 of my 100 GB file.

2. Compare checksums over satellite phone:

- I'll instruct my friend to call me via satellite phone.
- Once connected, I'll ask it to run the same **sha256sum** command on their 100 GB file in Antarctica.
- Then, we will compare the received checksum with my own (letter by letter or in small chunks to avoid errors). I'll ask my friend to repeat the dictated checksum to minimize the errors during transmission.

3. Determine corruption:

- If checksums match, the files are identical and their file is not corrupted.
- If the checksums don't match, their file is different from mine and might be corrupted.

Benefits:

- This method is efficient as it only transmits a small amount of data over the "expensive" satellite call.
- Checksums are reliable indicators of file integrity.

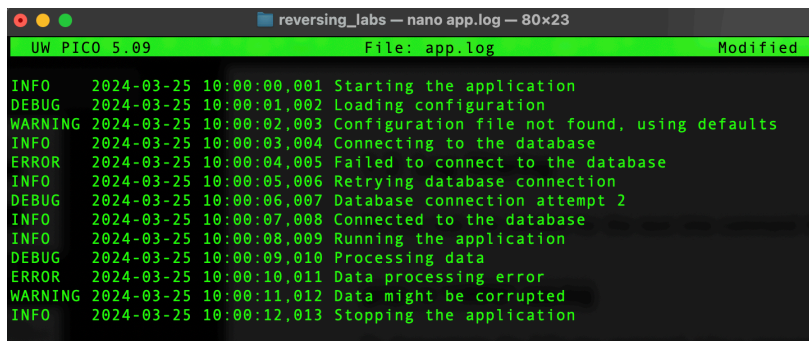
1.3 Log filtering

For this task I've decided for the next Unix command that will filter out (eliminate) lines beginning with INFO:

```
grep -v '^INFO' app.log
```

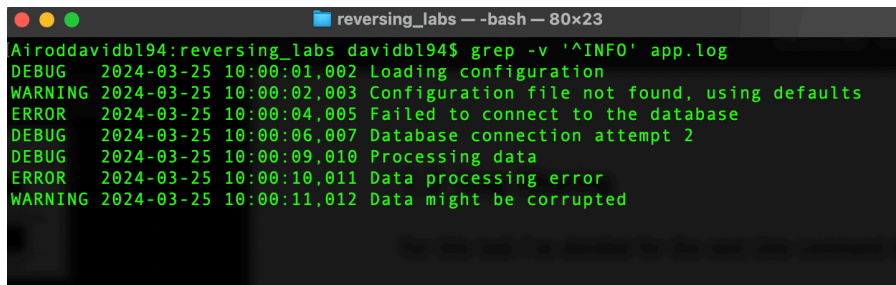
- **grep** -> command allows us to match lines in a file against a pattern and print out the results
- **-v** -> in our case, we want to exclude lines that begin with 'INFO', so we use -v option to invert the matching
- **'^INFO'** -> the pattern '^INFO' to match lines that start with 'INFO'

To demonstrate the following command, I have prepared a log file named **app.log**. This log file contains various log messages with the severity levels (INFO, DEBUG, WARNING, ERROR):



```
reversing_labs — nano app.log — 80x23
UW PICO 5.09 File: app.log Modified
INFO 2024-03-25 10:00:00,001 Starting the application
DEBUG 2024-03-25 10:00:01,002 Loading configuration
WARNING 2024-03-25 10:00:02,003 Configuration file not found, using defaults
INFO 2024-03-25 10:00:03,004 Connecting to the database
ERROR 2024-03-25 10:00:04,005 Failed to connect to the database
INFO 2024-03-25 10:00:05,006 Retrying database connection
DEBUG 2024-03-25 10:00:06,007 Database connection attempt 2
INFO 2024-03-25 10:00:07,008 Connected to the database
INFO 2024-03-25 10:00:08,009 Running the application
DEBUG 2024-03-25 10:00:09,010 Processing data
ERROR 2024-03-25 10:00:10,011 Data processing error
WARNING 2024-03-25 10:00:11,012 Data might be corrupted
INFO 2024-03-25 10:00:12,013 Stopping the application
```

Once the command **grep -v '^INFO' app.log** is executed, it will print all lines from app.log that do not start with 'INFO':



```
reversing_labs — -bash — 80x23
Airoddavidbl94:reversing_labs davidbl94$ grep -v '^INFO' app.log
DEBUG 2024-03-25 10:00:01,002 Loading configuration
WARNING 2024-03-25 10:00:02,003 Configuration file not found, using defaults
ERROR 2024-03-25 10:00:04,005 Failed to connect to the database
DEBUG 2024-03-25 10:00:06,007 Database connection attempt 2
DEBUG 2024-03-25 10:00:09,010 Processing data
ERROR 2024-03-25 10:00:10,011 Data processing error
WARNING 2024-03-25 10:00:11,012 Data might be corrupted
```