CS242-Assignment 2 MINISHELL

Vatsal Gupta
200101105

README

24th September 2021

- **Name:** Vatsal Gupta
- **Roll No:** 200101105
- **Programming Language Used:** C++
- **Operating System:** Linux (Ubuntu 20.04 LTS)

ZIP FILE CONTENTS

- 200101105_Assign02.cpp (Source Code)
- README(here)
- readme_plaintext

SPECIFICATIONS FOLLOWED FOR MINISHELL

- Works only on LINUX OS
- Written using C++ (POSIX based system) using g++ 9.3.0 for Ubuntu
- Flowchart as mentioned in FLOWCHART.pdf has been followed, and there are constraints in minishell's capability in line with this flowchart.
- MiniShell simulates bash and tries to include a reasonable subset of commands as per instructions in sir's mail.

KEY FEATURES INCLUDED:

- Environment variables:
  *printenv: lists all environment variables.
  *setenv(variable_name = variable_value): changes value of variable_name as per requirement.
  *Any particular environment variable can be seen from writing the name of that variable or writing echo $VAR where VAR is variable name.(shown later too.)
- Path functions: Most UNIX Commands are enterable.
- Parsing

- Piping
- Input and Output redirection:

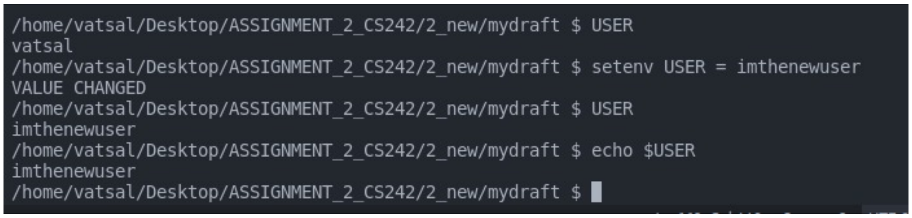
  Read mode, Write mode, Append mode

CONSTRAINTS FOR MINISHELL

- **Single Level Piping only** *(On recognising piping, the command is executed Directly without searching for redirection according to flowchart)*
- **Single Level Redirection Only** *(On recognising piping, the command is executed Directly without searching for redirection according to flowchart)*
- **No simultaneous Piping and Redirection***(Redirection is checked for only if piping is not present as per flowchart)*
- **Double Redirection is not allowed** *(in flowchart, further redirections are checked if and only if the previous specified redirection isn't found.)*
- Most of the following commands work flawlessly (internal+external), however due to limitations of C++ having access to the internal workings of an extended system,some natural functions which are internal to a shell but not to a C++ program may not work.Inspite of this, the following have been explicitly defined as internal commands to without using execvp and work perfectly:

  1. help
  2. cd
  3.setenv *
  4.exit (or quit or x)
  5.history
  6.printenv
  7.clear

  ***PLEASE NOTE:** setenv works in the following format:
  setenv variable_name = new_variable_value
  If this is successful, you should get a VALUE CHANGED prompt and the same can be checked by writing the variable name in the terminal which will output the value of the environment variable.

  Example: 

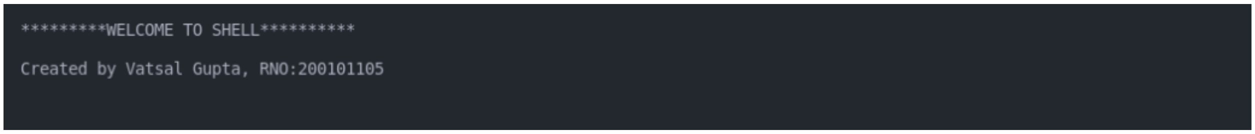
```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ USER
vatsal
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ setenv USER = imthenewuser
VALUE CHANGED
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ USER
imthenewuser
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ echo $USER
imthenewuser
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ █
```

INSTRUCTIONS TO RUN MINISHELL

- Compile with g++ 200101105_Assign02.cpp
- Run with ./a.out

OVERVIEW POINTS

- The Shell boots up with the following screen giving developer credits and information and welcoming user to the shell.

  ```
  *********WELCOME TO SHELL**********

  Created by Vatsal Gupta, RNO:200101105
  ```

  Which clears and starts the shell after a delay of 4(max) seconds. The shell takes in input and produces corresponding output, in tandem with the flowchart.
- The shell implements internal commands flawlessly and its details can be seen by typing help in the terminal. Most commands of UNIX work too.
- Main commands are :

  1.man (example man man) displays manual

  2.which

  3.chsh

  4.passwd

  5.date

  6.cal

  7.clear

  8.history

  9.sleep

  10.apropos

  11.exit

  12.shutdown

  13.ls

  14.cat

  15.more

  16.touch

  17.mv

  18.mkdir

  19.rmdir

  20.script

21.nano

22.find

23.touch

24.less

25.chmod

26.grep

27.printenv

28.getenv

- Shell exits with following prompt:

SPECIFICATIONS TO FOLLOW

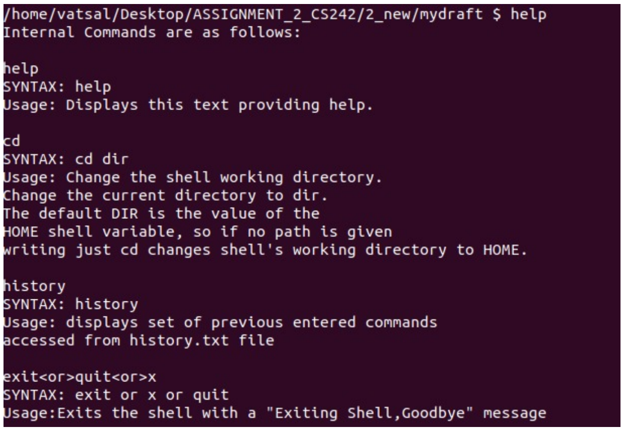
- Blank Lines are handled quite well, they don't get recorded in history.
- **Redirection** : MAKE SURE TO type both the command with no space before and after the redirection symbol (> , <, >>)
- Wrong commands give prompt to appropriate errors.
- setenv : type command as setenv (environment variable) = (new value)

  Make sure you use appropriate spacings.

- Exiting deletes history file(with exit or quit or x)

Examples of each step:

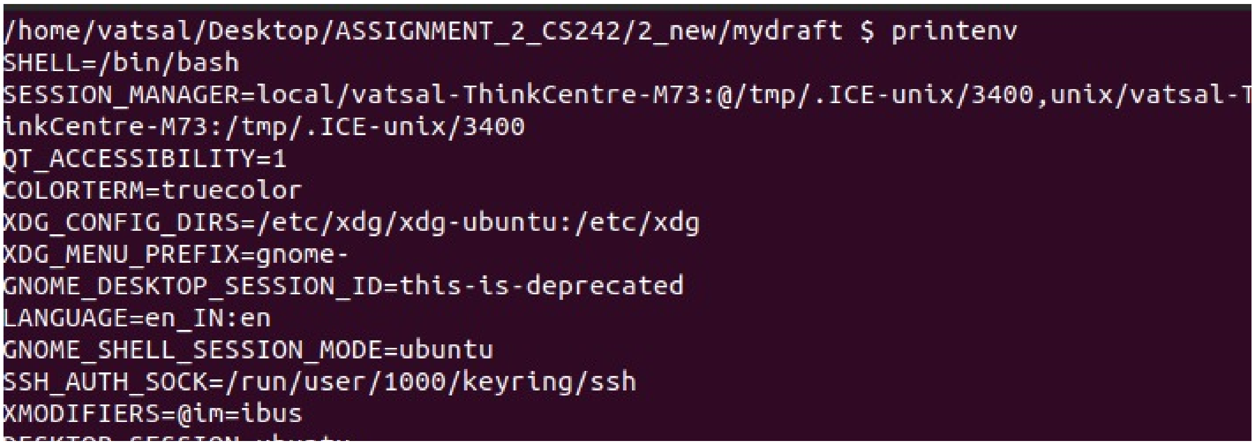
- Help



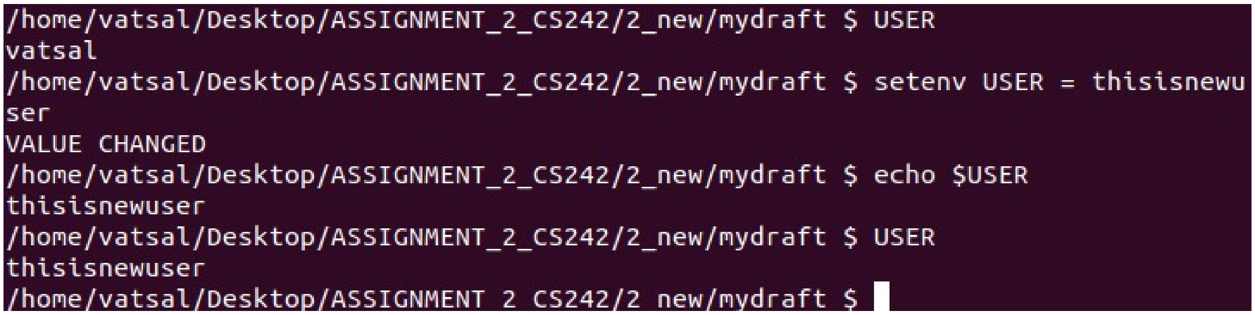
```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ help
Internal Commands are as follows:

help
SYNTAX: help
Usage: Displays this text providing help.

cd
SYNTAX: cd dir
Usage: Change the shell working directory.
Change the current directory to dir.
The default DIR is the value of the
HOME shell variable, so if no path is given
writing just cd changes shell's working directory to HOME.

history
SYNTAX: history
Usage: displays set of previous entered commands
accessed from history.txt file

exit<or>quit<or>x
SYNTAX: exit or x or quit
Usage:Exits the shell with a "Exiting Shell,Goodbye" message
```

- clear:(has to be seen by user of shell)

- Environment variables output (all kinds)

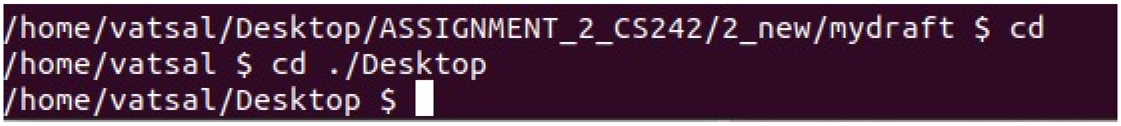


```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/vatsal-ThinkCentre-M73:@/tmp/.ICE-unix/3400,unix/vatsal-T
inkCentre-M73:/tmp/.ICE-unix/3400
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_IN:en
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
```



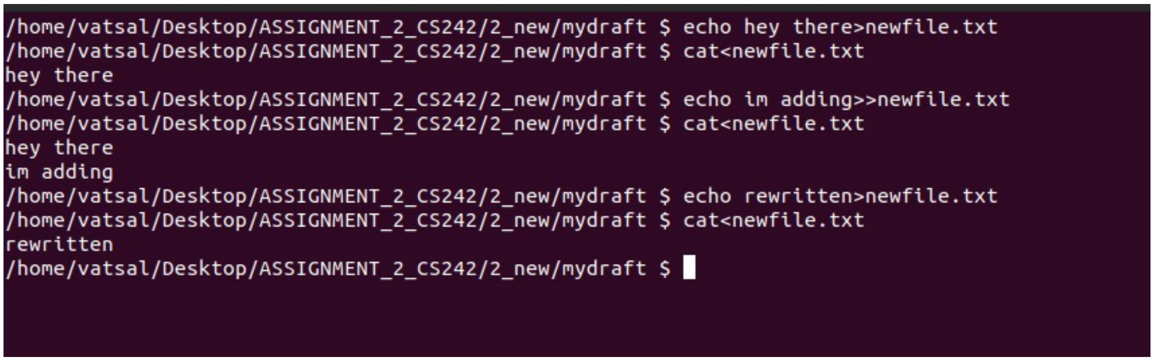
```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ USER
vatsal
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ setenv USER = thisisnewu
ser
VALUE CHANGED
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ echo $USER
thisisnewuser
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ USER
thisisnewuser
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $
```

- Only cd and cd with a different directory



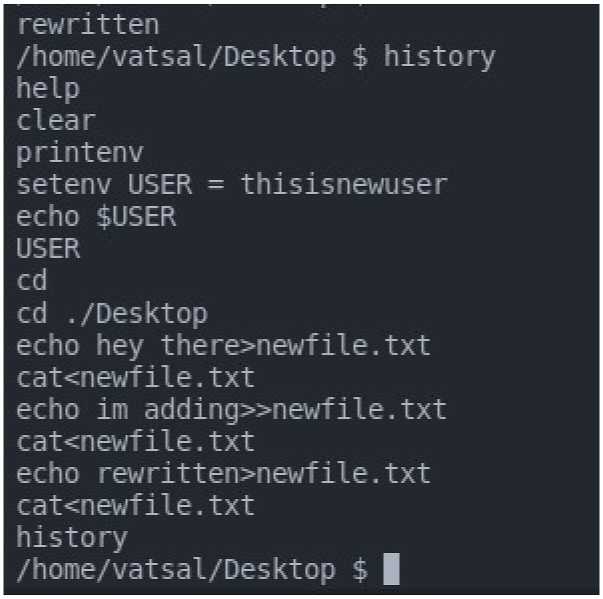
```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ cd
/home/vatsal $ cd ./Desktop
/home/vatsal/Desktop $
```

- Input redirection,output redirection(write and append)



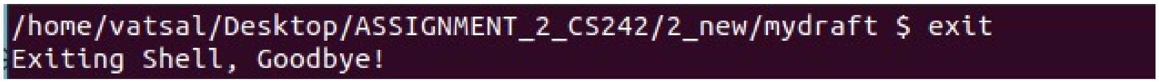
```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ echo hey there>newfile.txt
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ cat<newfile.txt
hey there
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ echo im adding>>newfile.txt
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ cat<newfile.txt
hey there
im adding
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ echo rewritten>newfile.txt
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ cat<newfile.txt
rewritten
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $
```

- History (the file is created in user’s tmp folder as per flowchart which MUST be present in all UNIX based systems)



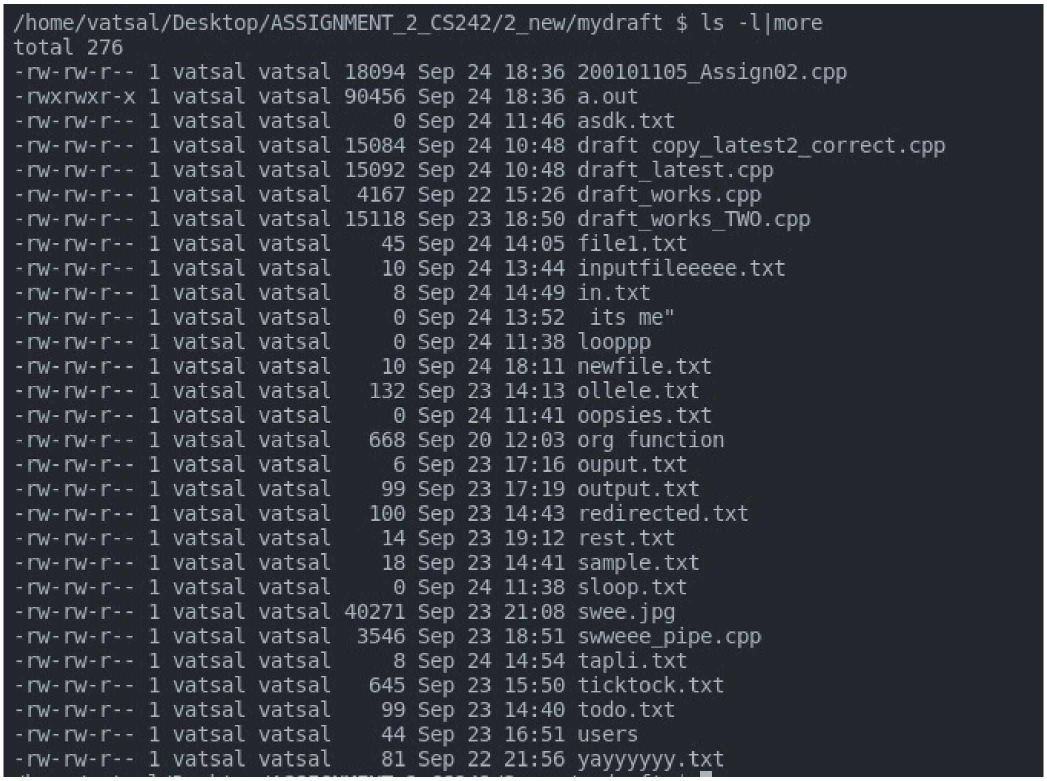
```
rewritten
/home/vatsal/Desktop $ history
help
clear
printenv
setenv USER = thisisnewuser
echo $USER
USER
cd
cd ./Desktop
echo hey there>newfile.txt
cat<newfile.txt
echo im adding>>newfile.txt
cat<newfile.txt
echo rewritten>newfile.txt
cat<newfile.txt
history
/home/vatsal/Desktop $
```

- exit



```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ exit
Exiting Shell, Goodbye!
```

* Piping example:



```
/home/vatsal/Desktop/ASSIGNMENT_2_CS242/2_new/mydraft $ ls -l|more
total 276
-rw-rw-r-- 1 vatsal vatsal 18094 Sep 24 18:36 200101105_Assign02.cpp
-rwxrwxr-x 1 vatsal vatsal 90456 Sep 24 18:36 a.out
-rw-rw-r-- 1 vatsal vatsal     0 Sep 24 11:46 asdk.txt
-rw-rw-r-- 1 vatsal vatsal 15084 Sep 24 10:48 draft copy_latest2_correct.cpp
-rw-rw-r-- 1 vatsal vatsal 15092 Sep 24 10:48 draft_latest.cpp
-rw-rw-r-- 1 vatsal vatsal  4167 Sep 22 15:26 draft_works.cpp
-rw-rw-r-- 1 vatsal vatsal 15118 Sep 23 18:50 draft_works_TWO.cpp
-rw-rw-r-- 1 vatsal vatsal    45 Sep 24 14:05 file1.txt
-rw-rw-r-- 1 vatsal vatsal    10 Sep 24 13:44 inputfileeeee.txt
-rw-rw-r-- 1 vatsal vatsal     8 Sep 24 14:49 in.txt
-rw-rw-r-- 1 vatsal vatsal     0 Sep 24 13:52  its me"
-rw-rw-r-- 1 vatsal vatsal     0 Sep 24 11:38 looppp
-rw-rw-r-- 1 vatsal vatsal    10 Sep 24 18:11 newfile.txt
-rw-rw-r-- 1 vatsal vatsal   132 Sep 23 14:13 ollele.txt
-rw-rw-r-- 1 vatsal vatsal     0 Sep 24 11:41 oopsies.txt
-rw-rw-r-- 1 vatsal vatsal   668 Sep 20 12:03 org function
-rw-rw-r-- 1 vatsal vatsal     6 Sep 23 17:16 ouput.txt
-rw-rw-r-- 1 vatsal vatsal    99 Sep 23 17:19 output.txt
-rw-rw-r-- 1 vatsal vatsal   100 Sep 23 14:43 redirected.txt
-rw-rw-r-- 1 vatsal vatsal    14 Sep 23 19:12 rest.txt
-rw-rw-r-- 1 vatsal vatsal    18 Sep 23 14:41 sample.txt
-rw-rw-r-- 1 vatsal vatsal     0 Sep 24 11:38 sloop.txt
-rw-rw-r-- 1 vatsal vatsal 40271 Sep 23 21:08 swee.jpg
-rw-rw-r-- 1 vatsal vatsal  3546 Sep 23 18:51 swweee_pipe.cpp
-rw-rw-r-- 1 vatsal vatsal     8 Sep 24 14:54 tapli.txt
-rw-rw-r-- 1 vatsal vatsal   645 Sep 23 15:50 ticktock.txt
-rw-rw-r-- 1 vatsal vatsal    99 Sep 23 14:40 todo.txt
-rw-rw-r-- 1 vatsal vatsal    44 Sep 23 16:51 users
-rw-rw-r-- 1 vatsal vatsal    81 Sep 22 21:56 yayyyyyy.txt
```