

myshell

Introduction

This is a simple shell program written in C.

Usage

`make` to compile the program.

`./myshell` to run the program.

Alternatively `make run` to compile and run the program.

`make clean` to clean the directory from the executable, object, alias and history files.

Some Design Decisions and Specifications

- Use of `getcwd()` as `cwd` for the prompt string. This is done to make sure that the prompt string is always up to date.
- Aliases are stored in a file called `.aliases` in the same directory as the `myshell` executable. It is created from scratch every time the shell is run.
- If there's a conflict between an alias and a command, the alias is prioritized
- Last executed command is stored in a file called `.history` in the same directory as the `myshell` executable. It is created if it does not exist. It is overwritten if it exists.
- Last executed command resolves into a raw command from the user, including all the arguments. (i.e. input: `ls -l >> a.txt`, output: `ls -l >> a.txt`)
- Process count takes into account the zombie processes as well since they are not waited for.
- Background processing yields prompt string to be printed before the command is finished executing, similar to how `Bash` handles.
- `bello` functionality is provided as an executable file in the `/bin` with respect to `myshell` executable. After it is compiled within the same makefile, the directory `/bin` is added to the `PATH`. Therefore, whenever `bello` is called, there guaranteed to be at least 1 child process.
- Alias resolves into corresponding command and arguments right after getting the input from the user. (i.e. input: `lp -l`, alias: `lp = ls -a`, output: `ls -l -a`)
- `.local` is maintained in the `@hostname` part of the prompt string. No manual intervention was made to change it.

Difficulties Encountered

- Whether or not tokenize anything in double quotes as a single token/word, tokenizing in general.
- Background processing, zombie processes, and process count.
- Resolving aliases while tokenizing the input from the user.
- Handling `>>>` operation since it required another `fork()` call.

Variable Limitations (`#defined` MACROS)

- `MAX_LINE_LENGTH`: 1024
- `MAX_TOKENS`: 256
- `MAX_TOKEN_LENGTH`: 256
- `MAX_ALIASES`: 512
- `MAX_ARGUMENTS`: 256
- `MAX_PATH_LENGTH`: 1024
- `MAX_INPUT_LENGTH`: 1024

Ümit Can Evleksiz (2020400114)