

Operating Systems: Playing Around With Bash

V. Markos
Mediterranean College

October 12, 2025

Exercise 1. Create a bash script that:

- Creates a directory named `animals`.
- Creates two subdirectories of `animals`, named `vertebrate` and `invertebrate`.
- Creates the following subdirectories in `vertebrate`:
 - `agnatha`;
 - `gnathostomata`.
- Creates the following subdirectories in `agnatha`:
 - `conodontata`;
 - `ostracodermi`;
 - `cyclostomi`.
- Creates the following subdirectories in `invertebrate`:
 - `insects`;
 - `arachnids`;
 - `anthozoa`.

In the end, your directory structure should look as the one shown in Figure 1. **Keep the above directory structure as it will be used throughout the labs.**

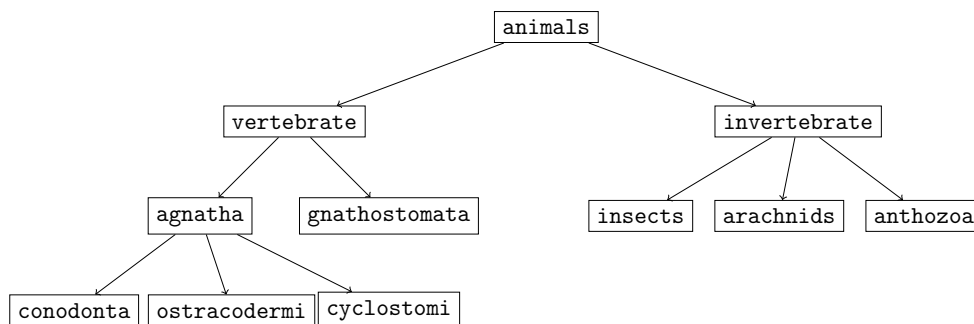


Figure 1: A nice directory structure about animal taxonomy.

Exercise 2. Using the directory structure created in Exercise 1, write a bash script that:

- Creates a file named `some-insects.txt` in the `insects` directory with the following contents (with the shown spacing):

```

1 # some-insects.txt
2 insect      legs  body-segments
3 spider      8     2
4 centipede   44    22
5 woodlouse   14     7
6 millipede   72    18
7

```

- Upon the creation of the file, prints on screen:
 - its character length;
 - the absolute path it has been created in.

Exercise 3. Write a bash script that:

- reads the file `some-insects.txt` created in Exercise 2;
- sorts its lines alphabetically by insect name, ignoring the first two lines which contain the file name and the corresponding column names, and prints the output into a file named `insect-by-names.txt`;
- sorts its lines by increasing insect legs, ignoring the first two lines which contain the file name and the corresponding column names, and prints the output into a file named `insect-by-legs.txt`;
- sorts its lines by decreasing number of body segments, ignoring the first two lines which contain the file name and the corresponding column names, and prints the output into a file named `insect-by-segments.txt`.

Exercise 4. Create a bash function that wraps the functionality created in Exercise 3 as follows:

- the function should accept a single argument which should be either `name`, `legs`, or `segments` corresponding to the `insect`, `legs` and `body-segments`, respectively;
- the function should then perform the desired sorting, exactly as in Exercise 3;
- the function should print a relevant error message on screen in case of invalid argument(s).

As a variation, consider allowing for a second argument that will be the target file's name. In that case, you should properly check if the file already exists and throw a relevant warning.

Exercise 5. Implement a bash script that:

- takes a single directory as argument;
- checks if that directory exists and is indeed a directory and, if not, it prints a relevant message on screen and terminates;
- if the provided argument is indeed a valid directory, then it transverses and prints all its subdirectories using the Breadth-First-Search (BFS) algorithm.

You can find more on the BFS algorithm here:

https://en.wikipedia.org/wiki/Breadth-first_search

Exercise 6. Implement a bash script that:

- takes a single directory as argument;
- checks if that directory exists and is indeed a directory and, if not, it prints a relevant message on screen and terminates;
- if the provided argument is indeed a valid directory, then it transverses and prints all its subdirectories using the Depth-First-Search (BFS) algorithm.

You can find more on the DFS algorithm here:

https://en.wikipedia.org/wiki/Depth-first_search

Exercise 7. Create a bash script that starting from a certain directory prints its subdirectories as a tree up to three levels on the terminal, excluding any files that might be contained in any of the encountered subdirectories. You should follow the instructions below for printing:

- Use the underscore, `_`, and pipe, `|`, characters to draw the arrows of the directory tree.
- Do not use any readymade bash scripts regarding directory tree printing.

For instance, for the directory structure shown in Figure 1 your script should print something like the following:

```
animals
|_vertebrate
|__agnatha
|___conodonta
|___ostracodermi
|___cyclostomi
|__gnathostomata
|_invertebrate
|__insects
|__arachnids
|__anthozoa
```