ft_tar

manipulating tape archives

Gaetan gaetan@42.us.org

Summary:

1. *Recode "tar"*
2. *Make it work*
3. *???*
4. *Profit!*

Contents

Chapter I

Foreword

I.1 Social Media Demographic Statistics

1. 75% of male internet users are on Facebook as well as 83% of female internet users.
2. 44% of teenagers asked to choose one social network if "trapped on a deserted island" chose Snapchat, ahead of Instagram (24 percent) and Facebook (14 percent). [1]
3. Female internet users are more likely to use Instagram than men, at 38% vs. 26%.
4. 29% of internet users with college degrees use Twitter, compared to 20% with high school degrees or less.
5. 81% of millennials check Twitter at least once per day.
6. 91% of Social Media Users Are Accessing Social Channels Via Mobile Devices [2]
7. 22% of the world's total population uses Facebook.
8. LinkedIn boasts more than 530 millions user profiles.
9. On any given day, Snapchat reaches 41
10. YouTube overall, and even YouTube on mobile alone, reaches more 18-34 and 18-49 year-olds than any cable network in the U.S.

I.2 Social Media Usage Statistics

1. Facebook continues to be the most widely used social media platform, with 79% of American internet users.
2. Instagram receives the silver medal with 32% of users, Pinterest coming in a close third with 31%, and LinkedIn and Twitter at 29% and 24% respectively.
3. In the last quarter of 2017, 1.37 billion active users visited Facebook on a daily basis.

[1] recode

[2] CoSchedule.

4. The average LinkedIn user spends 17 minutes on the site per month.
5. 51% of Instagram users access the platform daily, and 35% say they look at the platform several times per day.
6. Almost 80% of time spent on social media platforms happens on mobile.
7. Katy Perry has the most worldwide twitter followers, at 94.65 million.
8. Over 400 million snaps are shared on Snapchat per day, and almost 9,000 photos are shared every second.
9. Just 10 thousand YouTube videos have generated more than 1 billion views.
10. More than half of all YouTube views are on mobile devices.

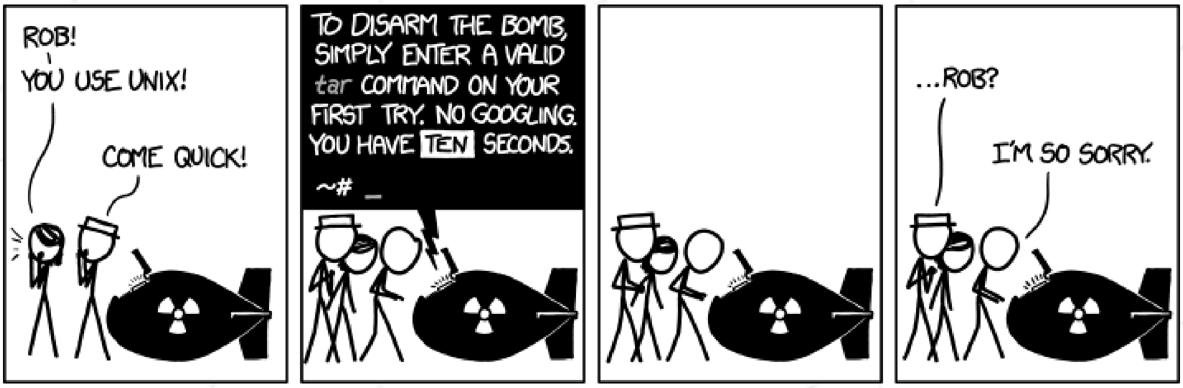
[3]

[3] Source

Chapter II

Introduction

1168: tar [1]


ROB!
YOU USE UNIX!
COME QUICK!
TO DISARM THE BOMB, SIMPLY ENTER A VALID tar COMMAND ON YOUR FIRST TRY. NO GOOGLING. YOU HAVE TEN SECONDS.
~# _
...ROB?
I'M SO SORRY.

google is man's bff

[1] Explain xkcd: It's 'cause you're dumb.

Chapter III

Goals

Make a "tar-like" program that is able to manipulate files and directories.

Chapter IV

General instructions

- You are only entitled to `lib c`.

- Your project must be written in accordance with the Norm.

- V.1 and V.2 are mandatory and will be thoroughly reviewed.

- You must submit a file named `author` containing your username followed by a newline at the root of your repository:

  ```
  $>cat -e author
  xlogin$
  ```

- Your project must compile on FreeBSD (ex: copana or neutron), Sun (ex: maya)

- You have to handle errors in a sensitive manner. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).

- The subject may change up until 5 hours before deadline.

- For any questions about this project, `man tar`.

- The correction will take into account the quality of the code.

`Sunday, Jan 28th 23h42`

Chapter V

Mandatory part

V.1 First Part

Where's your 42 Team Spirit? #catch42
Find and publish a photo of a "42" on twitter / instagram / social media
Make sure you tag @42SiliconValley

V.2 Second Part

V.2.1 Exercise 00

Exercise 00
Work directory: *ex*00/
Files to turn in: necessary files to make it work
Forbidden functions : None
Remarks : n/a

Make a **ft_archive** app that takes several files as parameters to make an archive:

Syntax: `./ft_archive file_archive file1 [file2 ... fileN]`

Example:

```
$>ls
file1 file2 file3 ft_archive
$>./ft_archive toto file1 file2 file3
-> Create archive from file1, file2, and file3 in toto
-> done
$>ls
  file1 file2 file3 ft_archive toto
```

And a **ft_unarchive** app that recovers files from an archive given in parameter:

Syntax: `./ft_unarchive archived_file`

Example: (using the `toto` of the previous example)

```
$>ls
ft_unarchive toto
$>./ft_unarchive toto
-> Get file1, file2, and file3 from toto
-> done
$>ls
file1 file2 file3 ft_archive toto
```

V.2.2 Exercise 01

Exercise 1
Work directory: *ex*01/
Files to turn in: necessary files to make it work
Forbidden functions : None
Remarks : n/a

To validate the second step, your binary **ft_archive** must be able to archive directories recursively (files and directories content)

Your **ft_unarchive** must be able to recreate directories and files preserving the original arborescence and contents.

The binary then takes the following parameters:

Syntax: `ft_archive archive_file directory [file2 / ...  dirfileN /]`
Syntax: `ft_unarchive archived_file`

Example:

```
$>ls -R
file1 file2 directory1

./directory1:
file3 file4 directory2

./directory1/directory2:
file5
$>ft_archive toto *
$>rm -rf file1 file2 directory1
$>ft_unarchive toto
$>ls -R
file1 file2 toto directory1

./directory1:
file3 file4 directory2

./directory1/directory2:
file5
```

V.2.3 Exercise 02

Exercise 2
Work directory: *ex*02/
Files to turn in: necessary files to make it work
Forbidden functions : None
Remarks : n/a

This step automatically validates the first two because it consists of realizing the application `ft_tar` which behaves as the command unix tar (same syntax) and which manages both `ft_archive` and recursive `ft_unarchive` functions, your program must also manage the permissions and the dates of the files archives (`-p` option). The arguments your program takes must be the same as those of the tar command. The `xcvftp` options must be managed.

V.2.4 Exercise 03

Exercise 3
Work directory: *ex*03/
Files to turn in: necessary files to make it work
Forbidden functions : None
Remarks : n/a

Archive files compatible with tar. We must be able to decompress an archive generated by your `ft_tar` with tar and vice versa.

V.2.5 Exercise 04

Exercise 4
Work directory: *ex*04/
Files to turn in: necessary files to make it work
Forbidden functions : None
Remarks : n/a

The (z) compression option. It must be done in the same way as the tar command; that is to say that this must be done on the fly. At no time should the archive be uncompressed on the disk.

Chapter VI

Bonus part

The other options of the tar.

Chapter VII

Turn-in and peer-evaluation

Turn your work in using your `GiT` repository, as usual.
Only work present in your repository will be graded during defense.