OS ASSIGNMENT-1

This is a basic linux shell that can handle the following internal commands:

1. **echo:** -n and –help options have been handled for this command. It checks for the case when user does not input anything after echo (and displays an error message for missing operand) and the case when user wants to print a string with multiple items.
2. **cd:** -L and -P options have been handled for this command. It checks for the case where user has not provided the new directory (and displays an error message for missing operand) and the case when the directory provided does not exist in current directory (it gives no output and directory remains same).
3. **pwd:** -L and -P options have been handled for this command. It checks for the case when there is an invalid option inputted (in this case, all options besides -L and -P would be invalid) and displays appropriate error message.

The shell also handles the following external commands, each of which is executed in a separate file

1. **ls:** -a and -A options have been handled for this command. Entering only “ls” or “ls -A” displays all files in the current directory except “.” and “..”. “ls -a” displays all files, including hidden files.
2. **cat:** -n and -E options have been handled for this command. The shell makes sure that the correct file is being checked by opening the file from the user’s current location instead of the directory where the program is running. -n option numbers all the lines while printing. -E option adds “$” at the end of each

line. The command checks whether the given file exists in that directory or not and displays appropriate error message in case the file does not exist. The shell also handles the error when user does not provide any input after typing the command (i.e., the user just enters “cat”).

3. **rm:** -f and -I options have been handled. -i option prompts the user for confirmation before removing the file. -f option removes the file without prompt and ignores the case when file does not exist or argument is not provided. The command checks for missing operand (file name) in -i option and the case when no flags are used.
4. **mkdir:** -m and -p options have been handled. -m allows user to enter the mode along with name of new directory. -p checks if parent directories exist, and creates them if they do not. The command checks for missing operand and the case when directory cannot be created or already exists.
5. **date:** -u and -R options have been handled. -u prints UTC date time. -R prints RFC 5322 date time. The case for invalid option has been handled.

All external commands are executed by creating a new child process using fork() system call and calling the executable file using execv() function in the child process.

If external command is followed by “&t” then thread is created and execution is done using system().

There is also an exit command in case the user wishes to exit the shell.

Test Cases:

$ echo

echo: missing operand

$ echo Hello

Hello

$ echo -n Hello

Hello$ ls

OS_ASSIGNMENT_1.c cat cat.c date date.c ls ls.c makefile mkdir mkdir.c rm rm.c simple_linux_shell

$ ls -a

. .. OS_ASSIGNMENT_1.c cat cat.c date date.c ls ls.c makefile mkdir mkdir.c rm rm.c simple_linux_shell

$ ls -A

OS_ASSIGNMENT_1.c cat cat.c date date.c ls ls.c makefile mkdir mkdir.c rm rm.c simple_linux_shell

$ pwd

/mnt/c/Users/ELL/Desktop/OS_ASSIGNMENT_1_FILES

$ cd ..

$ pwd

/mnt/c/Users/ELL/Desktop

$ cat abc.txt

ab

c

$ cat -n abc.txt

1 ab

2 c

$ cat -E abc.txt

ab$

```
c$

$ cat

cat: missing operand

$ cd OS_ASSIGNMENT_1_FILES

$ mkdir test_case

$ ls

OS_ASSIGNMENT_1.c cat cat.c date date.c ls ls.c makefile mkdir
mkdir.c rm rm.c simple_linux_shell test_case

$ date -u

Wed Oct 26 14:38:28 2022 UTC

$ date -R

Wed, 26 Oct 2022 20:08:51 +0530

$ date

Wed Oct 26 20:09:51 2022 IST

$ rm -i test_case

rm: remove regular file ‘test_case’? (y/n)y

$ ls

OS_ASSIGNMENT_1.c cat cat.c date date.c ls ls.c makefile mkdir
mkdir.c rm rm.c simple_linux_shell

$ cd

cd: missing operand

$ pwd -x

pwd: invalid option – ‘-x’

$ exit
```