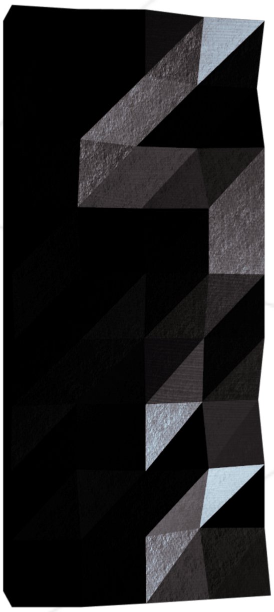

C Piscine

Day 24

Staff 42 pedago@42.fr

Summary: THE FOLLOWING TAKES PLACE BETWEEN 3.00 A.M. AND 4.00 A.M.

Contents

Chapter I

Instructions

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

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

- 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.

- Moulinette is not very open-minded. It won't try and understand your code if it doesn't respect the Norm.

- Using a forbidden function is considered cheating. Cheaters get `-42`, and this grade is non-negotiable.

- If ft_putchar() is an authorized function, we will compile your code with our `ft_putchar.c`.

- You'll only have to submit a main() function if we ask for a program.

- Moulinette compiles with these flags: -Wall -Wextra -Werror, and uses `gcc`.

- If your program doesn't compile, you'll get `0`.

- 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...

- By Odin, by Thor ! Use your brain !!!

Chapter II

Foreword

It was a classic awakening for Chloe O'Brian, rebel director of CTU New York.

In a groan, she stretched and cursed her alarm, settled like everyday on 6:00 AM. As usual, she decided to take stock of her situation. An unbearable headache gave him a clue that the night of yesterday was very watery. Confident, she opened her eyes to realize her error all along the line. Chloe was lying on the ground with her head slightly bloodied, in the CTU's elevator, where a strident alert message was broadcast continuously: "Windows Elevator 8 planted". "How long have I stuck?" She wondered as she got up.

Slowly, the chain of events came back to her: how she arrived late last night at the office, the shattering return of Nick Bauer, the copier repairer who had no screwdrivers, the grotesque explosions of his agents, Use the elevator, the fall of the apparatus between two floors and its brutal commotion.

"Yes, that's it. "

Chloe could begin to panic.

"No, I will not lose my means, I feel rather safe in an elevator cage already fallen to the last basement."

However, Chloe succumbed to panic.

"Especially since there is air for several hours, and when I open the evacuation door, I have nothing to fear."

For dramatic reasons, Chloe was obliged to panic.

" Help ! Chloe shouted as she drummed the stuck door of the elevator. I'm trapped!"

Yes, that was it.

`Who will be able to deliver Chloe? Nick Bauer - help him open the elevator.`

Chapter III

ft_door

	Exercise 12
ft_door.h ft_door.c	
Turn-in directory : *ex*12/	
Files to turn in : `ft_door.h, ft_door.c`	
Allowed functions : `write`	
Notes : `n/a`	

- Create the file ft_door.h and fix the following ft_door.c file :

```
#include "ft_door.h"

ft_putstr(char *str)
{
        int i = 0;

        while (str[i])
                write(1, str, i)
}

ft_bool close_door(t_door *door)
{
        ft_putstr(’Door closing...’);
        state = CLOSE;
        return  (TRUE);
}

void is_door_open(t_door door)
{
        ft_putstr("Door is open ?");
        return (door->state = OPEN);
}

ft_bool is_door_close(t_door* door)
{
        ft_putstr("Door is close ?") ;
}
```

- Here's an example of main and output.

```
$> cat main.c
#include <stdlib.h>
#include "ft_door.h"

int main()
{
    t_door      door;

    open_door(&door);
    if (is_door_close(&door))
        open_door(&door);
    if (is_door_open(&door))
        close_door(&door);
    if (door.state == OPEN)
        close_door(&door);
    return (EXIT_SUCCESS);
}
$>./ft_door | cat -e
Door opening...$
Door is close ?$
Door is open ?$
Door closing...$
$>
```