Name: Mohammed Hassan

ID: 120170672

Shell project in C programming

DR : Ibrahim Abo Haiba

1. The shell must support the following internal commands:

i. `cd <directory>`—Change the current default directory to `<directory>`.

If the `<directory>` argument is not present, report the current directory. If the directory does not exist, an appropriate error should be reported. This command should also change the `PWD` environment variable.

ii. `clr`—Clear the screen.

iii. `dir <directory>`—List the contents of directory `<directory>`.

iv. `environ`—List all the environment strings.

v. `echo <comment>`—Display `<comment>` on the display followed by a new line (multiple spaces/tabs may be reduced to a single space).

vi. `help`—Display the user manual using the `more` filter.

vii. `pause`—Pause operation of the shell until "`Enter`" is pressed.

viii. `quit`—Quit the shell.

ix. The shell environment should contain `shell=<pathname>/myshell` where `<pathname>/myshell` is the full path for the shell executable (not a hardwired path back to your directory, but the one from which it was executed).

2. All other command line input is interpreted as program invocation, which should be done by the shell `fork`ing and `exec`ing the programs as its own child processes. The programs should be executed with an environment that contains the entry: `parent=<pathname>/myshell` where `<pathname>/myshell` is as described in 1.ix above.

3. The shell must be able to take its command line input from a file. That is, if the shell is invoked with a command line argument:

`myshell batchfile`
then `batchfile` is assumed to contain a set of command lines for the shell to process. When the end-of-file is reached, the shell should exit. Obviously, if the shell is invoked without a command line argument, it solicits input from the user via a prompt on the display.

4. The shell must support I/O redirection on either or both *stdin* and/or *stdout*. That is, the command line

`programname arg1 arg2 < inputfile > outputfile`

will execute the program `programname` with arguments `arg1` and `arg2`, the *stdin FILE stream* replaced by `inputfile` and the *stdout FILE stream* replaced by `outputfile`.

Project Requirements **PP1-3**

`stdout` redirection should also be possible for the internal commands `dir`, `environ`, `echo`, and `help`.

With output redirection, if the redirection character is `>` then the `outputfile` is created if it does not exist, and truncated if it does. If the redirection token is `>>` then `outputfile` is created if it does not exist, and appended to if it does.

5. The shell must support background execution of programs. An ampersand (`&`) at the end of the command line indicates that the shell should return to the command line prompt immediately after launching that program.

6. The command line prompt must contain the pathname of the current directory.

Run the command

```
Mohammed -SHELL:/home/mohammed$ ls
aaa      circle.dat  Downloads  labos     Pictures  tempdir    vdfa
adasd    DD.png      ha         mohammed  proj      Templates  Videos
a.out    Desktop     hassan     Music     Quiz_1    Test
backup   Documents   khiri      nan.mp4   snap      Test.c
Mohammed -SHELL:/home/mohammed$ ls -a
.               Downloads  snap
..              .gnupg     .ssh
aaa             ha         .sudo_as_admin_successful
adasd           hassan     tempdir
a.out           khiri      Templates
backup          labos      Test
.bash_history   .local     Test.c
.bash_logout    mohammed   .thunderbird
.bashrc         .mozilla   .vboxclient-clipboard.pid
.cache          Music      .vboxclient-display-svga-x11.pid
circle.dat      nan.mp4    .vboxclient-draganddrop.pid
.config         Pictures   .vboxclient-seamless.pid
DD.png          .profile   vdfa
Desktop         proj       Videos
Documents       Quiz_1
Mohammed -SHELL:/home/mohammed$
```

```
Mohammed -SHELL:/home/mohammed$ mkdir mohammed_hassan
Mohammed -SHELL:/home/mohammed$ cat Test.txt
sadasdasdasdsadasd
Mohammed -SHELL:/home/mohammed$ cd mohammed_hassan
Mohammed -SHELL:/home/mohammed/mohammed_hassan$
```

```
PIPE(2)                    Linux Programmer's Manual                    PIPE(2)

NAME
       pipe, pipe2 - create pipe

SYNOPSIS
       #include <unistd.h>

       /* On Alpha, IA-64, MIPS, SuperH, and SPARC/SPARC64; see NOTES */
       struct fd_pair {
           long fd[2];
       };
       struct fd_pair pipe();

       /* On all other architectures */
       int pipe(int pipefd[2]);

       #define _GNU_SOURCE             /* See feature_test_macros(7) */
       #include <fcntl.h>              /* Obtain O_* constant definitions */
       #include <unistd.h>

       int pipe2(int pipefd[2], int flags);

 Manual page pipe(2) line 1 (press h for help or q to quit)
```

```
Mohammed -SHELL:/home/mohammed/mohammed_hassan$ clear
```

```
Mohammed -SHELL:/home/mohammed$ help

This is Mohammed_Hassan Shell.......HELLO:)
Type program names and arguments, and hit enter.
The following are built in commands:
  cd
  help
  exit
  pause
  environ
Use the man command for information on other programs.
Mohammed -SHELL:/home/mohammed$
```

```
Mohammed -SHELL:/home/mohammed$ pause
PRESS [ENTER] TO CONTINUE
Mohammed -SHELL:/home/mohammed$
```

```
Mohammed -SHELL:/home/mohammed$ exit
mohammed@mohammed-VirtualBox:~$
```

```
Mohammed -SHELL:/home/mohammed$ echo Mohammed Hassan Hacker
Mohammed Hassan Hacker
Mohammed -SHELL:/home/mohammed$
```

```
Mohammed -SHELL:/home/mohammed$ echo Mohammed Hassan Hacker
Mohammed Hassan Hacker
Mohammed -SHELL:/home/mohammed$ environ
SHELL=/bin/bash
SESSION_MANAGER=local/mohammed-VirtualBox:@/tmp/.ICE-unix/1241,unix/mohammed-Vi
rtualBox:/tmp/.ICE-unix/1241
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
GTK_IM_MODULE=ibus
QT4_IM_MODULE=ibus
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1131
GTK_MODULES=gail:atk-bridge
PWD=/home/mohammed
LOGNAME=mohammed
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
```

Also in my code I use the function

Strtok you can see that in my code

```
//strtok is used to Seperate a string
char* command_token = strtok (*command, " ");
//the count of arguments
int argscounter = 0;
```

```
    ,,
    command_token= strtok(NULL, " ");
    //END OF THE SPLITING THE INPUTS INTO
IMENTS LOOP
    }

    //Setting the Last argument in args to NULL
    args[argscounter] = NULL;
```

```
Mohammed -SHELL:/home/mohammed$ dir
aaa     circle.dat  Downloads  labos           nan.mp4   snap       Test.c
adasd   DD.png      ha         mohammed        Pictures  tempdir    Test.txt
a.out   Desktop     hassan     mohammed_hassan proj      Templates  vdfa
backup  Documents   khiri      Music           Quiz_1    Test       Videos
Mohammed -SHELL:/home/mohammed$
```