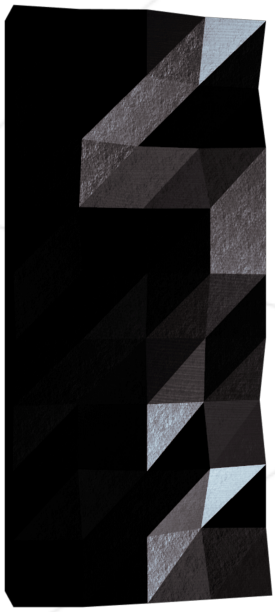

ft_ping

Summary: This project is about recoding the ping command.

Contents

Chapter I

Introduction

Ping is the name of a command that allows to test the accessibility of another machine through the IP network. The command measures also the time taken to receive a response, called round-trip time.

Chapter II

General Instructions

- This project will be corrected by humans only. You're allowed to organise and name your files as you see fit, but you must follow the following rules
- You must use C and submit a Makefile
- Your Makefile must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.
- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).
- Within the mandatory part, you are allowed to use the following functions:
  - getpid.
  - getuid.
  - getaddrinfo.
  - gettimeofday.
  - inet_ntop.
  - inet_pton.
  - exit.
  - signal.
  - alarm.
  - setsockopt.
  - recvmsg.
  - sendto.
  - socket.
  - printf and its family.
  - Your libft functions.
  - You are allowed to use other functions to complete the bonus part as long as their use is justified during your defense. Be smart.

ATTENTION: Usage of fcntl, poll et ppoll is strictly forbidden.

Chapter III

Mandatory Part

- The executable must be named ft_ping.
- You have to manage the -v -h options.

The -v option here will also allow us to see the results in case of a problem or an error linked to the packets, which logically shouldn't force the program to stop (the modification of the TTL value can help to force an error).

- You will have to manage a simple IPv4 (address/hostname) as parameters.
- You will have to manage FQDN without doing the DNS resolution in the packet return

You are allowed to use all the functions of the printf family as well as one global variable.

For the smarty pants (or not)... Obviously you are NOT allowed to call a real ping.

Chapter IV

Bonus Part

We will look at your bonuses if and only if your mandatory part is EXCELLENT. This means that your must complete the mandatory part, beginning to end, and your error management must be flawless, even in cases of twisted or bad usage. If that's not the case, your bonuses will be totally IGNORED.

Find below a few ideas of interesting bonuses:

- IPv6 management
- Additional -f -m -l -I -m -M -n -w -W -p -Q -S -t -T flags...

-V is not a bonus

Chapter V

Turn-in and peer evaluation

- Submit your work on your GiT repository as usual. Only the work on your repository will be graded.
- You have to be in a VM with a Linux kernel > 3.14. Note that grading was designed on a Debian 7.0 stable.