Exercises — Seven int

version #

Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

*https://intra.assistants.epita.fr

1 Seven int

Files to submit:

- seven_int/seven_int.c

Provided files:

- seven_int/seven_int.h

Authorized functions: You are only allowed to use the following functions:

- open(2)
- read(2)
- write(2)
- close(2)

Authorized headers: You are only allowed to use the functions defined in the following headers:

- assert.h
- stddef.h
- errno.h
- err.h

1.1 Goal

The goal of this exercise is to make you more comfortable with `write(2)` and `read(2)` syscalls.

1.2 Write the ints

In this part of the exercise, you will have to write a function called `dump_ints` which will take an array `arr` of seven integers and write them all to a file given in the second parameter `path`. Be careful, we want to write the actual bytes of each integer in the input array, not the characters of their string representation.

For example, when dumping the integer `-1`, you should write the bytes `\xff\xff\xff\xff` assuming integers are 4 bytes long on your architecture.

The prototype of the `dump_ints` function is as follows:

```
int dump_ints(int *arr, const char *path);
```

Your function should return `1` on success and `0` if any error occurs.

Tips

You can use `hexdump(1)` to see the content of a file in hexadecimal

1.3 Read the int

In this part you are going to write the `read_ints` function. This function takes an array `arr` of seven ints and fills it with the integers stored in the file given in the second parameter `path`. Be careful, we want to read the actual bytes of each integer into the input array.

For example, when reading the integer `-1`, the bytes found in the input file will be `\xff\xff\xff\xff` assuming integers are `4` bytes long on your architecture.

The prototype of the `read_ints` function is as follows:

```
int read_ints(int *arr, const char *path);
```

Your function should return `1` on success and `0` if any error occurs.

It is my job to make sure you do yours.