EXERCISES — The Right Tarball

version **#cb69afd89690e908c9ef3ff2545b93df54f86b1f**

I MUST NOT FEAR. FEAR IS THE MIND-KILLER.

ASSISTANTS C/UNIX 2024 <assistants@tickets.assistants.epita.fr>

Copyright

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

Contents

---

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

File Tree

```
right_tarball/
└── my_tarball.tar.gz   (to submit)
```

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

- builtins
- touch(1)
- mkdir(1)
- chmod(1)

Reminder : Grant execution permission to your scripts before pushing them

1 Goal

A tarball is often used for **backups** or **releases** purposes. In a release, you generally find information about the author(s) of the project, some documentation, and source code. Let us make one.

Generate a tarball with a specific architecture and permissions.

Start by creating a folder `my_tarball` at the root of your exercise directory.

2 File handling and permissions

This section must be done inside the `my_tarball` folder.

To check that you have understood notions about how to handle files, directories and their associated permissions correctly, you must:

- create an `AUTHORS` file and modify its permissions. An `AUTHORS` file contains any information about the author(s) of a project. It must have the `640` mode and must not be empty. The content is irrelevant.

- create a new directory named `doc`. It must have the `700` mode.

> **Tips**
>
> A conventional `AUTHORS` file contains a line that can easily be parsed:
>
> ```
> 42sh$ cat -e AUTHORS
> * prenom.nom$
> ```

> **Tips**
>
> If the octal permission system is still unclear to you, take time to read the chmod manual.

Once this is done, you must create a text file:

- This file must be named `rights`, located in the `doc` directory you just created, and have the `640` access rights.

- It should be at least 5 lines long.

- The text must be wrapped at 80 characters, meaning no line should exceed 80 columns.

If you lack imagination, you can write about the permission system and the different commands you can use to manipulate them.

Finally, you must create an `src` directory containing a script:

- The `src` directory must have the `755` mode.

- The script must be located in the `src` directory and named `authors.sh` with the 644 access rights - thus, not executable.

The script will handle the **creation** of an `AUTHORS` file with **custom permissions** in the current working directory (the directory in which you are when you run the script). The created file must have `640` access rights.

3 Assignment tarball

To validate this exercise, you must compress your previous work in a tarball and submit it.

Your tarball must be called `my_tarball.tar.gz`. This file must be a tar archive compressed in the **gzip** format. Take time to read the manual of `tar` to find out how to create this archive.

The tarball must contain the directory `my_tarball` and all its content.

> **Be careful!**
>
> Do not include any trash or git related file.

You can check that your submission is correct by comparing it with this output:

```
42sh$ tar tf my_tarball.tar.gz
my_tarball/
my_tarball/AUTHORS
my_tarball/doc/
my_tarball/doc/rights
my_tarball/src/
my_tarball/src/authors.sh
```

> **Tips**
>
> Using the verbose option can be helpful when debugging your tarball.

I must not fear. Fear is the mind-killer.