OS Assignment1 report

Aditya Raj
2021512

My system:

- Done on an Ubuntu pc.
- Taking line by line input through stdin.
- Separating the input into tokens separated by space
- Checking if &t is present **at the end.** If so, then proceeding with external command execution through threads.
- If internal command, then executing within the shell itself.
- If external command, then either forking and calling execvp **(through a system specific path),** or using pthread_create and pthread_join to call the appropriate executable.
- All my external commands are in different c files.

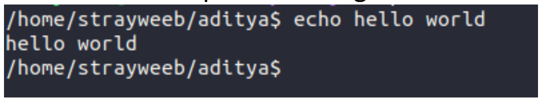
Commands:

Invalid command entered (not any one of these is also handled)
Commands with &t at the end use threads and are interlaced within the examples

Echo:

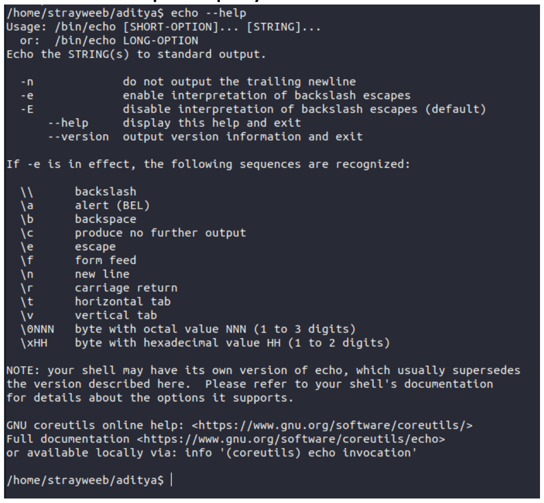
Options implemented: --help and -n

1. Default: prints message with newline



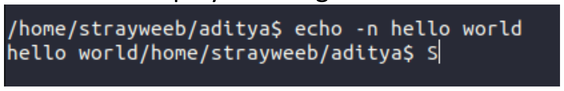
```
/home/strayweeb/aditya$ echo hello world
hello world
/home/strayweeb/aditya$
```

2. –help: displays manual



```
/home/strayweeb/aditya$ echo --help
Usage: /bin/echo [SHORT-OPTION]... [STRING]...
  or:  /bin/echo LONG-OPTION
Echo the STRING(s) to standard output.

  -n             do not output the trailing newline
  -e             enable interpretation of backslash escapes
  -E             disable interpretation of backslash escapes (default)
      --help     display this help and exit
      --version  output version information and exit

If -e is in effect, the following sequences are recognized:

  \\      backslash
  \a      alert (BEL)
  \b      backspace
  \c      produce no further output
  \e      escape
  \f      form feed
  \n      new line
  \r      carriage return
  \t      horizontal tab
  \v      vertical tab
  \0NNN   byte with octal value NNN (1 to 3 digits)
  \xHH    byte with hexadecimal value HH (1 to 2 digits)

NOTE: your shell may have its own version of echo, which usually supersedes
the version described here.  Please refer to your shell's documentation
for details about the options it supports.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/echo>
or available locally via: info '(coreutils) echo invocation'

/home/strayweeb/aditya$
```

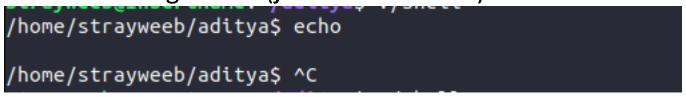
3. -n: displays message without newline at the end



```
/home/strayweeb/aditya$ echo -n hello world
hello world/home/strayweeb/aditya$ S
```

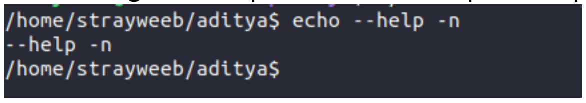
Corner cases handled:

1. No argument (just echo written)

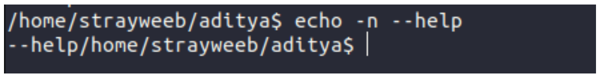


```
/home/strayweeb/aditya$ echo

/home/strayweeb/aditya$ ^C
```

2. Arguments specified after –help or –help written after an option

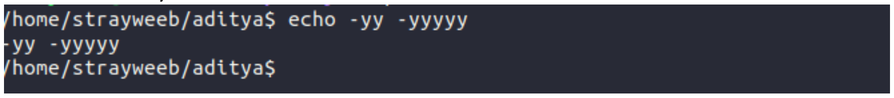


```
/home/strayweeb/aditya$ echo --help -n
--help -n
/home/strayweeb/aditya$
```



```
/home/strayweeb/aditya$ echo -n --help
--help/home/strayweeb/aditya$
```

3. Invalid options similarly just get printed by echo (since that's how it happens with the actual echo)

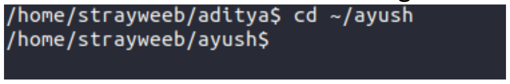


```
/home/strayweeb/aditya$ echo -yy -yyyyy
-yy -yyyyy
/home/strayweeb/aditya$
```

Cd:

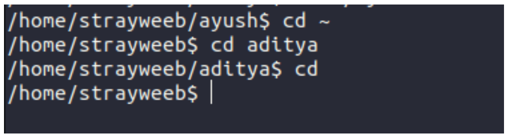
Options implemented: -L and -P and various specific cd commands

1. Default: same as -L. changes directory



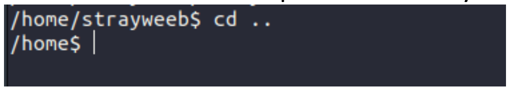
```
/home/strayweeb/aditya$ cd ~/ayush
/home/strayweeb/ayush$
```

- cd and cd ~ - takes to home directory



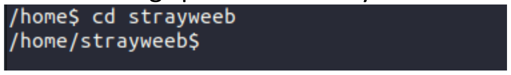
```
/home/strayweeb/ayush$ cd ~
/home/strayweeb$ cd aditya
/home/strayweeb/aditya$ cd
/home/strayweeb$
```

- cd .. – takes to parent directory



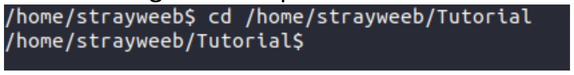
```
/home/strayweeb$ cd ..
/home$
```

- using specific directory:



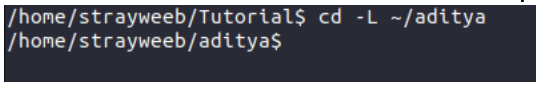
```
/home$ cd strayweeb
/home/strayweeb$
```

- using absolute path



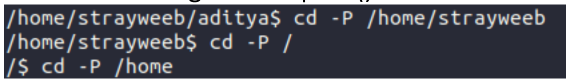
```
/home/strayweeb$ cd /home/strayweeb/Tutorial
/home/strayweeb/Tutorial$
```

2. -L: the default mode cd uses if no option is specified. Uses symlinks too.



```
/home/strayweeb/Tutorial$ cd -L ~/aditya
/home/strayweeb/aditya$
```

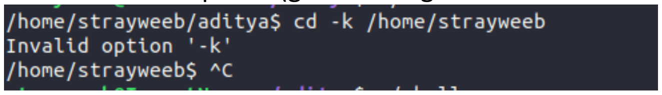
3. -P: using the realpath() function in c to use the physical directory.



```
/home/strayweeb/aditya$ cd -P /home/strayweeb
/home/strayweeb$ cd -P /
/$ cd -P /home
```

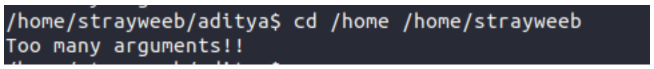
Errors handled:

1. Invalid option: (graceful degradation taken in all invalid option cases)



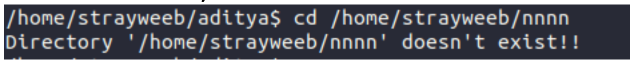
```
/home/strayweeb/aditya$ cd -k /home/strayweeb
Invalid option '-k'
/home/strayweeb$ ^C
```

2. Too many arguments



```
/home/strayweeb/aditya$ cd /home /home/strayweeb
Too many arguments!!
```

3. Directory doesn’t exist

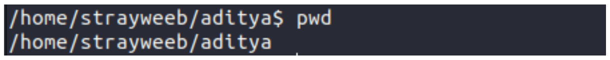


```
/home/strayweeb/aditya$ cd /home/strayweeb/nnnn
Directory '/home/strayweeb/nnnn' doesn't exist!!
```

Pwd:

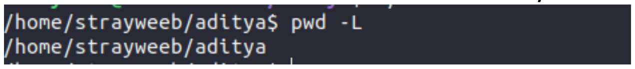
Options implemented:

1. Default:



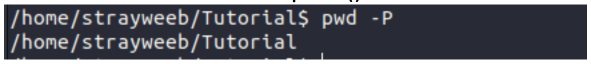
```
/home/strayweeb/aditya$ pwd
/home/strayweeb/aditya
```

2. -L: same as the default mode. Uses symlinks ($PWD)



```
/home/strayweeb/aditya$ pwd -L
/home/strayweeb/aditya
```

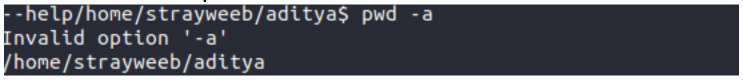
3. -P: uses the realpath() c function to use the physical directory.



```
/home/strayweeb/Tutorial$ pwd -P
/home/strayweeb/Tutorial
```

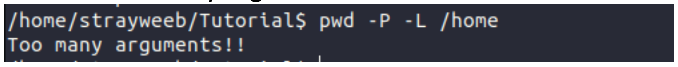
Errors handled:

1. Invalid option:



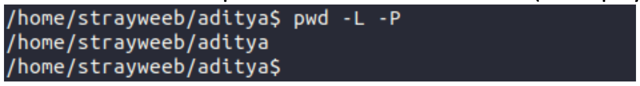
```
--help/home/strayweeb/aditya$ pwd -a
Invalid option '-a'
/home/strayweeb/aditya
```

2. Too many arguments



```
/home/strayweeb/Tutorial$ pwd -P -L /home
Too many arguments!!
```

3. -L and -P specified at the same time (uses physical address)



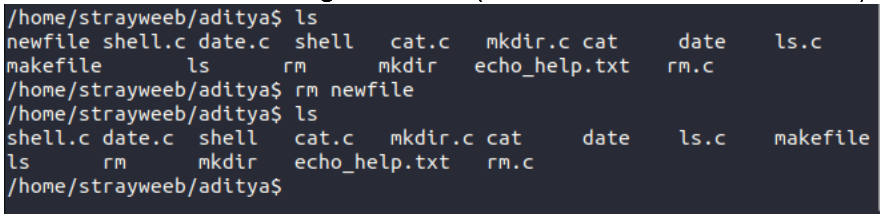
```
/home/strayweeb/aditya$ pwd -L -P
/home/strayweeb/aditya
/home/strayweeb/aditya$
```

For all external commands juggling of option order (and specifying more than one option) has also been handled as a part of corner cases. Examples given.

Rm:

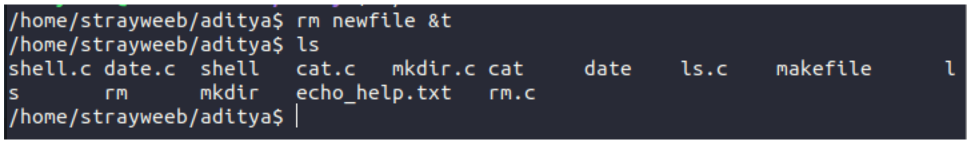
Options implemented:

1. Default: deletes argument files (can take more than one file too)



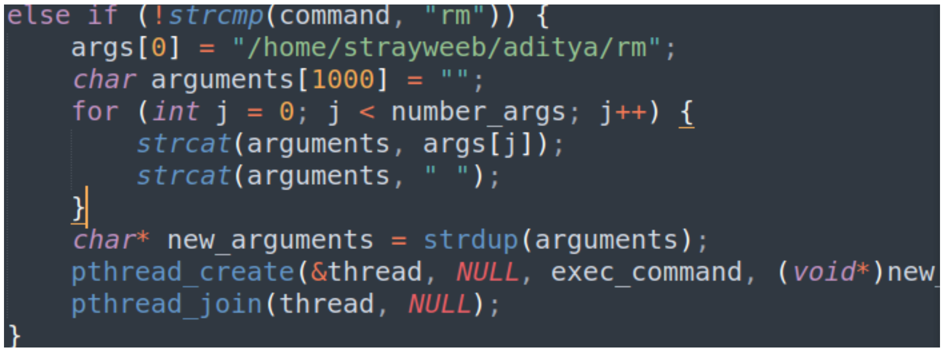
```
/home/strayweeb/aditya$ ls
newfile shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c
makefile       ls      rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$ rm newfile
/home/strayweeb/aditya$ ls
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile
ls      rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$
```

With threads:



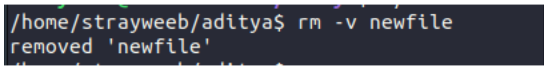
```
/home/strayweeb/aditya$ rm newfile &t
/home/strayweeb/aditya$ ls
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile      l
s       rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$
```

Thread implementation (similar for the other external commands)



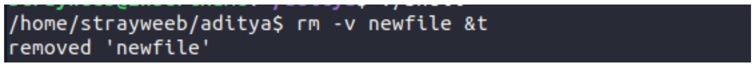
```
else if (!strcmp(command, "rm")) {
    args[0] = "/home/strayweeb/aditya/rm";
    char arguments[1000] = "";
    for (int j = 0; j < number_args; j++) {
        strcat(arguments, args[j]);
        strcat(arguments, " ");
    }
    char* new_arguments = strdup(arguments);
    pthread_create(&thread, NULL, exec_command, (void*)new
    pthread_join(thread, NULL);
}
```

2. -v: shows what has been deleted



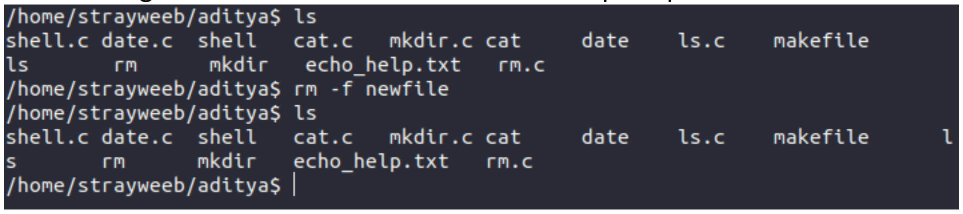
```
/home/strayweeb/aditya$ rm -v newfile
removed 'newfile'
```

With threads:



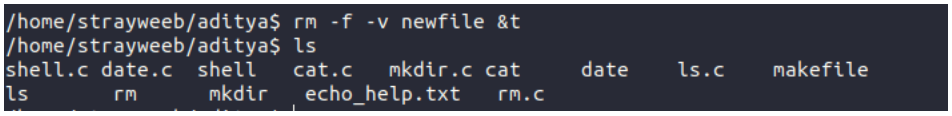
```
/home/strayweeb/aditya$ rm -v newfile &t
removed 'newfile'
```

3. -f: ignores non-existent files and does not prompt either



```
/home/strayweeb/aditya$ ls
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile
ls       rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$ rm -f newfile
/home/strayweeb/aditya$ ls
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile      l
s       rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$ 
```

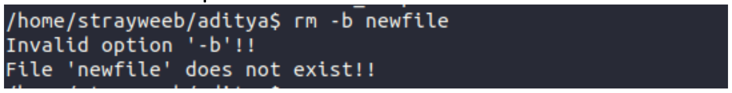
With threads:



```
/home/strayweeb/aditya$ rm -f -v newfile &t
/home/strayweeb/aditya$ ls
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile
ls       rm      mkdir   echo_help.txt   rm.c
```

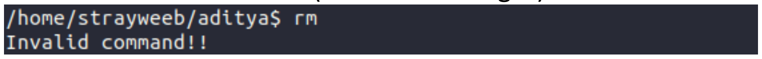
Errors handled:

1. Invalid options:



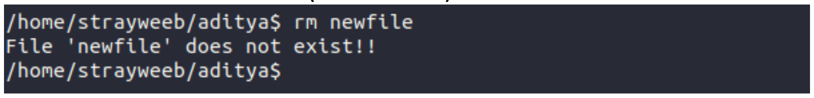
```
/home/strayweeb/aditya$ rm -b newfile
Invalid option '-b'!!
File 'newfile' does not exist!!
```

2. Invalid command (insufficient length)



```
/home/strayweeb/aditya$ rm
Invalid command!!
```

3. File doesn't exist (when no -f)

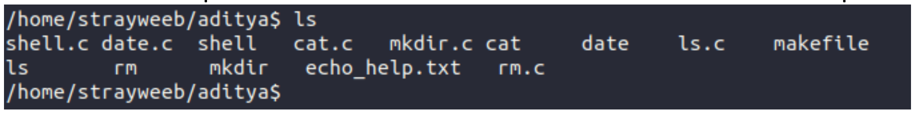


```
/home/strayweeb/aditya$ rm newfile
File 'newfile' does not exist!!
/home/strayweeb/aditya$
```

Ls:

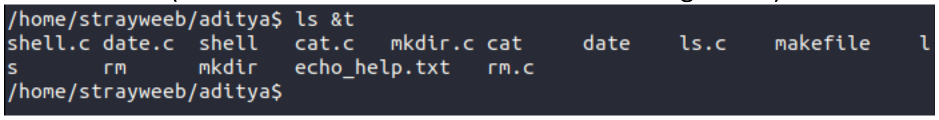
Options implemented:

1. Default: prints out the contents of the cwd or the directories specified (no hidden files (., ..))



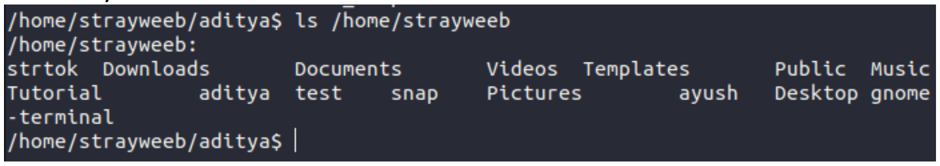
```
/home/strayweeb/aditya$ ls
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile
ls       rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$
```

with threads: (other calls also work with &t as the end argument)



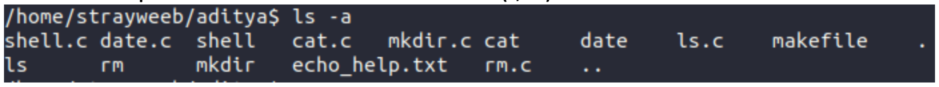
```
/home/strayweeb/aditya$ ls &t
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile     l
s       rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$
```

Directory:



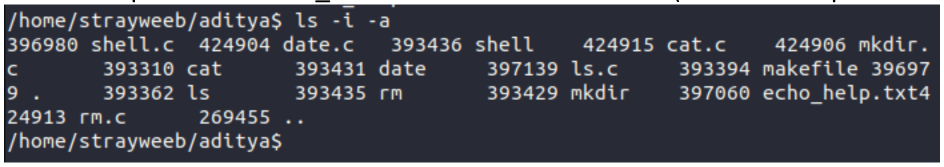
```
/home/strayweeb/aditya$ ls /home/strayweeb
/home/strayweeb:
strtok  Downloads       Documents       Videos  Templates       Public  Music
Tutorial        aditya  test    snap    Pictures        ayush   Desktop gnome
-terminal
/home/strayweeb/aditya$ |
```

2. -a: prints out the hidden files too (., ..)



```
/home/strayweeb/aditya$ ls -a
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile    .
ls      rm      mkdir   echo_help.txt   rm.c    ..
```

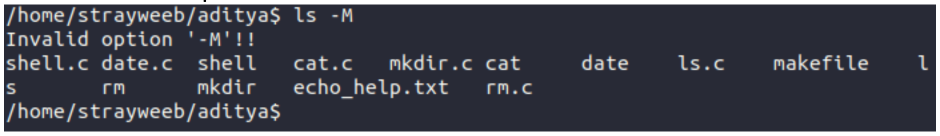
3. -i: prints out the d_ino or index of each content (shows two options here, similar for others)



```
/home/strayweeb/aditya$ ls -i -a
396980 shell.c  424904 date.c   393436 shell    424915 cat.c    424906 mkdir.
c       393310 cat      393431 date     397139 ls.c     393394 makefile 39697
9 .     393362 ls       393435 rm       393429 mkdir    397060 echo_help.txt4
24913 rm.c      269455 ..
/home/strayweeb/aditya$
```

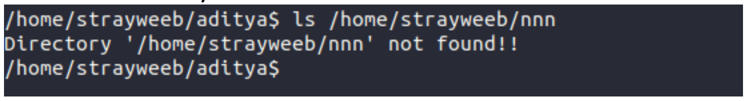
Errors handled:

1. Invalid option:



```
/home/strayweeb/aditya$ ls -M
Invalid option '-M'!!
shell.c date.c  shell   cat.c   mkdir.c cat     date    ls.c    makefile    l
s       rm      mkdir   echo_help.txt   rm.c
/home/strayweeb/aditya$
```

2. Directory not found:

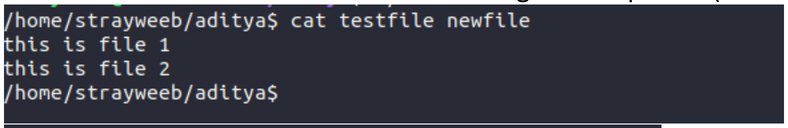


```
/home/strayweeb/aditya$ ls /home/strayweeb/nnn
Directory '/home/strayweeb/nnn' not found!!
/home/strayweeb/aditya$
```

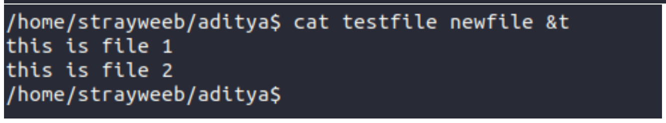
Cat:

Options implemented:

1. Default concatenates files into the given output file(stdout being the default):

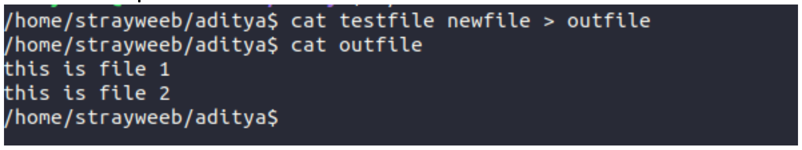


```
/home/strayweeb/aditya$ cat testfile newfile
this is file 1
this is file 2
/home/strayweeb/aditya$
```



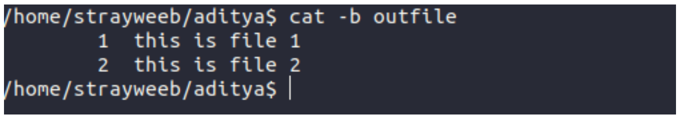
```
/home/strayweeb/aditya$ cat testfile newfile &t
this is file 1
this is file 2
/home/strayweeb/aditya$
```

Into an output file:



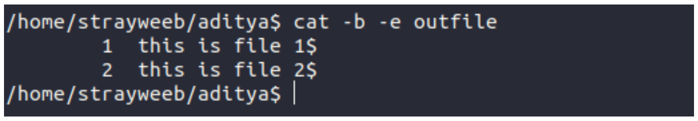
```
/home/strayweeb/aditya$ cat testfile newfile > outfile
/home/strayweeb/aditya$ cat outfile
this is file 1
this is file 2
/home/strayweeb/aditya$
```

2. -b: marks the nonblank lines with a number



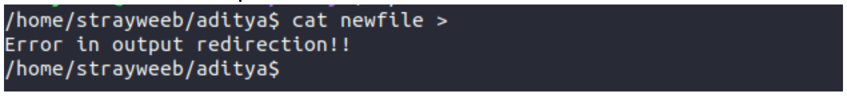
```
/home/strayweeb/aditya$ cat -b outfile
       1  this is file 1
       2  this is file 2
/home/strayweeb/aditya$ |
```

3. -e: marks the end of the line with $

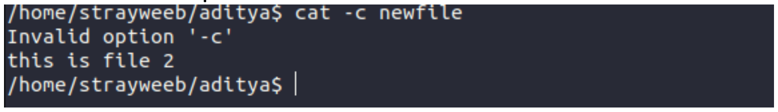

```
/home/strayweeb/aditya$ cat -b -e outfile
       1  this is file 1$
       2  this is file 2$
/home/strayweeb/aditya$ 
```

Errors handled:

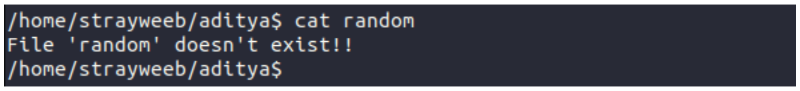
1. Error in output redirection:


```
/home/strayweeb/aditya$ cat newfile >
Error in output redirection!!
/home/strayweeb/aditya$
```

2. Invalid option:


```
/home/strayweeb/aditya$ cat -c newfile
Invalid option '-c'
this is file 2
/home/strayweeb/aditya$ 
```

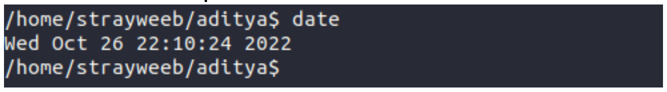
3. File doesn't exist:


```
/home/strayweeb/aditya$ cat random
File 'random' doesn't exist!!
/home/strayweeb/aditya$
```

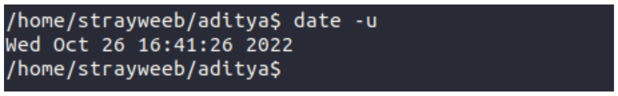
Date:

Options implemented:

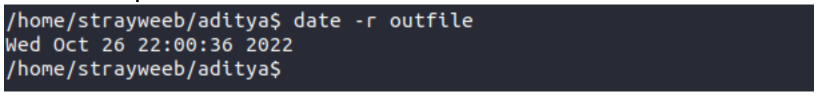
1. Default: prints date in localtime


```
/home/strayweeb/aditya$ date
Wed Oct 26 22:10:24 2022
/home/strayweeb/aditya$
```

2. -u: utc time

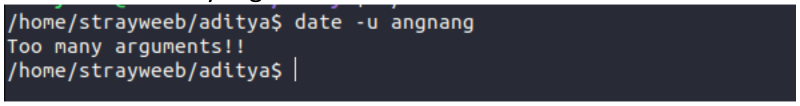

```
/home/strayweeb/aditya$ date -u
Wed Oct 26 16:41:26 2022
/home/strayweeb/aditya$
```

3. -r: prints the last date when file was modified

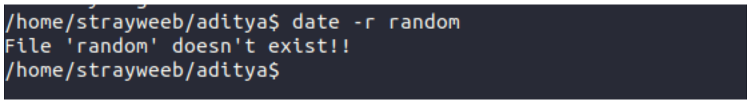

```
/home/strayweeb/aditya$ date -r outfile
Wed Oct 26 22:00:36 2022
/home/strayweeb/aditya$
```

Errors handled:

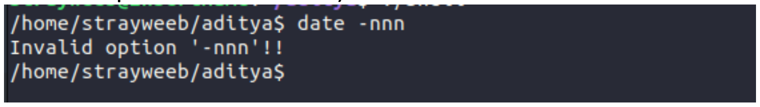
1. Too many arguments:


```
/home/strayweeb/aditya$ date -u angnang
Too many arguments!!
/home/strayweeb/aditya$ 
```

2. File for date -r doesn't exist


```
/home/strayweeb/aditya$ date -r random
File 'random' doesn't exist!!
/home/strayweeb/aditya$
```

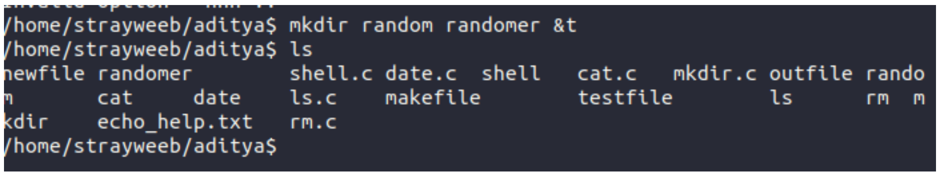
3. Invalid option: (no degradation since I tried this on my system and date doesn't allow invalid options whatsoever)


```
/home/strayweeb/aditya$ date -nnn
Invalid option '-nnn'!!
/home/strayweeb/aditya$
```

Mkdir:

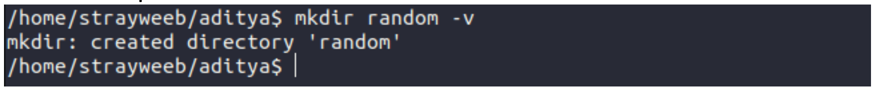
Options implemented:

1. Default: makes an empty directory (no parent directories allowed though)



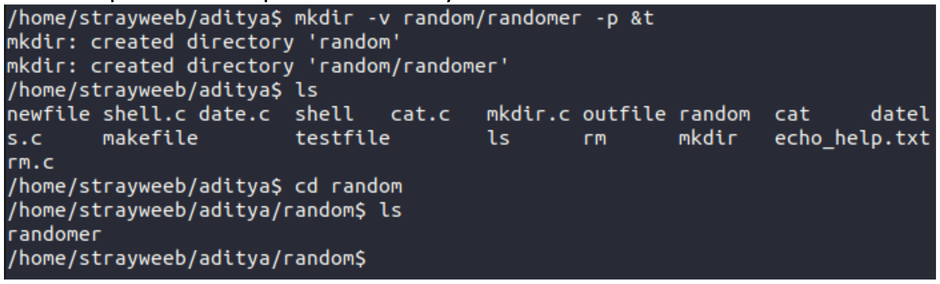
```
/home/strayweeb/aditya$ mkdir random randomer &t
/home/strayweeb/aditya$ ls
newfile randomer        shell.c date.c  shell   cat.c   mkdir.c outfile rando
m       cat     date    ls.c    makefile        testfile        ls      rm   m
kdir    echo_help.txt   rm.c
/home/strayweeb/aditya$
```

2. -v: prints the directories that have been created



```
/home/strayweeb/aditya$ mkdir random -v
mkdir: created directory 'random'
/home/strayweeb/aditya$ |
```

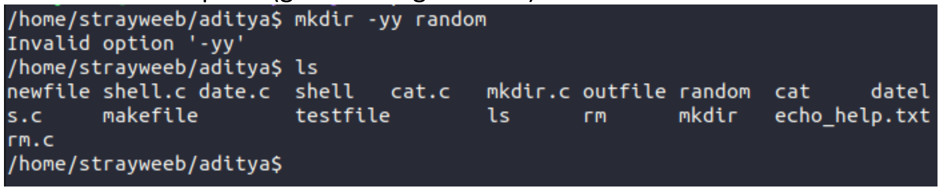
3. -p: allows for parent directory creation



```
/home/strayweeb/aditya$ mkdir -v random/randomer -p &t
mkdir: created directory 'random'
mkdir: created directory 'random/randomer'
/home/strayweeb/aditya$ ls
newfile shell.c date.c  shell   cat.c   mkdir.c outfile random  cat     datel
s.c     makefile        testfile        ls      rm      mkdir   echo_help.txt
rm.c
/home/strayweeb/aditya$ cd random
/home/strayweeb/aditya/random$ ls
randomer
/home/strayweeb/aditya/random$
```

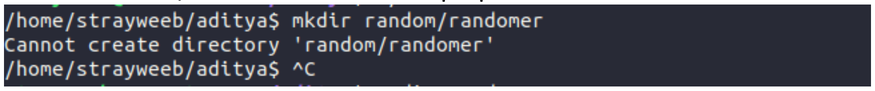
Errors handled:

1. Invalid option (graceful degradation)



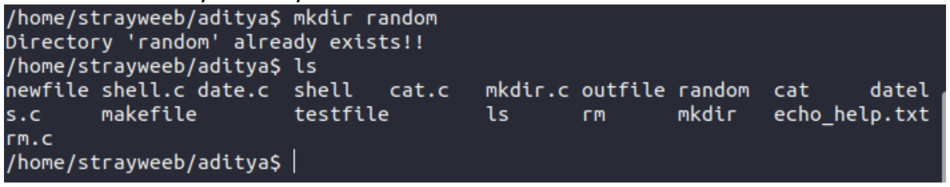
```
/home/strayweeb/aditya$ mkdir -yy random
Invalid option '-yy'
/home/strayweeb/aditya$ ls
newfile shell.c date.c  shell   cat.c   mkdir.c outfile random  cat     datel
s.c     makefile        testfile        ls      rm      mkdir   echo_help.txt
rm.c
/home/strayweeb/aditya$
```

2. When / is used without the -p option



```
/home/strayweeb/aditya$ mkdir random/randomer
Cannot create directory 'random/randomer'
/home/strayweeb/aditya$ ^C
```

3. Directory already exists



```
/home/strayweeb/aditya$ mkdir random
Directory 'random' already exists!!
/home/strayweeb/aditya$ ls
newfile shell.c date.c  shell   cat.c   mkdir.c outfile random  cat     datel
s.c     makefile        testfile        ls      rm      mkdir   echo_help.txt
rm.c
/home/strayweeb/aditya$ |
```

Makefile:

Simply type make and an executable named shell will be generated.