Project 1: mysh

Introduction

My implementation of mysh is a basic command interpreter. It can generally execute any executable found in the user executing the binary's path. It also supports piping output from one executable to another for up to to 3 executables.

Basic user authentication is also provided. User passwords are hashed using Argon2 and a random salt seeded from `dev/urandom`. User data is stored inside the project root, under `etc/passwd`. Another file `etc/util` is used to keep track of the number of users registered.

List of Features

Feature Name	Command	Status
List files and directories	`ls`	Working
Grep	`grep`	Working
Print working directory	`pwd`	Working
Change directory	`cd [path]`	Working
Adding a user	`add_user`	Partially Working (only on init)
Login	-	Working
Exiting	`exit`	Working

Manual

Installation

1- Dependencies

This project depends on `libargon2`, a password hashing library. The source code of this library needs to be compiled before `mysh` is compile. To do that simply run `make dependencies` from the project root folder.

```
fish /Users/georgionicolas/academic/Spring-2020/Operating-Systems/terminal/terminal (fish)
~/a/S/O/terminal/terminal    fish    master - •    4%
~/a/S/O/t/terminal ❯❯❯ ls
Makefile        README.pdf      main.c
README.md       etc             project1(1).pdf
~/a/S/O/t/terminal ❯❯❯ make dependencies
Compiling dependency: libargon2Building with optimizations for native
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native  src/argon2.c src/core.c src/b
lake2/blake2b.c src/thread.c src/encoding.c src/opt.c src/run.c -o argon2
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native -dynamiclib -install_name @rpa
th/libargon2.1.dylib   src/argon2.c src/core.c src/blake2/blake2b.c src/thread.c src/encoding.
c src/opt.c -o libargon2.1.dylib
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native   -c -o src/argon2.o src/argon
2.c
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native   -c -o src/core.o src/core.c
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native   -c -o src/blake2/blake2b.o s
rc/blake2/blake2b.c
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native   -c -o src/thread.o src/threa
d.c
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native   -c -o src/encoding.o src/enc
oding.c
cc -std=c89 -O3 -Wall -g -Iinclude -Isrc -pthread -march=native   -c -o src/opt.o src/opt.c
ar rcs libargon2.a src/argon2.o src/core.o src/blake2/blake2b.o src/thread.o src/encoding.o sr
c/opt.o
sed '/^##.*$/d; s#@PREFIX@#/usr#g; s#@EXTRA_LIBS@##g; s#@UPSTREAM_VER@#ZERO#g; s#@HOST_MULTIAR
CH@#lib#g; s#@INCLUDE@#include#g;' < 'libargon2.pc.in' > 'libargon2.pc'
                    OK
~/a/S/O/t/terminal ❯❯❯
```

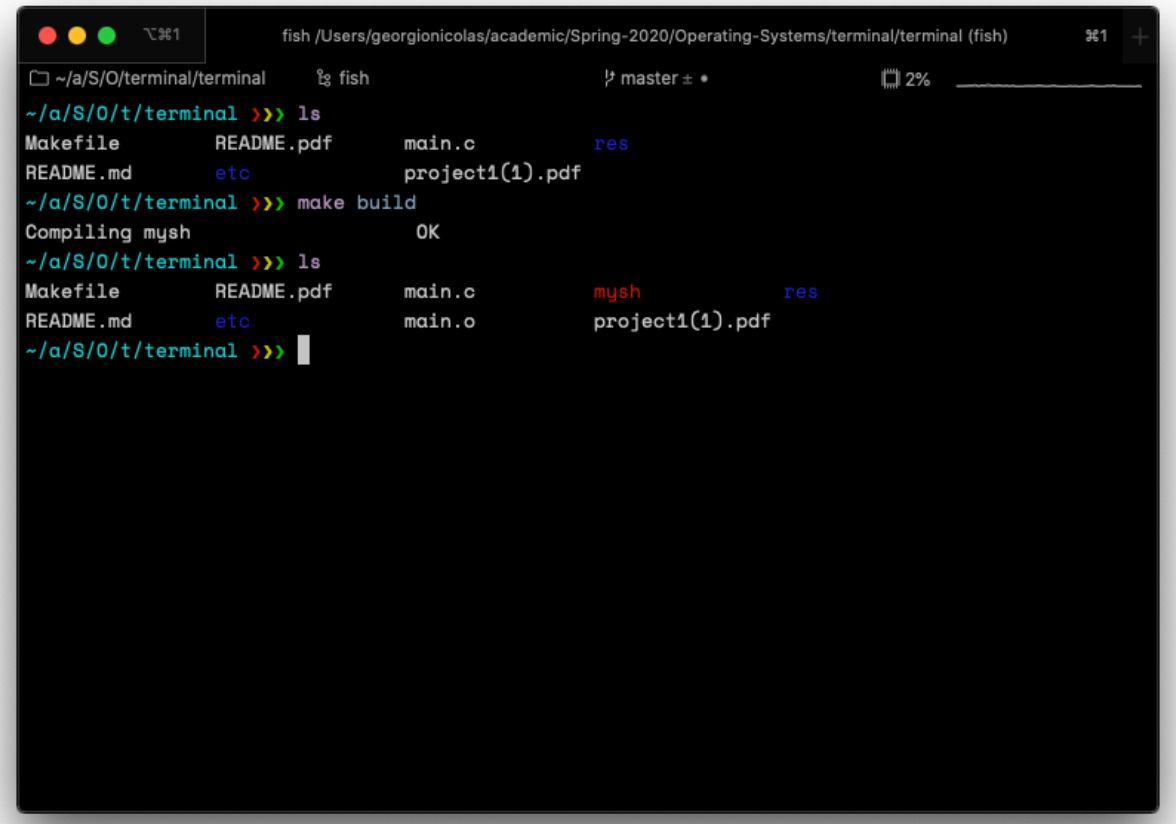
2- User Database Initialization

To initialize the database of user data, run `make init` from the project root folder. This delete any previously saved users. This command can be run at any given time should you wish to re-initialize the program or clear user data.

```
fish /Users/georgionicolas/academ...
~/a/S/O/t/t    fish    master ± •
~/a/S/O/t/terminal ❯❯❯ make init
Initializing                    OK
~/a/S/O/t/terminal ❯❯❯
```

3- Compiling The Project

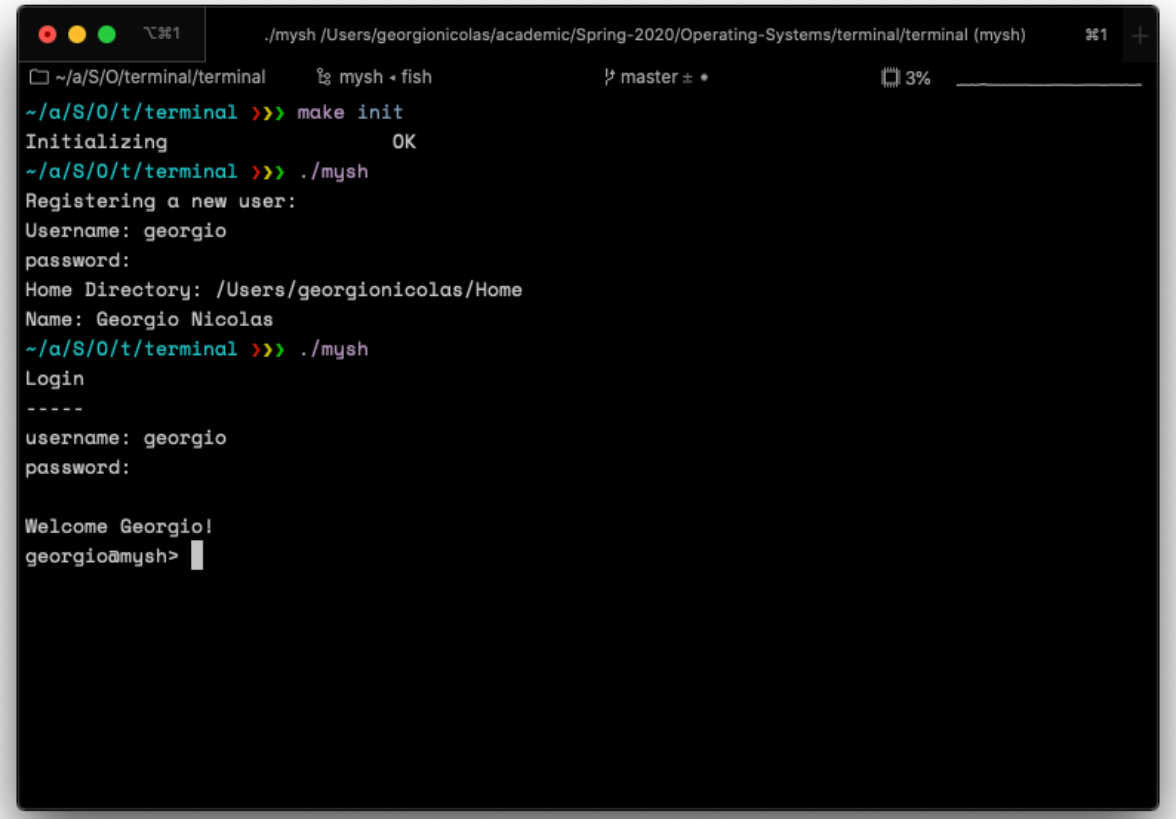
To compile this project, run `make build` in the project root directory. The outputs will be an intermediary binary `main.o`, and the main executable `mysh`. The project will be compiled using `gcc`.



4- Running The Executable

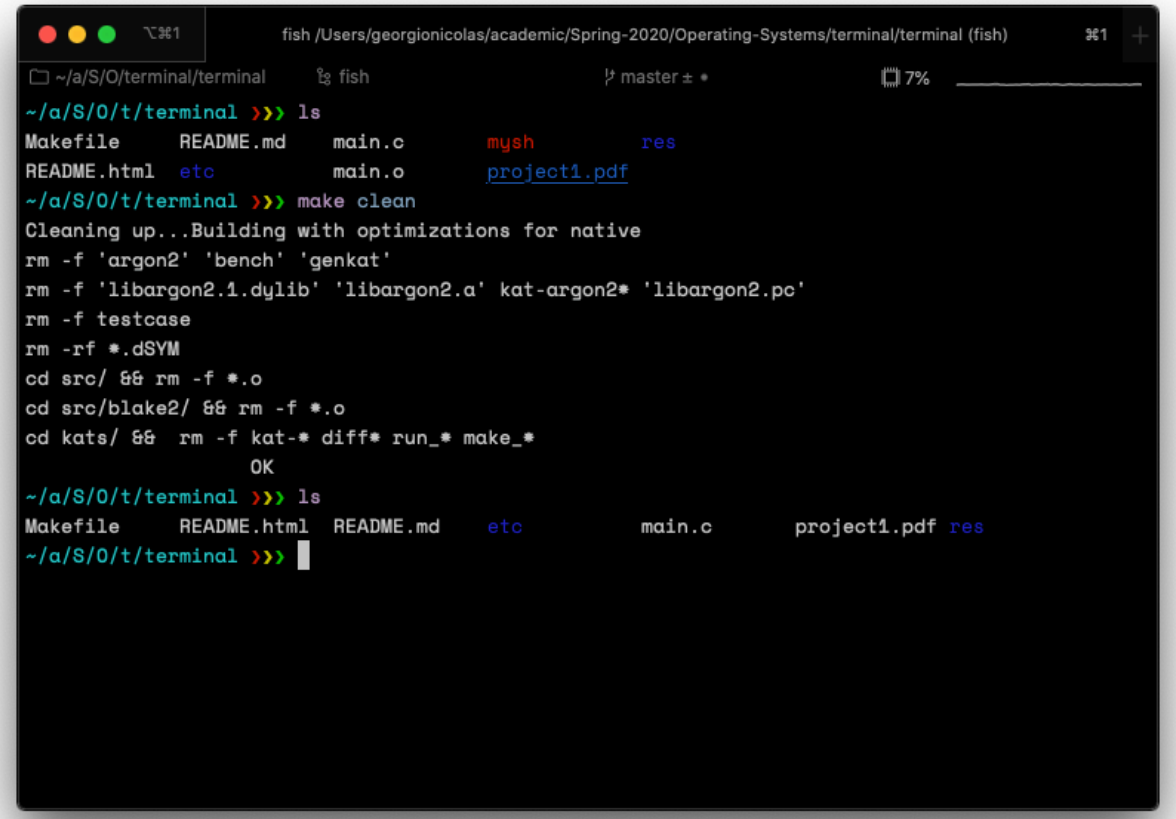
To run mysh, execute the following command from the project root folder `./mysh`.

When running for the first time after initialization, you will be prompted to register a user. Otherwise, you will be directed to the login page right away.



5- Cleaning Up

To remove the compiled files for `libagron2` and the project, simply run `make clean`. Note that the user database will remain persistent under `etc/passswd`. To remove user data refer to part 3 of the manual.



Correctness Testing

I have performed some tests which can be reproduced. The tests and results can be found in the screenshot below.

```
fish /Users/georgionicolas/academic/Spring-2020/Operating-Systems/terminal/terminal (fish)
~/a/S/O/terminal/terminal   fish   master ± •   3%
~/a/S/O/t/terminal ››› ./mysh
Login
-----
username: georgio
password:

Welcome Georgio!
georgio@mysh> cd res
georgio@mysh> ls
build.png              dependencies.png       run.png
clean.png              init.png
georgio@mysh> cd ..
georgio@mysh> ls
Makefile       README.pdf      main.c          mysh            res
README.md      etc             main.o          project1.pdf
georgio@mysh> pwd
/Users/georgionicolas/academic/Spring-2020/Operating-Systems/terminal/terminal
georgio@mysh> ls | grep README
README.md
README.pdf
georgio@mysh> exit
~/a/S/O/t/terminal ›››
```