UNIVERSITÉ DU LUXEMBOURG

Operating Systems 1

Christian Grévisse (christian.grevisse@uni.lu)
Alisa Vorokhta (alisa.vorokhta@uni.lu)

Lab 1 – Basic Linux Commands

2024–09–20

Exercise 1 – `cd`, `nano`, `rmdir`, `rm`

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Go to the `Documents` directory.

3° Create a new directory within the `Documents` directory.

4° Within this directory, create a new text file containing some piece of text.

5° *Try* to delete the new directory with `rmdir`, what do you notice?

6° *Try* to delete the new directory using `rm`, what do you notice?

7° What option do you need to specify to `rm` in order to delete a non-empty directory? (Hint: You might want to consult the `man` page of `rm` ...)

Exercise 2 – `touch`, `mv`, `cp`, `cat`, `diff`

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Create a new, empty file.

3° Move it to the `Desktop`.

4° Change its name.

5° Create a copy of the file in your home folder.

6° Change the contents of that copy with `nano`.

7° Display the contents of both files.

8° Verify the difference between both files using the `diff` utility.

9° Clean up by removing both files.

Exercise 3 – Copying directories

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Create a new directory `A`.

3° Add a few files to `A`.

4° Verify their creation.

5° Create another directory `B`.

6° Copy `A` into `B`.

7° Check that the copying process was successful.

8° Clean up.

Exercise 4 – Copying files

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Create a new, empty file.

3° Copy it to the `Desktop`.

4° Copy another file with the same name to the Desktop. What happens?

5° Clean up.

6° Do the same exercise again, but this time, specify the `-i` option when copying files. What do you observe?

Exercise 5 – Creating and unpacking archives

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Create a few files and/or directories.

3° Create a tarball archive from these files and directories and compress it with `gzip`.

4° Create a new folder on the `Desktop`.

5° Uncompress the previously created archive to that folder.

6° Clean up.

7° Do the same exercise with `bzip2` compression.

Exercise 6 – `find`

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Move to the `Documents` directory and create a file `test`.

3° Move to the `Desktop` **without** switching first back to the home folder.

4° Create a directory named `test`.

5° Move back to the home directory.

6° Search for both files *and* directories in the home directory and its subfolders named "test".

7° Search only for files named "test" in the home directory and its subfolders.

8° Search only for directories named "test" in the home directory and its subfolders.

9° Clean up **without** using `rm` or `rmdir`.

Exercise 7 – `wc`, `grep`

Open a new shell and perform the following tasks:

1° Make sure you are currently within your home directory.

2° Create a new text file containing the following text:

> *This*
> *is*
> *a*
> *text*
> *file*
> *containing*
> *some*
> *plaintext.*

3° How many lines are there (use `wc`)?

4° Search for all lines containing "is".

5° Search for all lines containing the **word** "is".

6° Display all lines which do **not** contain "is".

7° Remove the file.

Exercise 8 – Wrapping it up

Open a new shell and perform the following tasks:

1° In the directory `/usr/include` (and its subdirectories), find all *files* with extension `.h` that contain the *word* `#define` and color the matches.

2° For each of these files, print the name and the frequency of `#define`.