Documentation

Internal Commands

1. Echo
    a. -n
    b. --help

Corner case/errors/bugs handled -

1. When only echo is written, the shell will print an empty string with a newline.
2. If echo --help + “string” then invalid command will be printed.

2. Cd
    a. -L
    b. -P

Corner case/errors/bugs handled

1.If a flag other than -L or -P is used then invalid command will be printed.

2.If commands with -L or -P are written incorrectly then an error message showing invalid command will be printed.

3.If the path of a directory is not written correctly or the directory does not exist then an error message is shown.

3. Pwd
    a. -L
    b. -P

Corner case/errors/bugs handled

1.When pwd is used with an invalid flag error is thrown.

2.When pwd with -P or -L is written incorrectly an error message is shown.
For eg (pwd -P HI)

External Commands

1. Ls (&tls)
    a. -a
    b. -1

Corner cases

1.If a flag other than -a or -1 is used then invalid command will be printed.

2. If -a and -l are written incorrectly then an error message saying invalid command will be shown .

2. Rm (&trm)
    a. -i
    b. -r

Corner cases

1.If a flag other than -r or -i is used then an error message saying invalid command will be printed.

2. If a directory is written with rm without -r then an error message showing that it is a directory is given.

3. If a file or directory which does not exist is given with rm then an error message saying that file or directory does not exist is printed.

3. Mkdir(&tmkdir)
    a. -v
    b. -p

Corner cases -

1. If an invalid flag is used then invalid command will be printed.

2. If a directory already exists then it is shown that the directory exists and a new directory is not created.

3. If the -p option is not used and the parent directory does not exist an error message is displayed.

4. Cat(&cat)
    a. -E
    b. -n

Corner cases -

1. If one of the files does not exist it will not be read and all other files will be read normally and printed.
2. If an invalid flag is used invalid command will be printed.

5. Date(&tdate)
    a. -u
    b. -r

Corner cases-

1.If a flag other than -r or -u is used then an error message saying invalid command will be printed.
2.If -r or -u is written incorrectly then an error message is displayed.
3. If something other than the flags is written with date then an error message showing invalid command is displayed.

4. If the file does not exist in -r then an error message is shown that the file does not exist.

TestCases:-

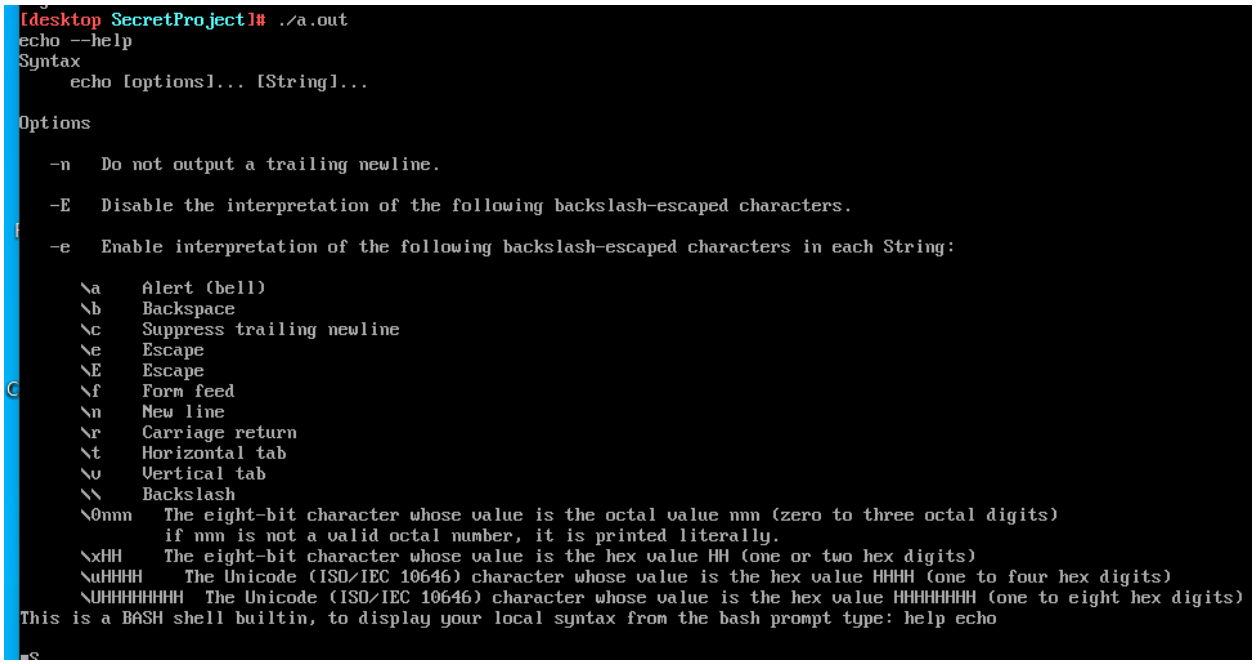
Three textfiles named(test1.txt, test2.txt and test3.txt) have been included.
Echo.txt has also been included for echo -- help option
Shell will terminate on pressing ctrl+c
 make
./a.out

echo hi hello
Output - hi hello

 echo -n hi hello
Output - hi hello without new line char

echo --help

echo.txt will be read and printed on the screen



echo

echo --help hinkn

Output - Invalid Command

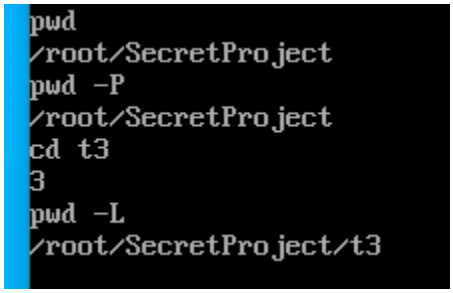
pwd

Output - present working directory is printed on the Screen

pwd -L

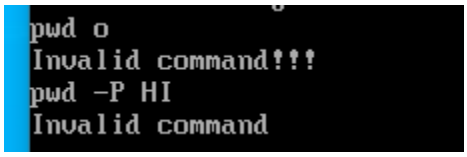
Output - present working directory is printed on the Screen

pwd -P
Output - present working directory is printed on the Screen



pwd -o
Output- Invalid Command

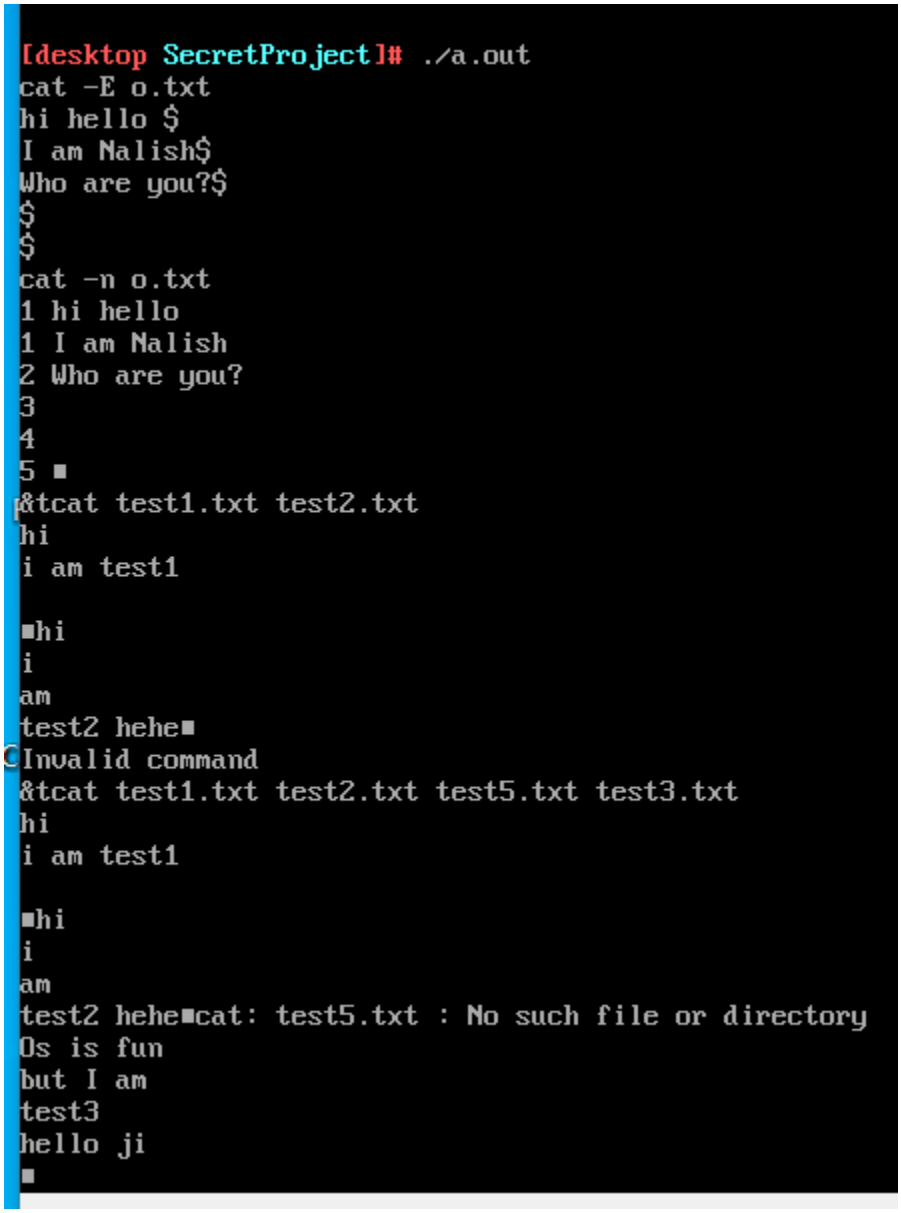
pwd -P HI
Output- Invalid Command



Cat/&tcat test1.txt test2.txt
Output - Files will be read and printed
Cat/&tcat test1.txt test2.txt test5.txt test3.txt
Output - Files will be read and printed

Cat/&tcat -n test1.txt test3.txt

Nano o.txt to create a new file write something in it
Cat/&tcat -E o.txt
Output - file with “$” at the end of every line will be printed.



Cat -j
Output - Invalid command

```
cat -j
Invalid commad
```

```
cat test1.txt test2.txt > test5.txt
cat test5.txt
hi
i am test1

hi
i
am
test2 hehe
Invalid command
cat -n test1.txt test2.txt > test6.txt


cat test6.txt
1 hi
1 i am test1
2
3 4 hi
4 i
5 am
6 test2 hehe
```

```
[desktop SecretProject]# ./a.out
cat -E o.txt iu.txt > hj.txt
cat hj.txt
nejrer$
ekne$
$
sfkfs$
dsnn$
$
$
enwknf$
dksn$
 gfb  $
dd $
 fff$
$
h$
$
```

Mkdir/&tmkdir t1 t3 t4

Output - t1 t3 t4 directories are created.

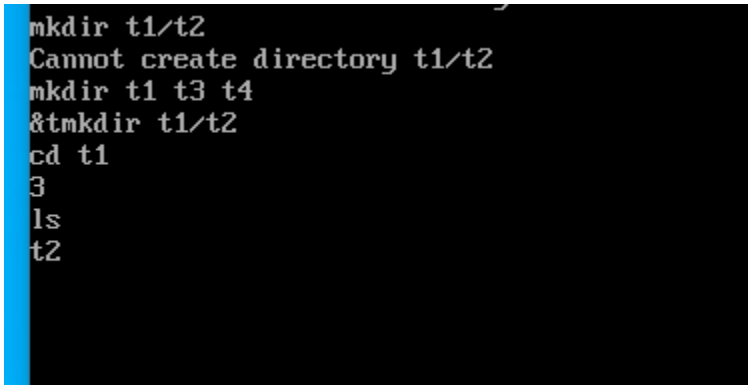
```
ls
test3.txt mkdir Myshell.c ls.c date rm.c cat.c ls rm t6 date.c echo.txt a.out cat mkdir.c t7 makefile
mkdir t1 t3 t4
ls
test3.txt t3 mkdir Myshell.c ls.c date rm.c cat.c ls rm t6 date.c t1 echo.txt a.out cat mkdir.c t4 t7 makefile
```

mkdir

Mkdir /&tmkdir t1/t2

Output - t2 will be created inside t1

Mkdir/&tmkdir t1



Output- Error message - directory already exists will be shown

To check whether it is there or not we do

Cd t1

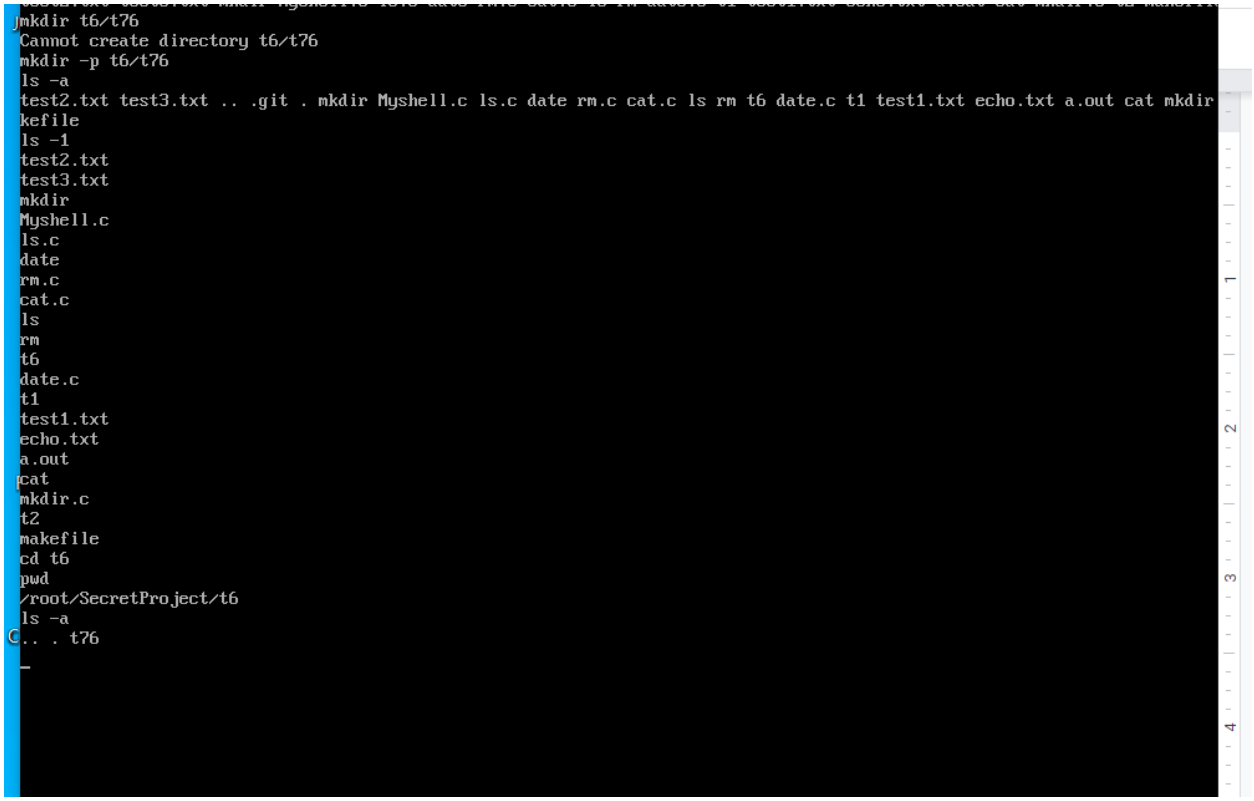
And then ls

Cd .. to get back

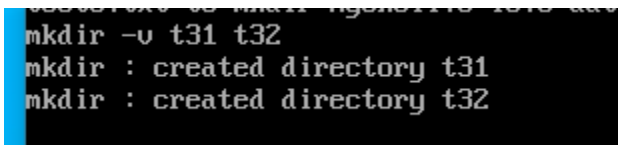
Mkdir/&tmkdir t6/t76

Output - Error message as directory t6 does not exist

Mkdir/&tmkdir -p t6/t76
Output - Directory t6 will be created first and then t76 will be created inside it

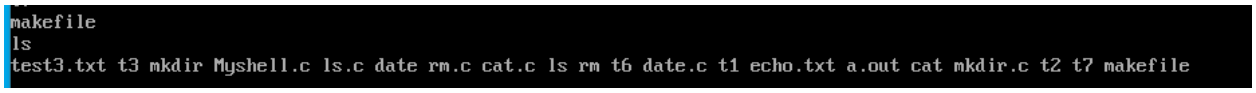


Cd .. to get back
Mkdir/&tmkdir -v t31 t32
Output - Directory t31 and t32 will be created and message for the creation of the said directories will be printed.



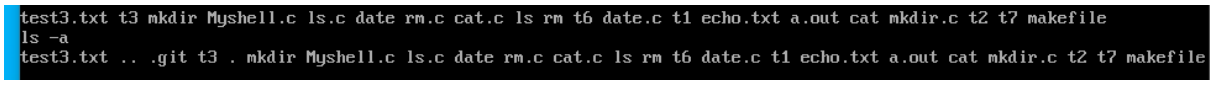
Now to see the contents of the original directory

ls/&tls



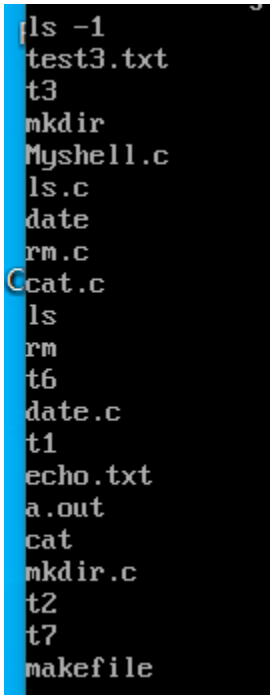
To see hidden files

ls/&tls -a



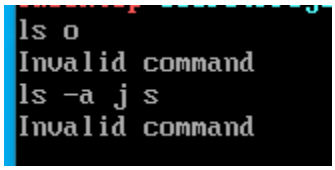
To see the contents of the file line by line

ls/&tls -1



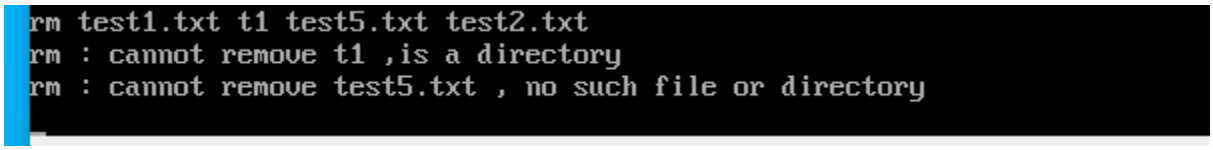
ls/&tls o
Output - Error message saying Invalid command will be shown.

ls /&tls -a j s
Output - Error message saying invalid flag will be shown.



rm/&trm -o
Output - Invalid command

rm/&trm test1.txt t1 test5.txt test2.txt
Output - test1.txt and test2.txt are deleted



rm/&trm -r t31
ls
rm/&trm -r t31 t32
Output

```
rm -r t31
ls
test3.txt t3 mkdir t32 Myshell.c ls.c date rm.c cat.c ls rm t6 date.c t1 echo.txt a.out cat mkdir.c t7 makefile
rm -t t31 t32
Invalid command
rm -r t31 t32
rm : cannot remove t31 , no such file or directory
ls
test3.txt t3 mkdir Myshell.c ls.c date rm.c cat.c ls rm t6 date.c t1 echo.txt a.out cat mkdir.c t7 makefile
```

&trm -i test1.txt

&trm -i test3.txt

ls

Choosing n

&trm -i test.txt

ls

choosing y

ls

```
&trm -i test3.txt
rm: remove regular file test3.txt ?(y/n)n
ls
test3.txt t3 mkdir Myshell.c ls.c date rm.c cat.c ls rm t6 date.c t1 echo.txt a.out cat mkdir.c t4 t7 makefile
&trm -i test3.txt
rm: remove regular file test3.txt ?(y/n)y
ls
t3 mkdir Myshell.c ls.c date rm.c cat.c ls rm t6 date.c t1 echo.txt a.out cat mkdir.c t4 t7 makefile
```

date/&tdate

```
date
Mon Oct 24 11:39:56 2022
```

```
&tdate
Mon Oct 24 11:40:17 2022
```

date/&date -u

```
date -u
Mon Oct 24 11:41:54 UTC 2022
&tdate -u
Mon Oct 24 11:42:6 UTC 2022
```

date -r echo.txt

```
date -r echo.txt
Mon Oct 24 11:39:18 2022
```

&tdate -r test7.txt

```
&tdate -r test7.txt
date : no such file or directory exists
```

&tdate -r test1.txt test5.txt

&tdate -u jl

date -o

date ifo

```
date -r test1.txt test5.txt
Invalid command
&tdate -u jl
Invalid command
date -o
Invalid command
date ifo
Invalid command
```

Cd and pwd mixed -

```
&tmkdir t1 t2
ls
test2.txt test3.txt mkdir Myshell.c ls.c date rm.c cat.c ls rm date.c t1 test1.txt echo.txt a.out cat mkdir.c t2 makefil
cd -o hi
Invalid command
cd -L ks
cd : no such file or directory exists
cd -L t1 t3
Invalid command
cd -L t1
pwd
/root/SecretProject/t1
```

```
ls
test2.txt test3.txt mkdir Myshell.c ls.c date rm.c cat.c ls rm date.c t1 test1.txt echo.txt a.out cat mkdir.c t2 makefil
pwd
/root/SecretProject
cd
pwd
/root
cd SecretProject
pwd
/root/SecretProject
cd -L t1
pwd -L
/root/SecretProject/t1
cd -P t2
cd : no such file or directory exists
ls
t3
cd -P t3
ls

pwd
/root/SecretProject/t1/t3
cd ..
pwd
/root/SecretProject/t1
cd ..
pwd -P
/root/SecretProject
```