Report for OS Assignment1

Mehar Khurana

2021541

Program specifications:

- Programmed on an Ubuntu 20.04 LTS system
- Taking input through stdin
- Input parameters are separated by spaces
- Internal commands are executed by the shell itself and external commands are run by either fork() and exec() or by creating a new thread with functions from <pthread.h>
- All external commands are separate .c files, which are compiled on executing make in the terminal
- Execution of make creates an executable file shell, which can be run by typing ./shell in the terminal

Commands Implemented:

All internal commands follow graceful degradation of arguments.

1. cd

   Options implemented: cd - and cd ~

   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ cd ..
   mehar@len-y520:/home/mehar/Desktop>>
   $ cd -
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ cd ~
   mehar@len-y520:/home/mehar>>
   $ cd -
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   ```

```
$ cd ..
mehar@len-y520:/home/mehar/Desktop>>
$ cd vs-codes
mehar@len-y520:/home/mehar/Desktop/vs-codes>>
$ cd -
mehar@len-y520:/home/mehar/Desktop>>
```

Errors/Edge cases handled:

1. Edge case where no argument is given:
   ```
   mehar@len-y520:/home/mehar/Desktop>>
   $ cd
   mehar@len-y520:/>>
   ```
2. In case more than 2 arguments are provided, we omit the extra arguments (graceful degradation):
   ```
   mehar@len-y520:/home/mehar/Desktop>>
   $ cd vs-codes os-repo
   mehar@len-y520:/home/mehar/Desktop/vs-codes>>
   ```

2. echo

Options implemented: -n (omits newline at the end) and –help (displays the manpage for echo)

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ echo

mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ echo hello bye
hello bye
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ echo -n hello bye
hello bye mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ echo --help
Usage: echo [-n] [--help] [<string>/$ENV]
mehar@len-y520:/home/mehar/Desktop/os-repo>>
```

Errors/Edge cases handled:

1. Edge case where no argument is given
2. In case user wants an environment variable, they can access it as usual using the $ prefix

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi
n:/bin:/usr/games:/usr/local/games:/snap/bin
```

3. pwd

Assumption: the default behaviour for pwd was taken as -L

Options implemented: -L (to get symbolic path) and -P (to get real path)

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ pwd
/home/mehar/Desktop/os-repo
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ pwd -L
/home/mehar/Desktop/os-repo
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ pwd -P
This source is at /home/mehar/Desktop/os-repo.
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ pwd -X
Incorrect use of the command, pwd.
mehar@len-y520:/home/mehar/Desktop/os-repo>>
```

Errors/Edge cases handled:

1. If an incorrect flag is given, the command is not executed.

2. In case more than 2 tokens are provided, we omit the extra tokens (graceful degradation):

External Commands

For all external commands, interchange of option order and use of multiple options in the same command have been handled as corner cases. Examples of the corner cases and other testcases have been given along with each command.

1. date
   Options implemented: -u (to display UTC time) and -r (prints the last time a file was modified)
   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ date
   Local Time: 2022-10-31 22:54:10
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ date -u
   UTC: 2022-10-31 17:24:14
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ date -r shell.c
   File 'shell.c' last modified on: Mon Oct 31
   21:57:39 2022
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   ```

   Errors/Edge cases handled:
   1. In case more than 2 tokens are provided, we omit the extra tokens (graceful degradation):
      ```
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      $ date -u hello
      UTC: 2022-10-31 17:25:54
      ```
   2. Handled the case where incorrect flags are specified.
      ```
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      $ date -x
      Incorrect use of the command, date.
      ```

2. ls

   Options implemented: -a (shows all files and directories, including hidden) and -i (shows the serial number of the file or directory in a list)

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ ls -a
rm              2021541_Assignment0             shell.c         Makefile                mkdir.c
        date.c          mkdir           rm.c            ls              cat             shell
        cat.c           date            .git            ls.c            .DS_Store       ...

mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ ls -i
11817505        rm
11816271        2021541_Assignment0
11816280        shell.c
11816274        Makefile
11816278        mkdir.c
11816276        date.c
11817503        mkdir
11816279        rm.c
11817502        ls
11817458        cat
11817508        shell
11816275        cat.c
11817507        date
11816277        ls.c
```

   Errors/Edge cases handled:

   1. In case a non-existent directory is specified in the path:

      ```
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      $ ls hello/
      Error: this directory does not exist.
      ```

   2. Both -a and -i can be used simultaneously:

      ```
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      $ ls -ai
      11817505   rm
      11816271   2021541_Assignment0
      11816280   shell.c
      11816274   Makefile
      11816278   mkdir.c
      11816276   date.c
      11817503   mkdir
      11816279   rm.c
      11817502   ls
      ```

```
11817458   cat
11817508   shell
11816275   cat.c
11817507   date
11816143   .git
11816277   ls.c
11816270   .DS_Store
11816007   .
11796527   ..
mehar@len-y520:/home/mehar/Desktop/os-repo>>
```

3. rm

   Options implemented: -v (shows what is being deleted) and -f (ignores files that do not exist and have been specified in the command)

   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ rm one.txt
   Successful.
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ rm -v two.txt
   Removed 'two.txt'
   Successful.
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ rm -f three.txt
   Successful.
   ```

   Errors/Edge cases handled:

   1. User cannot do `rm .` or `rm ..`:

      ```
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      $ rm .
      Could not remove '.'
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      $ rm ..
      Could not remove '..’
      ```

   2. Invalid command of insufficient length:

      ```
      mehar@len-y520:/home/mehar/Desktop/os-repo>>
      ```

```
$ rm
Usage: rm [-vf] <path> …
```

3. Both options can be used together.

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ rm -vf four.txt
Removed 'four.txt'
Successful.
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ rm -fv five.txt
Removed 'five.txt'
Successful.
```

4. mkdir

Options implemented: -m=[mode] (specifies the permissions for the directory we are creating) and -v (shows what directory is being created)

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ mkdir test1 test2
Successful.
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ mkdir -m=rw- test3
Successful.
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ mkdir -v test4 test5
Created directory 'test4'.
Created directory 'test5'.
Successful.
```

Permissions as in `ls -l:`

```
drw-rw-r-- 2 mehar mehar 4.0K Oct 31 23:19 test3
```

Errors/Edge cases handled:

1. Invalid command of insufficient length:

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
```

```
$ mkdir
Incorrect usage of command, mkdir.
```

2. If directory already exists, the command terminates:

   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ mkdir test1
   Directory exists.
   ```

3. Both options can be used together:

   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ mkdir -v -m=rwx test6 test7
   Created directory 'test6'.
   Created directory 'test7'.
   Successful.
   ```

5. cat

   Options implemented: -n (numbers each line of the output) and -e (marks the end of each line with a $)

```
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ cat a.txt b.txt
start of a

hello world



end of a
b starts

goodbye
b ends
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ cat a.txt b.txt > output.txt
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ cat output.txt
In child.
start of a

hello world



end of a
b starts
```

```
goodbye
b ends
mehar@len-y520:/home/mehar/Desktop/os-repo>>
$ cat -n output.txt
1  start of a
2
3  hello world
4
5
6
7  end of a
8  b starts
9
10  goodbye
11  b ends
mehar@len-y520:/home/mehar/Desktop/os-repo>>
```

Errors/Edge cases handled:

1. The destination file does not exist:

   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ cat a.txt c.txt > output2.txt
   Cannot open file 'output2.txt'.
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   ```

2. If only a single token is given, the usage of the command is displayed:

   ```
   mehar@len-y520:/home/mehar/Desktop/os-repo>>
   $ cat
   Usage: cat [-ne] <file1> ... >
   <infile>mehar@len-y520:/home/mehar/Desktop/os-repo>>
   ```