

Task list

[]

Dec 08.

- ☐ Fully understand “shift” and “simplify” operations in LR algorithm.

Dec 07.

- ☐ Init Minishell

Dec 06.

- ☐ Understand the tokenizing more, so we can design a data structure.
- ☐ Understand behavior of expander
 - ☐ \$...
- ☐ Parser: ... study
- ☐ Lexer: Find out all possible syntax errors.
- ☐ Schedule plan: when to make decisions, when to have prototypes, etc...

Dec 05.

- ☒ ~~Continue going through the bubbles.~~
- ☐ Understand the tokenizing more, so we can design a data structure.
- ☐

operator:

- |
- <
 - redirect input
- >
 - redirect output
- <<
 - here doc
- >>
 - append mode
- &&
 - AND
- ||
 - OR

token:

- def: string between 2 operators

- \$
- quotes:
 - “, ‘

token type:

- it will have different behavior by different types
 - >> APPEND
 - << HERE_DOC
 - < INPUT
 - > OUTPUT
 - && AND
 - || OR
 - WORD

special symbol:

- quotes
- operators
- \$
- \

LEXER:

- def: handle grammar
 - error and exit
 - empty string does nothing

Test cases:

Mandatory

- How to handle the input redirect operator, **it will impact how to split tokens**
 - echo 1 < out haha
 - echo 1 > out haha
 - echo 1 | < out cat
 - ANSWERS:
 - Positions of redirections don't matter. They can be anywhere between pipes.

- Both < and > always get executed, unless their parents node in the tree is an && or || - then it depends on the result of the sibling.
 - echo 1 | (echo 2 && cat && echo 4 > out > out2 echo A | echo 5) || echo 6
 - 2
 - 1
 - 5
 - The cat receives the pipe as input, bc echo 2 didn't read from it.
 - echo 1 | echo 2 && echo 3 && cat > out > out2 | echo 5 || echo 6 > out3
 - 2
 - 3
 - 5
 - hanging...
 - cat hangs, but the files still get created/overwritten.
- output redirect. **it will impact how to split tokens**
 - > out
 - ANSWER:
 - After < or > there always has to be a file.
 - So < / > and the word after it belong together.
 - If multiple redirections, bash first tests all files, and if any of them doesn't exist, it only displays error for this cmd.
 - However, if there's a heredoc inbetween, the files right of are separate from the ones before the heredoc.
 - Multiple redirections get evaluated from left to right, and every right one replaces the the previous one.
- print-order: Apparently all successful outputs come first. Error after, but error-order random.
 - Be careful with ft_printf, bc it uses write() bc we do not have mutex. Also no dprintf.
- Assignment:
 - a=1 echo \$a → <newline>
 - a=1 | echo \$a → <newline>
 - a=1 && echo \$a → 1
- Lexer:
 - Check if two operators connect to each other (whitespace ignored) - Error.
 - Handle odd amount of quotes. - Just error.
 - ~~Wait for matching pair.~~
 - ~~Escape character '\ escapes a special symbol.~~

- Lexer skips whitespace - it does nothing.
- Anything touching the opening quote before or the closing quote after belongs to the string.
 - Inside of quotes, whitespace gets kept.
- Here_doc:
 - cat << end " [ENTER]
" [ENTER]
end
 - << a << b
Do until a, then until b.
 - Signal interaction with here_doc:
 - Ctrl+C:
bash-5.1\$ cat << eof | echo 1
> ^C
 - Ctrl+D:
bash-5.1\$ cat << eof | echo 1
>
bash: warning: here-document at line 56 delimited by end-of-file
(wanted `eof`)
1
- Expander:
 - export A=12\$A
echo \$A -> 12
export A=12\$A
echo \$A -> 1212
 - export B=bbbb
echo aaaa\$B -> aaaa bbbb *(space before expansion)*
 - ">">' -> always expands to >>
- Builtins:
 - export:
 - export must only do something if there is no pipe before or after it.
 - exit:
 - exit | exit does not exit and \$? stays as it was.
 - exit | exit 42 does not exit and \$? is 42.
 - exit 42 | exit does not exit and \$? stays as it was.

Optional

- builtins:
 - cd:
 - autocomplete
 - Escape special characters in files and directories.

Notes:

- **FILES:**

```
int fd = open(path, O_RDWR|O_CREAT, 00644);
```
- **TOKEN:**

```
enum { int type
char *data }
```

Implementation:

General:

- If malloc fails anywhere, should we have a `clean_and_exit_shell` function and let shell be a global variable?

Lexer:

- data is a void * -> operator will be converted to int later, word to char *
- Operators should always be detected, unless they are in a quoted string. The touching rule of quotes does not apply here.
- `"" echo 1 -> bash: : command not found ... ""` should be a WORD that will be expanded to a nul-terminated string.

Heredoc:

- ~~Heredoc happens before expander.~~
- ~~After heredoc finished all of the heredoc input gets expanded.~~
- Wrong, [Bash Manual](#) describes it.

Executor:

- Builtin commands that are invoked as part of a pipeline are also executed in a subshell environment. [[Bash manual](#)]

Built-ins:

- export:
 - Difference between export and plain variable assignment:
<https://www.baeldung.com/linux/bash-variables-export>

Questions:

Lexer:

- ☒ export""G=e -> bash: exportG=e: command not found ... Should quotes be preserved in data?
Yes.