C Piscine

Shell 01

Staff 42 pedago@42.fr

Summary: *This document is the subject for the module Shell 01 of the C Piscine @ 42.*

Contents

Chapter I

Instructions

- Only this page will serve as reference; do not trust rumors.

- Watch out! This document could potentially change up before submission.

- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We `will not` take into account a successfully completed harder exercise if an easier one is not perfectly functional.

- Make sure you have the appropriate permissions on your files and directories.

- You have to follow the submission procedures for 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. So if you want to avoid bad surprises, be as thorough as possible.

- Exercises in Shell must be executable with /bin/sh.

- You cannot leave any additional file in your directory than those specified in the subject.

- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.

- Your reference guide is called `Google / man / the Internet / ...`.

- Check out the "C Piscine" part of the forum on the intranet.

- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...

Chapter II

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 III

Exercise   00 : Exam

	Exercise : 00
	Exam
Notes : `n/a`	

- During the week, you will be able to sign up for Friday's exam in the agenda, don't forget.

- You also have to register for the Exam00 project.

- Make sure you've registered for the exam (the event AND the project !).

- Make sure you've made sure you've registered for the exam (the event AND the project ! Yep, both !).

Chapter IV

Exercise 01 : print_groups

	Exercise 01
	print_groups.sh
Turn-in directory : *ex01/*	
Files to turn in : `print_groups.sh`	
Allowed functions : `None`	
Notes : `n/a`	

- Write a command line that will display the list of groups for which the login, contained in the environment variable FT_USER , is a member. Separated by commas without spaces.

- Examples :

  - for FT_USER=nours, the result is "god,root,admin,master,nours,bocal" (without quotation marks)
    ```
    $>./print_groups.sh
    god,root,admin,master,nours,bocal$>
    ```

  - for FT_USER=daemon, the result is "daemon,bin" (without quotation marks)
    ```
    $>./print_groups.sh
    daemon,bin$>
    ```

```
man id
```

Chapter V

Exercise 02 : find__sh

	Exercise 02
	find__sh.sh
Turn-in directory : *ex02/*	
Files to turn in : `find_sh.sh`	
Allowed functions : `None`	
Notes : `n/a`	

- Write a command line 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 VI

Exercise 03 : count_files

	Exercise 03
	count_files.sh
Turn-in directory : *ex03/*	
Files to turn in : `count_files.sh`	
Allowed functions : `None`	
Notes : `n/a`	

- Write a command line 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 VII

Exercise 04 : MAC

	Exercise 04
	MAC.sh
Turn-in directory : *ex04/*	
Files to turn in : `MAC.sh`	
Allowed functions : `None`	
Notes : `n/a`	

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

```
man ifconfig
```

Chapter VIII

Exercise 05 : Can you create it ?

	Exercise 05
	Can you create it ?
Turn-in directory : *ex05/*	
Files to turn in : "\?$*'MaRViN'*$?\"	
Allowed functions : `None`	
Notes : `n/a`	

- Create a file containing <u>only</u> "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 IX

Exercise 06 : Skip

	Exercise 06
	skip.sh
Turn-in directory : *ex06/*	
Files to turn in : `skip.sh`	
Allowed functions : `None`	
Notes : `n/a`	

- Write a command line that displays every two lines for the command ls -l, starting from the first line.

Chapter X

Exercise 07 : r__dwssap

	Exercise 07
	r__dwssap.sh
Turn-in directory : *ex07/*	
Files to turn in : `r_dwssap.sh`	
Allowed functions : `None`	
Notes : `n/a`	

- Write a command line that displays the output of a `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 XI

Exercise 08 : add_chelou

	Exercise 08
	add_chelou.sh
Turn-in directory : *ex08/*	
Files to turn in : `add_chelou.sh`	
Allowed functions : `None`	
Notes : `n/a`	

- Write a command line 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=dcrcmcmooodododmrrrmorcmcrmomo
        ```

    - The sum is :
        ```
        Segmentation fault
        ```