# MINISHELL: as beautiful as a shell

## init_shell() & main()

~/.minishell_history
t_mish meta
print welcome badge
readline loop

→ read file history and add to readline history

→ struct with all the vars needed

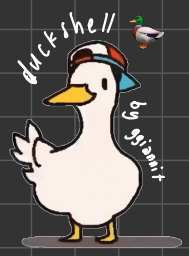
## handle_readline()

readline wait INPUT
fill history
parse INPUT
exec INPUT

## parse_line()

loop until pipe
split all words (POT)
split all redirects (RED)
create node with POT & RED
addfront() the node in CMD list
repeat for all 'pipe'

iter node and parse POT & RED
- remove extra space
- handle single and double quote
- expand variables




## redirects()

if more than one: the previous fd is closed, so only last fd_in and last fd_out are used

in a loop for all (RED)
if "<<"
  append until EOF
  in /tmp/.heredoc
if "<"
  open file READONLY
if ">>"
  append or create
  file and open
if ">"
  create and/or truncate
  file and open

the handling of ctrl+C is executed in a child process that will be killed if ctrl+C is pressed

## is_pipe_or_not()

1 node → exec all redirects (RED) → exec CMD (POT) → handle_cmd

2+ nodes → mini_pipe()

mini_pipe()
call recursively using pipe() and pass the node->next at each call until it's the last node. this way, it will execute CMDs in the right order

## handle_cmd

if it is a builtin
  call the function for it
if not
  search and exec the binary using $PATH

## exec_bin()

it's executed with the function execve() in a child process; if it exits with an error, the exit code is saved in $?

## BUILTIN

- cd → change directory can go back with the dash "-"
- history → print the history, "-c" clean the file
- echo → print on stdin, -n doesn't add \n
- unset → unset all variables passed as arguments
- pwd → print current dir, not with $PWD
- clear → clear page
- export → with no arguments, it prints ALL exported variables. All the args are set as exported variables (with the VALUE if the KEY already exists)
- env → prints all the exported variables, but only if they have a VALUE
- exit → quit the shell
- KEY=VALUE → this "command" sets a new VALUE if the KEY is not present, or modifies the VALUE of a variable
- ... → these are two easter eggs as memes, try finding them! :D

## example parse INPUT

INPUT
"<< EOF cat < infile | wc -l outfile"
OUTPUT (cmd list)

node 1
- POT: "wc", "-l", NULL
- RED: ">> outfile", NULL
- next -> node2

node 2
- POT: "cat", NULL
- RED: "<< EOF", "< infile", NULL
- next -> NULL

## handling of variables

the "envp" matrix is cloned using the heap memory, so the exported variables can be easily added with NO memory leaks. the non-exported variables and those exported with NO VALUE are saved in a linked list.

example

head
- KEY = example, VALUE = NULL, next → the variable is exported, but with NO VALUE (so far)
- KEY = foo, VALUE = bar, next → the variable is not exported
- KEY = NULL, VALUE = NULL, next → the variable has been unset