C Piscine

Day 01

Staff 42 pedago@42.fr

Summary: This document is the subject for Day01 of the C Piscine @ 42.

Contents

Chapter 1

Instructions

- The exercises are carefully laid out in order of difficulty, from easiest to hardest. An exercise is only graded if all previous ones are correct. In other words: the grading for a day stops at the first mistake.
- Be mindful of the submission procedures indicated at the start of every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. Be as thorough as possible!
- You should not leave any additional file in your directory than those specified in the subject.
- Examine the examples thoroughly. They often contain details that are not explicitly mentioned in the subject.
- All shell scripts and commands must be executable with **/bin/sh**

ⓘ **The forewords are entirely unrelated to the subjects and can safely be ignored.**

Chapter 2

Topics

Today, you will have to learn about:

- More basic Unix commands and utilities.

Chapter 3

Foreword

Here's what **Wikipedia** has to say about otters :

The European otter (Lutra lutra), also known as the Eurasian otter, Eurasian river otter, common otter and Old World otter, is a European and Asian member of the Lutrinae or otter subfamily, and is typical of freshwater otters.

The European otter is a typical species of the otter subfamily. Brown above and cream below, these long, slender creatures are well-equipped for their aquatic habits. Its bones show osteosclerosis, increasing their density to reduce buoyancy.
This otter differs from the North American river otter by its shorter neck, broader visage, the greater space between the ears and its longer tail.

However, the European otter is the only otter in its range, so it cannot be confused for any other animal. Normally, this species is 57 to 95 cm (23-37 in)
long, not counting a tail of 35–45 cm (14–18 in).
The female is shorter than the male.

The otter's average body weight is 7 to 12 kg (15.4–26.4 lbs), although occasionally a large old male may reach up to 17 kg (37 lbs).
The record-sized specimen, reported by a reliable source but not verified, weighed over 24 kg (53 lbs).

The European otter is the most widely distributed otter species, its range including parts of Asia and Africa, as well as being spread across Europe. Though currently believed to be extinct in Liechtenstein, and Switzerland, they are now very common in Latvia, along the coast of Norway and across Great Britain, especially Shetland, where 12% of the UK breeding population exist. Ireland has the highest density of Eurasian otters in Europe.
In Italy, they can be found in southern parts of the peninsula.
The South Korean population is endangered.

Otters are cute.

Chapter 4

Exercise 00 : Exam

	Exercise : 00
Exam	

- During the week, you will be able to sign up for Friday's exam on the Intra. Don't forget!
- You also have to register to the Exam00 project.

Make sure you have registered for both the event AND the project!

Chapter 5

Exercise 01 : print_groups

Turn-in directory : ex01/

Files to turn in: print_groups.sh

Allowed functions: None

- Write a script that will display the list of groups for which the user, contained in the environment variable FT_USER, is a member (separated by commas without spaces).
- Examples :
  - for FT_USER=student, the result is ”student,:,student,staff,main,basecamp,fortytwo” (without quotation marks)

    ```
    $>./print_groups.sh
    student,:,student,staff,main,basecamp,fortytwo$>
    ```

  - for FT_USER=staff, the result is ”staff,:,staff,god,main,bocal” (without quotation marks)

    ```
    $>./print_groups.sh
    staff,:,staff,god,main,bocal$>
    ```

man groups

Chapter 6

Exercise 02 : find_sh

Turn-in directory : ex02/

Files to turn in: find_sh.sh

Allowed functions: None

- Write a script that searches for all file names that end with ”.sh” (without quotation marks) in the current directory and all its sub-directories. It should display only the file names without the .sh.
- Example of output :

```
$>./find_sh.sh | cat -e
find_sh$
file1$
file2$
file3$
$>
```

Chapter 7

Exercise 03 : count_files

Turn-in directory : ex03/

Files to turn in: count_files.sh

Allowed functions: None

- Write a script that counts and displays the number of regular files and directories in the current directory and all its sub-directories. It should include ".", the starting directory.
- Example of output :

```
$>./count_files.sh | cat -e
42$
$>
```

Chapter 8

Exercise 04 : MAC

Turn-in directory : ex04/

Files to turn in: MAC.sh

Allowed functions: None

- Write a script that displays your machine's MAC addresses. Each address must be followed by a line break.

Example:

```
>$ ./MAC.sh | cat -e
f2:5b:44:60:f5:aa$
8a:ec:c7:ac:30:81$
ca:17:6b:da:36:fc$
02:42:0a:02:03:0a$
```

man ifconfig

Chapter 9

Exercise 05 : Can you create it ?

Turn-in directory : `ex05/`
Files to turn in: `"\?$*'MaRViN'*$?\"`
Allowed functions: `None`

- Create a file containing only "42", and NOTHING else.
- Its name will be :

```
    "\?$*'MaRViN'*$?\"
```

- Example :

```
$>ls -lRa *MaRV* | cat -e
-rw---xr-- 1 75355 32015 2 Oct 2 12:21 "\?$*'MaRViN'*$?\"$
$>
```

Chapter 10

Exercise 06 : Skip

Turn-in directory : ex06/

Files to turn in: skip.sh

Allowed functions: None

- Write a script that displays every other line of the output of the command **ls -l**, starting from the first line.

Chapter 11

Exercise 07 : r_dwssap

Turn-in directory : ex07/

Files to turn in: r_dwssap.sh

Allowed functions: None

- Write a script that displays the output of the **cat /etc/passwd** command, removing comments, every other line starting from the second line, reversing each login, sorted in reverse alphabetical order, and keeping only logins between FT_LINE1 and FT_LINE2 included, and they must separated by ″, ″ (without quotation marks), and the output must end with a ″.″.
- Example: Between lines 7 and 15, the result should be something like this :

```
$> ./r_dwssap.sh
sstq_, sorebrek_brk_, soibten_, sergtsop_, scodved_, rlaxcm_, rgmecived_,
    revreswodniw_, revressta_.$>
```

Rigorously follow the order indicated in the instructions.

Chapter 12

Exercise 08 : add_chelou

Turn-in directory : ex08/

Files to turn in: add_chelou.sh

Allowed functions: None

- Write a script that takes numbers from variables **FT_NBR1**, in **'\"?!** base, and **FT_NBR2**, in **mrdoc** base, and displays the sum of both in **gtaio luSnemf** base.
  - Example 1:

    ```
    FT_NBR1=\'?"\"'\
    FT_NBR2=rcrdmddd
    ```

  - The sum is :

    ```
    Salut
    ```

  - Example 2 :

    ```
    FT_NBR1=\"\"!\"\"!\"\"!\"\"!\"\"!\"\"
    FT_NBR2=dcrcmcmooododmrrrmorcmcrmomo
    ```

  - The sum is :

    ```
    Segmentation fault
    ```