# REPORT 2018021
# ANUNAY YADAV

input :/bin/ls | /usr/bin/sort | uniq

working :

main function :

1. getcwd gets the current working directory to give the user idea of which directory is he on.
2. tmp contains the command with pipes
3. tmp (containing the command) is parsed first by pipe character and then by space character
4. each token with pipe character as seperator is stored in a struct pipe_command which contains all the commands
5. each command in pipe_command is stored in struct command which tells the output file and input file of the command and do we have to append or not and what command are stored in it and arguments and the number of arguments.
6.parsing first takes place in parse_pipe which gets all the tokens "|" seperated and then parses those tokens individually with parse_new_cmd. All these tokens are stored in a pipe struct.
7. parse_new cmd calls parse which with parsing checks for redirection and updates output and input file descriptor. And removes those tokens.

In every if condition first previous files are closed and new ones are created .

First if checks if redirection '> filename' : is used

if it is it updates the outputfile to the current filename after the getting file descriptor and closes the previous one .

Previous files are closed. And creat is used to creat new fd

Second '>>' : updates the output but opens the output file with append mode

as >> adds to end of file. First checks if the file exists or not with the help. of access if it does than open in append mode else creat new. Open and creat are used accordingly.

Third '1<' : updates output file with the mentioned file. Checks for error and updates .

4Rth ' <' : updates input file but checks wether file exists or not

5th '2 < ' : updates error file to this filename .

6Th '2<&1' : redirects error to output by giving error output fd.

8. pipes are created . For piping between all the commands . Number of pipes is stored in pipe_command struct.

9. first command is alloted 0 as stdin and its output to pip[1]. And than iteratively it gives pipe[0] to next command which is reading from the command before allocated with pipe[1]. End command is given output file = 1 so that ouput is displayed on the terminal screen.

10. now execution takes place.

Every command is executed iteratively

first it is checked if the command has any error or not . Next cd is handled as it not in /bin next exit is checked to exit from the shell.

11. child process is created

in child process :

if current command output is different from the default one dup2 makes the curr command ouput the standard output.

Same is done for stdin,stderr,stdout.

Command is executed with command name and command arguments given as perimeters. Execvp is used which takes care of the path and finds thye path itself.

Parent process:

waits for child process

closes all unnessecary file descriptors this command has produced.

Steps :

ls is found in bin and executed with default output and input files. And piping in step 9 takes care of piping and ls output is piped to next command.

Sort is found and output in ls is redirected here using pipe and sort is executed and output is redirected to the next pipe using redirection.

Uniq is foind in bin and output of sort is given to it using pipe and as this is the end command output of uniq is redirected standard output screen.