$ psh progress
week-2

Psh aims to make a POSIX like compliant shell capable of running all commands currently available with zsh and bash, while still attempting to make it as POSIX compliant as possible.

We also aim to add a few accessibility features such as syntax highlighting, auto complete, signal interruptions and the usage of arrow keys to browse through the history of commands entered.

By the end of week 5 our aim is to achieve all above.

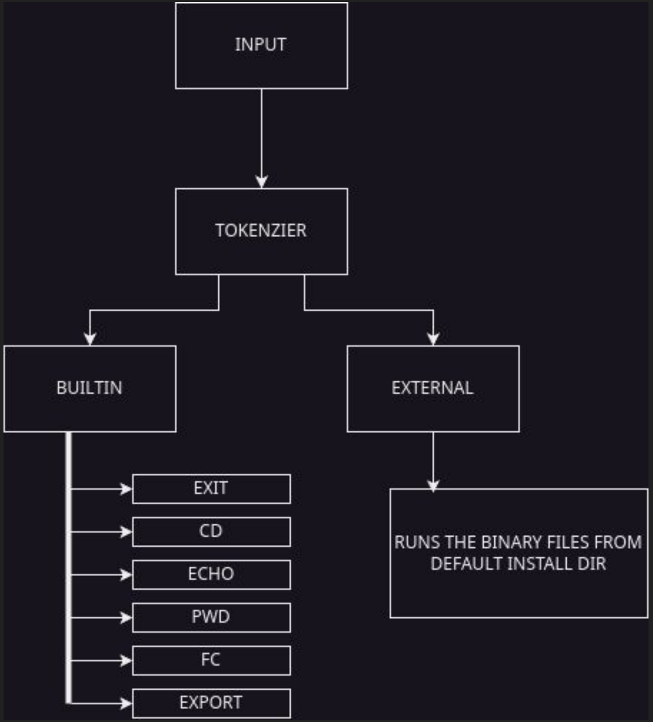
POSIX & POSIX like compliancy

A POSIX compliant shell follows the POSIX standards strictly, it does not allow you to have any other external features or enhancements to the shell itself, this allows immense portability but reduces functionality

A POSIX LIKE Shell follows the POSIX standards, but allows for additional features and behaviour not specified in the POSIX standards.

The aim of psh is to make a POSIX like compliant shell with the best of both worlds; portability as well as usability

WORKFLOW



Once the user enters a command ,The input is tokenized into separate arguments.
Ex:- ls -l -a

This will be stored as [ls,-l,-a]

These tokens are first checked for inbuilt and then passed into the functions as args.

🏁 Week -2 milestones

- refactor codebase into modular .c and .h files
- fc
- export
- pwd
- echo

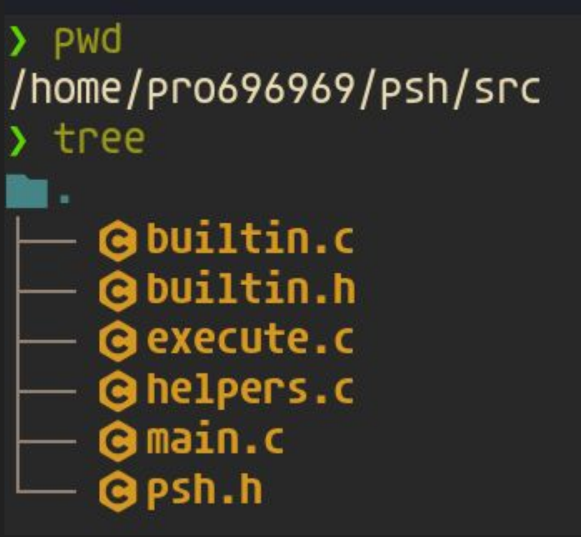
🐛 Bug bashing

- Following symlinks
- Rewriting history function
- segfaults

💡 KK : refactoring

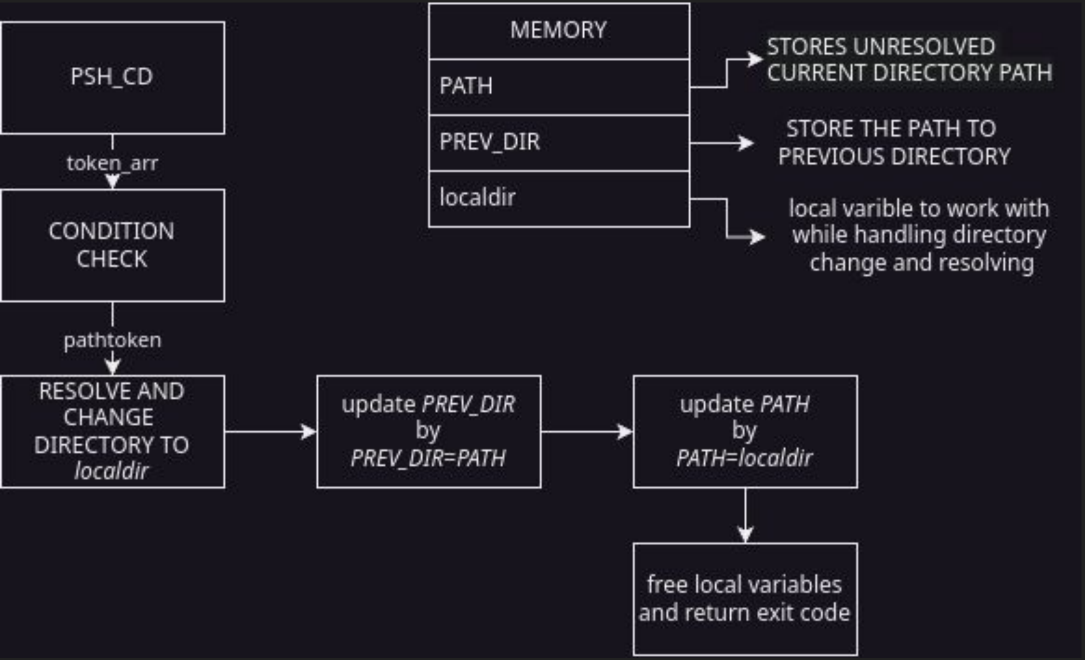
- Refactoring codebase
  - Makes it easier to read.
  - Code is now easily extensible and easy to work with.

- History file issue
- Inputting arrow keys (pushes 3 values into buffer)



```
❯ pwd
/home/pro696969/psh/src
❯ tree
.
├── builtin.c
├── builtin.h
├── execute.c
├── helpers.c
├── main.c
└── psh.h
```

💡 Adi:Symlinks


PSH_CD
token_arr
CONDITION CHECK
pathtoken
RESOLVE AND CHANGE DIRECTORY TO *localdir*
update *PREV_DIR* by *PREV_DIR=PATH*
update *PATH* by *PATH=localdir*
free local variables and return exit code
MEMORY
PATH
PREV_DIR
localdir
STORES UNRESOLVED CURRENT DIRECTORY PATH
STORE THE PATH TO PREVIOUS DIRECTORY
local varible to work with while handling directory change and resolving

- Symlinks are linux equivalent of windows shortcuts
- A Global variable PATH hold the path of current working directory
- The tokenized input is processed and then used change directory
- This local variable is then used to update the PATH and PREV_DIR

💡 Alayna:fc

- fc - allows you to view, edit and re-execute the commands previously entered to a shell without re-writing them again.

- history is not a posix-compliant command and hence it was replaced the fc command this week.

- The psh shell includes various flags for fc such as -l, -lr , -ln, -lnr, -e, -s and extra ones like -d, -c and -p.

- Dealt with file handling and executed all of the flags mentioned above.

- Dealt with several segmentation faults and handled some other error cases.

💡 Sid:echo

- echo : Displays text passed as arguments to it, or basically “echoes” your argument through the terminal :)
- There are also ways you can modify the output of the echo using flags such as :

    -n : Omits \n at the end of the output string.

    -e : Enables backslash escape interpretation such as \t, \b, etc.

- Your output can also be stored into a text file 🔥🔥.

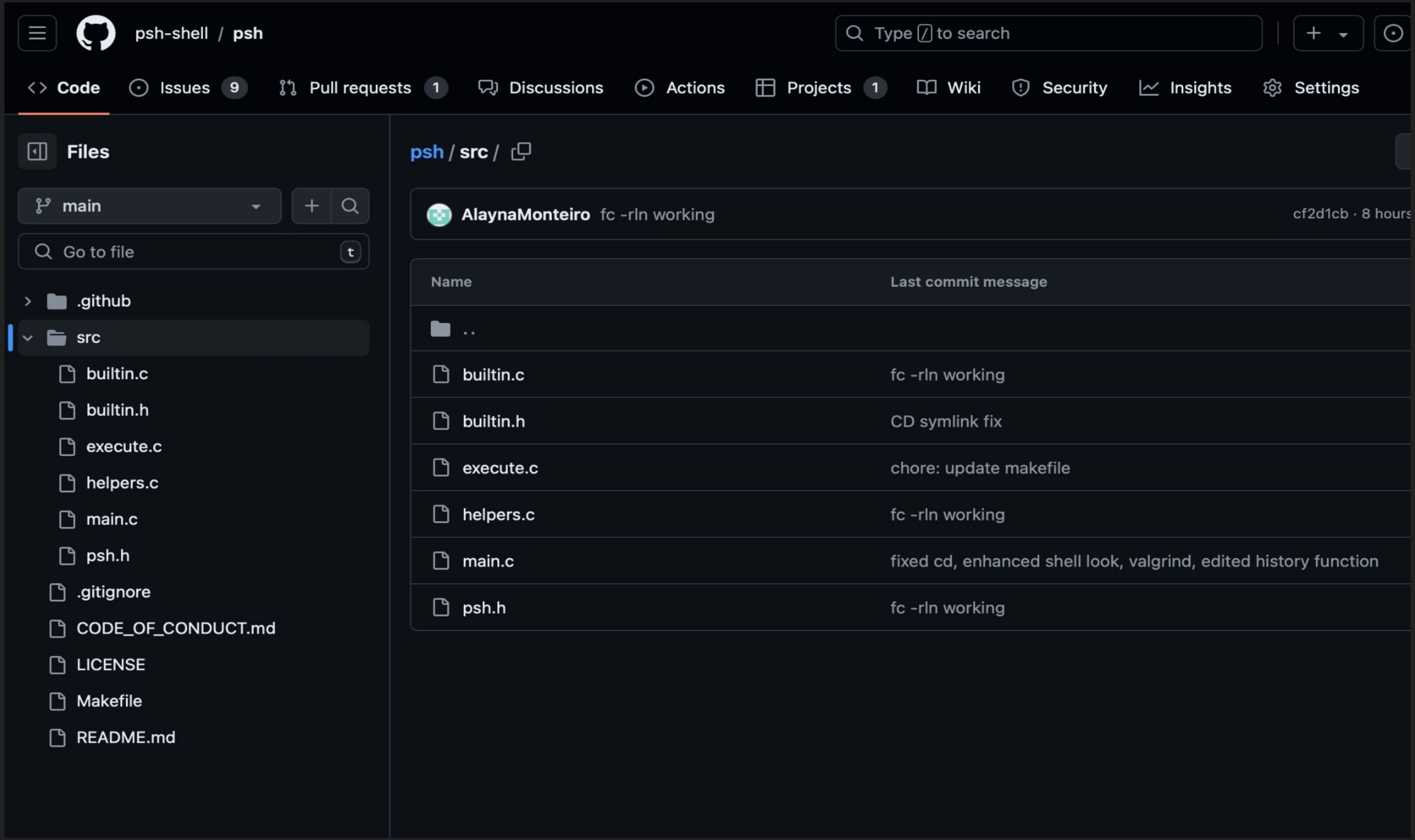
💡 Sumithra:export

- Used to mark any shell variables so that they are available to any child processes spawned by the shell.
- Usage

  ```
  $ export var_name=value
  ```

- Export options include:
  - export -p - Displays all exported variables along with values
  - export -n - Removes export attribute of the variable

DEMO



psh-shell / psh

Code | Issues 9 | Pull requests 1 | Discussions | Actions | Projects 1 | Wiki | Security | Insights | Settings

Files

main

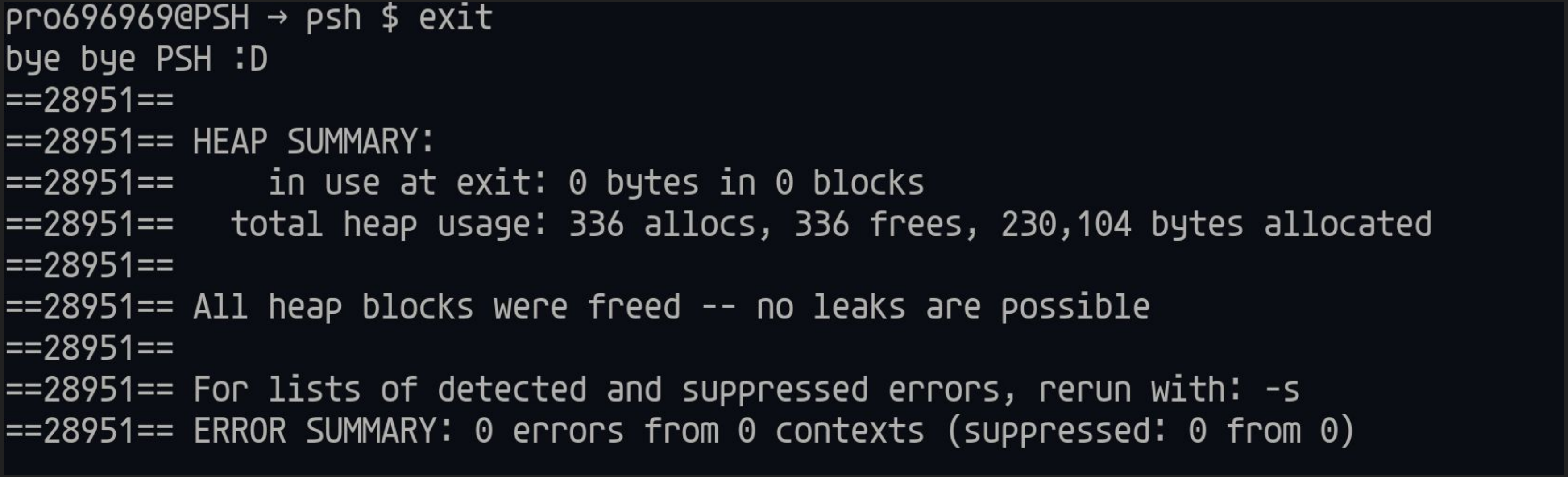
- .github
- src
  - builtin.c
  - builtin.h
  - execute.c
  - helpers.c
  - main.c
  - psh.h
- .gitignore
- CODE_OF_CONDUCT.md
- LICENSE
- Makefile
- README.md

psh / src /

AlaynaMonteiro fc -rln working cf2d1cb · 8 hours

Name	Last commit message
..	
builtin.c	fc -rln working
builtin.h	CD symlink fix
execute.c	chore: update makefile
helpers.c	fc -rln working
main.c	fixed cd, enhanced shell look, valgrind, edited history function
psh.h	fc -rln working

Memory Check



```
pro696969@PSH → psh $ exit
bye bye PSH :D
==28951==
==28951== HEAP SUMMARY:
==28951==     in use at exit: 0 bytes in 0 blocks
==28951==   total heap usage: 336 allocs, 336 frees, 230,104 bytes allocated
==28951==
==28951== All heap blocks were freed -- no leaks are possible
==28951==
==28951== For lists of detected and suppressed errors, rerun with: -s
==28951== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

🎯 Target for next weeks

- arrow keys
- completing all built in functions
- handling signals/interrupts
- looping constructs {for & while loops}
- shell expansion
- script mode: read and parse .sh files
- bug fixes !

✏️ feedback?