Minishell in C

In this project, we were taked with creating a simple shell program that uses the fork() API to process commands entered by the user in child processes created by fork(). The parent function is used to continuously ask the user for new commands.

## Authors

Zachary Essoh

Arielle Gilmore

API Reference

fork()

Parameters:

child\_pid == 0 - indicates if the program is inside the child process of the fork.

child\_pid > 0 - indicates if the program is inside of the parent process of the fork.

Function:

breaks a single process into 2 separate processes and creates a child process with its own process ID (PID)

execv()

Parameters:

file\_path - specifies the location of the commands within the operating system.

arg\_size - specifies the the length of the command the user is allowed to enter.

Function:

replaces the current process image with a new process image specified by a given path. In our case the path we needed to use was /bin/. So the child process image gets replaced with the user entered command that is in the bin folder, then the command is executed by the child process.

## Deployment

To run this program first cd into the directory of the file.

cd file directory location

In my case the file is located on the desktop

cd Desktop

Once in the proper directory type

gcc AZminishell.c -o AZminishell

to compile the program.

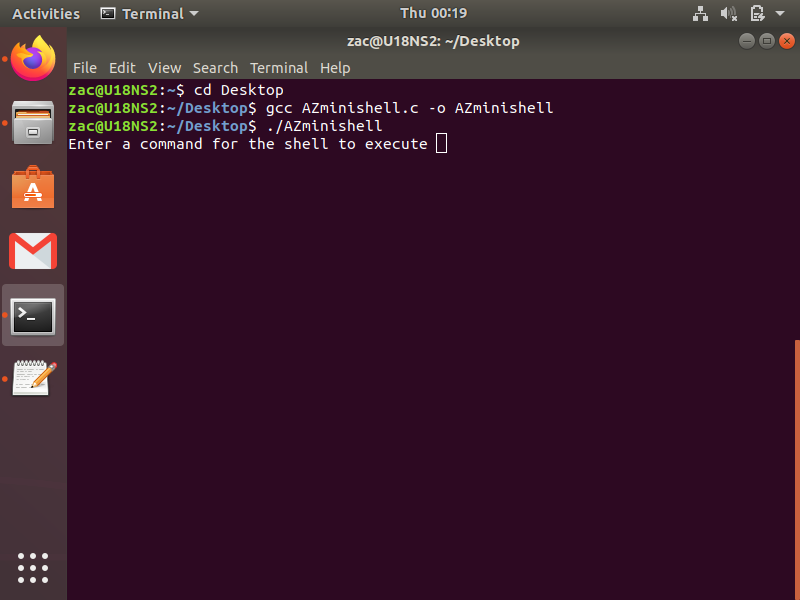
To execute the program type

./AZminishell

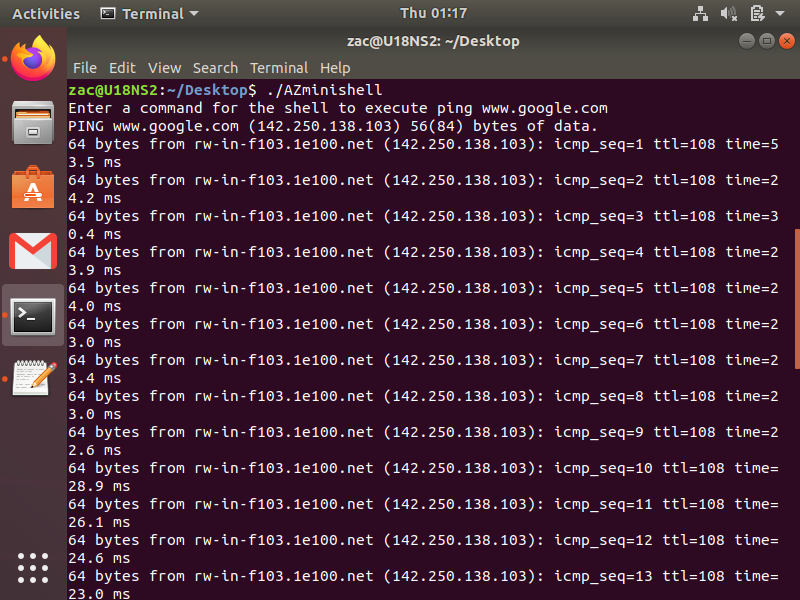
To exit the program press

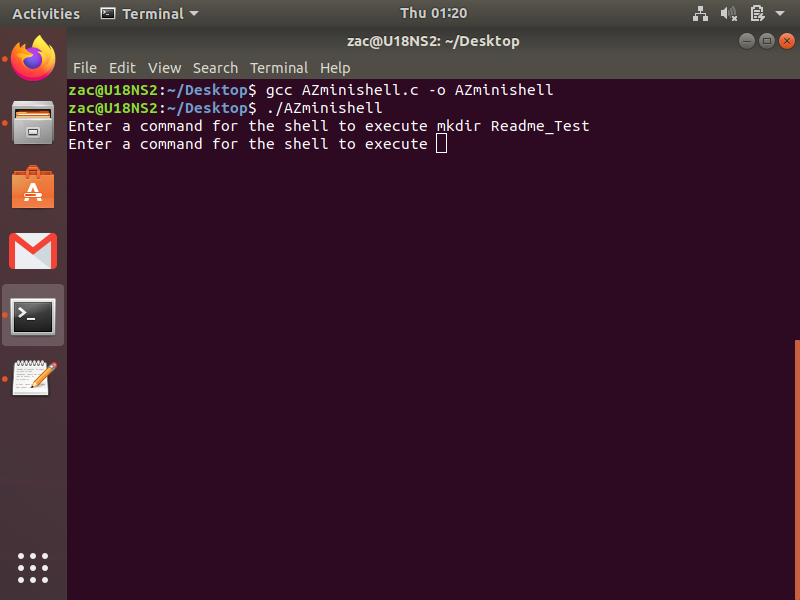
ctrl c

## Demo



## Screenshots







## Appendix

Any additional information goes here