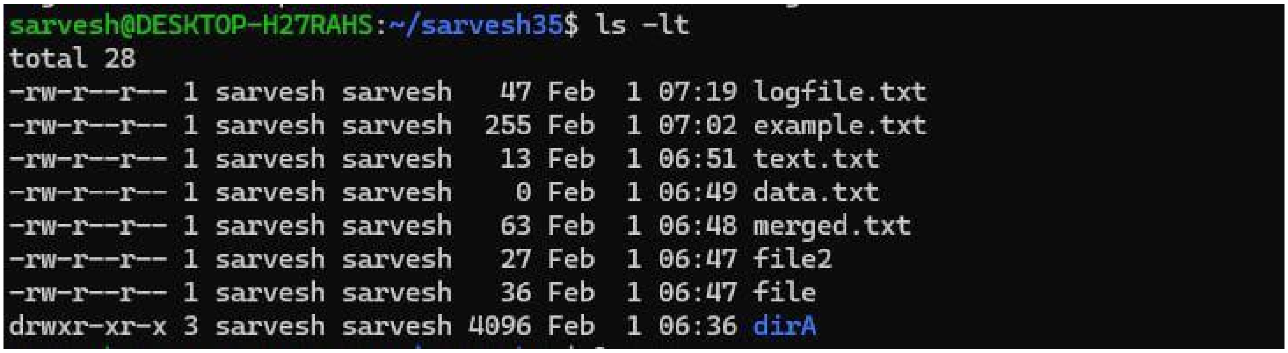
NAME: Sarvesh Parab A2-35

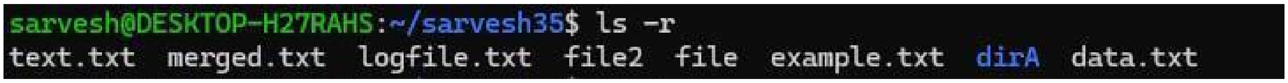
1. File and Directory Operations (`ls`, `mkdir`, `cd`, `rmdir`, `pwd`, `rm`)

1. Use `ls` to display files sorted by modification time.



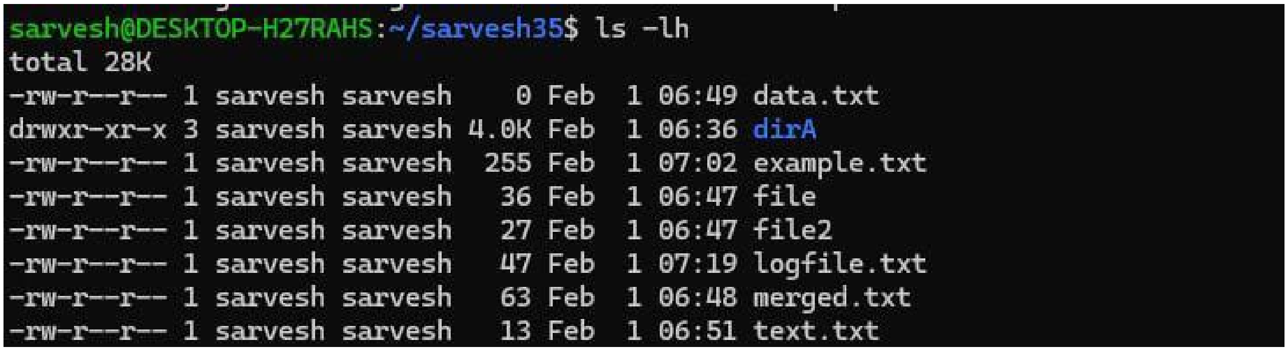
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -lt
total 28
-rw-r--r-- 1 sarvesh sarvesh   47 Feb  1 07:19 logfile.txt
-rw-r--r-- 1 sarvesh sarvesh  255 Feb  1 07:02 example.txt
-rw-r--r-- 1 sarvesh sarvesh   13 Feb  1 06:51 text.txt
-rw-r--r-- 1 sarvesh sarvesh    0 Feb  1 06:49 data.txt
-rw-r--r-- 1 sarvesh sarvesh   63 Feb  1 06:48 merged.txt
-rw-r--r-- 1 sarvesh sarvesh   27 Feb  1 06:47 file2
-rw-r--r-- 1 sarvesh sarvesh   36 Feb  1 06:47 file
drwxr-xr-x 3 sarvesh sarvesh 4096 Feb  1 06:36 dirA
```

2. List files and directories in reverse order using `ls`.



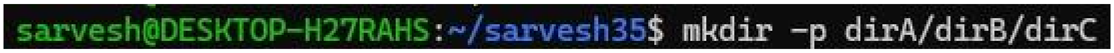
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -r
text.txt  merged.txt  logfile.txt  file2  file  example.txt  dirA  data.txt
```

3. Use `ls -lh` to display file sizes in a human-readable format.



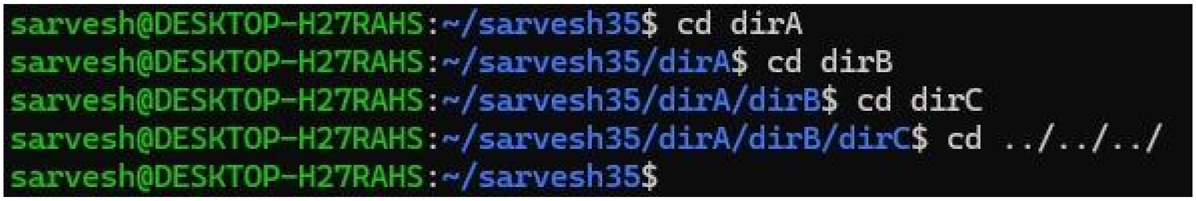
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -lh
total 28K
-rw-r--r-- 1 sarvesh sarvesh    0 Feb  1 06:49 data.txt
drwxr-xr-x 3 sarvesh sarvesh 4.0K Feb  1 06:36 dirA
-rw-r--r-- 1 sarvesh sarvesh  255 Feb  1 07:02 example.txt
-rw-r--r-- 1 sarvesh sarvesh   36 Feb  1 06:47 file
-rw-r--r-- 1 sarvesh sarvesh   27 Feb  1 06:47 file2
-rw-r--r-- 1 sarvesh sarvesh   47 Feb  1 07:19 logfile.txt
-rw-r--r-- 1 sarvesh sarvesh   63 Feb  1 06:48 merged.txt
-rw-r--r-- 1 sarvesh sarvesh   13 Feb  1 06:51 text.txt
```

4. Create a directory structure `dirA/dirB/dirC` using `mkdir -p`.



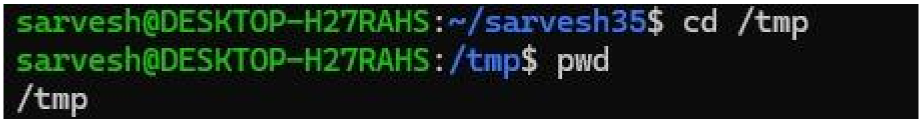
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mkdir -p dirA/dirB/dirC
```

5. Navigate to the parent directory of the current directory using `cd ..`



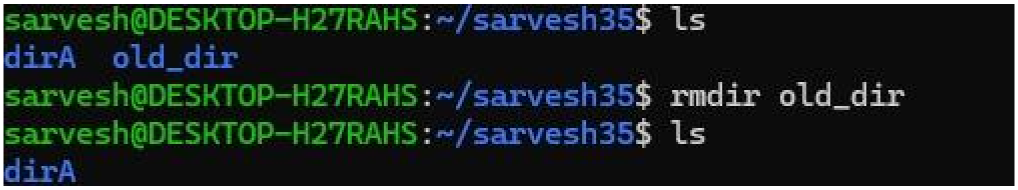
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cd dirA
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/dirA$ cd dirB
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/dirA/dirB$ cd dirC
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/dirA/dirB/dirC$ cd ../../../
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

6. Use `cd` to move into the `/tmp` directory and confirm your location with `pwd`.



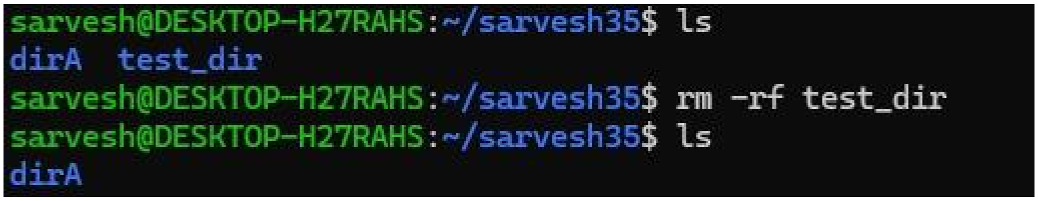
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cd /tmp
sarvesh@DESKTOP-H27RAHS:/tmp$ pwd
/tmp
```

7. Remove an empty directory named `old_dir` using `rmdir`.



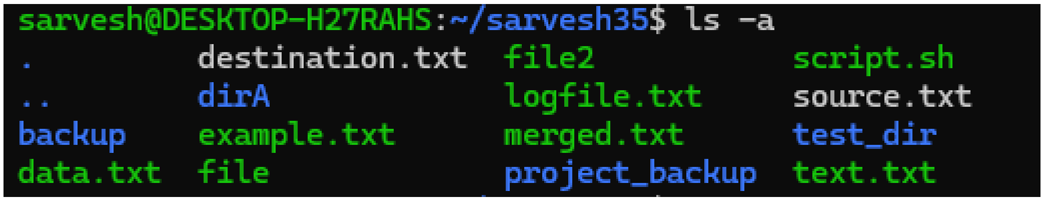
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
dirA  old_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ rmdir old_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
dirA
```

8. Use `rm -rf` to delete a directory `test_dir` and all its contents.



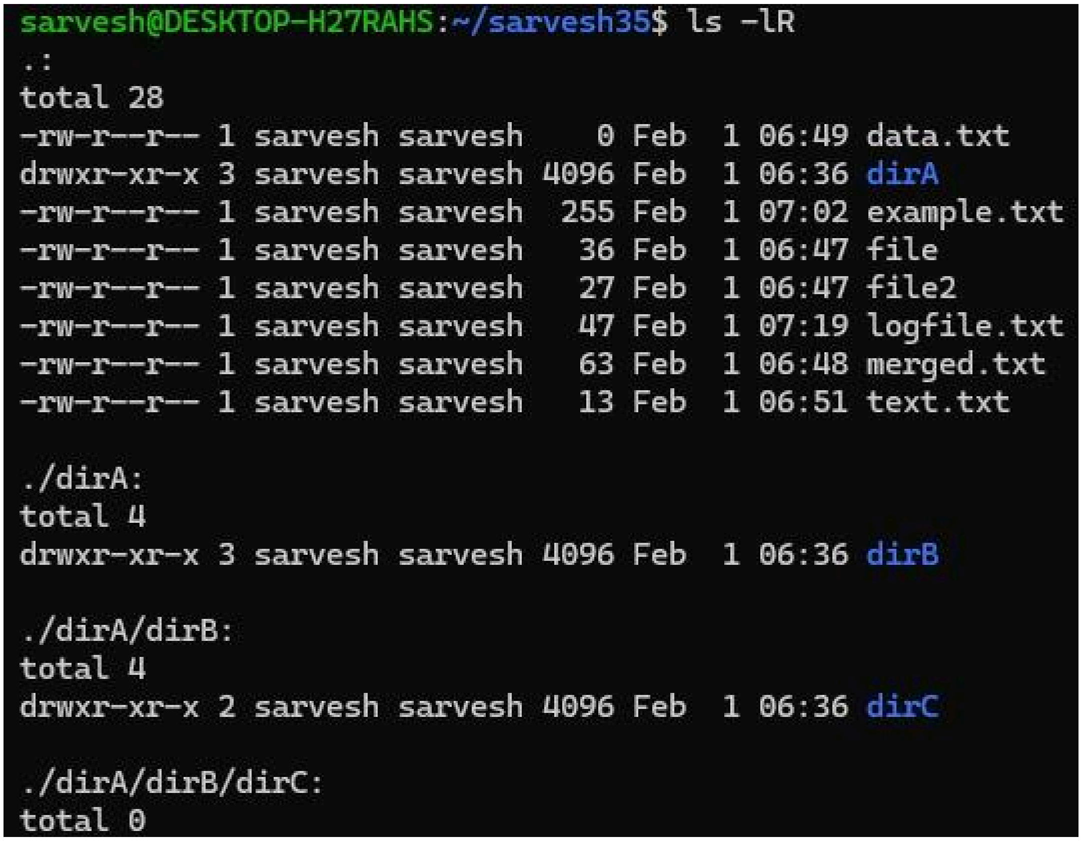
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
dirA  test_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ rm -rf test_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
dirA
```

9. Use `ls` with the `-a` flag to display all files, including hidden ones.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -a
.         destination.txt  file2           script.sh
..        dirA             logfile.txt     source.txt
backup    example.txt      merged.txt      test_dir
data.txt  file             project_backup  text.txt
```

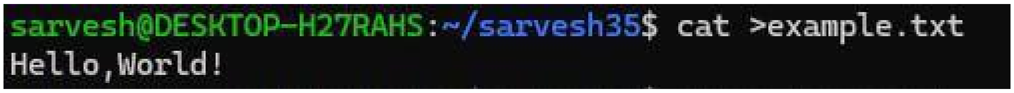
10. Use `ls -lR` to list all files and directories recursively.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -lR
.:
total 28
-rw-r--r-- 1 sarvesh sarvesh    0 Feb  1 06:49 data.txt
drwxr-xr-x 3 sarvesh sarvesh 4096 Feb  1 06:36 dirA
-rw-r--r-- 1 sarvesh sarvesh  255 Feb  1 07:02 example.txt
-rw-r--r-- 1 sarvesh sarvesh   36 Feb  1 06:47 file
-rw-r--r-- 1 sarvesh sarvesh   27 Feb  1 06:47 file2
-rw-r--r-- 1 sarvesh sarvesh   47 Feb  1 07:19 logfile.txt
-rw-r--r-- 1 sarvesh sarvesh   63 Feb  1 06:48 merged.txt
-rw-r--r-- 1 sarvesh sarvesh   13 Feb  1 06:51 text.txt

./dirA:
total 4
drwxr-xr-x 3 sarvesh sarvesh 4096 Feb  1 06:36 dirB

./dirA/dirB:
total 4
drwxr-xr-x 2 sarvesh sarvesh 4096 Feb  1 06:36 dirC

./dirA/dirB/dirC:
total 0
```

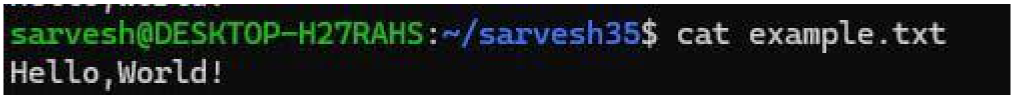
2. File Viewing and Manipulation (`cat`, `wc`, `head`, `tail`, `tac`, `more`, `less`) `cat`

11. Create a file named `example.txt` and write "Hello, World!" into it using `cat > example.txt`.



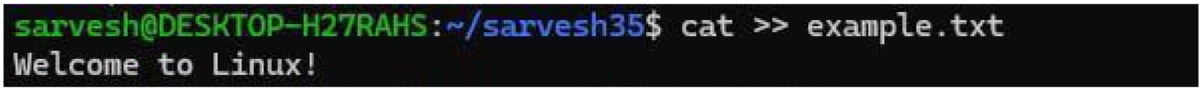
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat >example.txt
Hello,World!
```

12. Use `cat` to display the contents of `example.txt`.



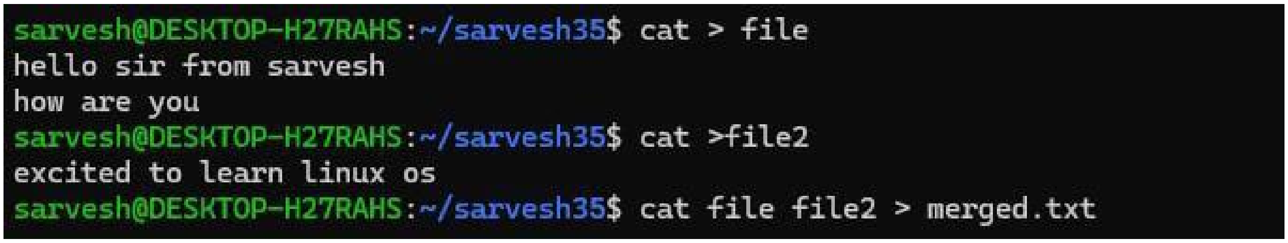
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat example.txt
Hello,World!
```

13. Append "Welcome to Linux!" to `example.txt` using `cat >> example.txt`.

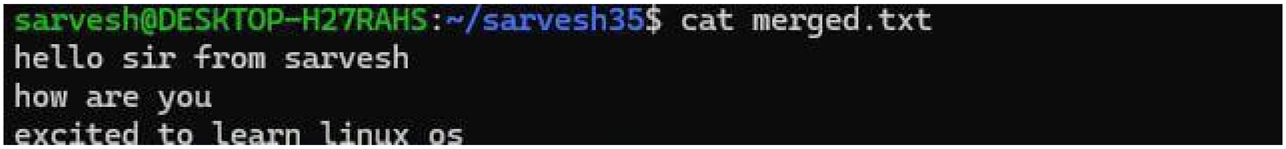


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat >> example.txt
Welcome to Linux!
```

14. Concatenate two files `file1.txt` and `file2.txt` and save the result into `merged.txt`.

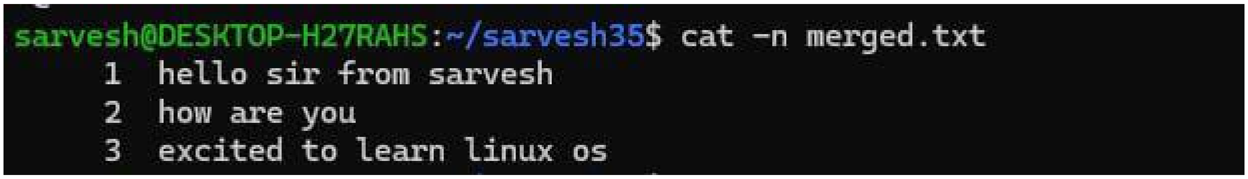


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat > file
hello sir from sarvesh
how are you
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat >file2
excited to learn linux os
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat file file2 > merged.txt
```



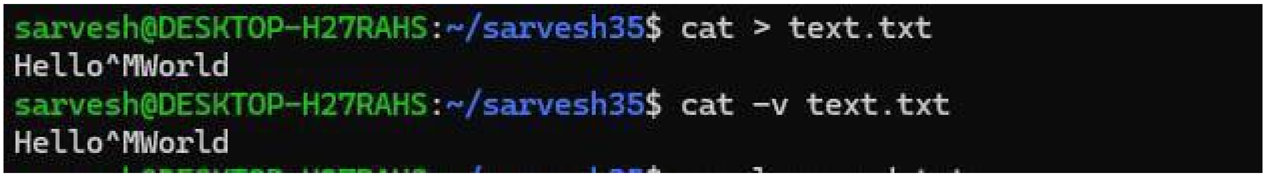
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat merged.txt
hello sir from sarvesh
how are you
excited to learn linux os
```

15. Display the contents of a file `data.txt` with line numbers using `cat -n`.



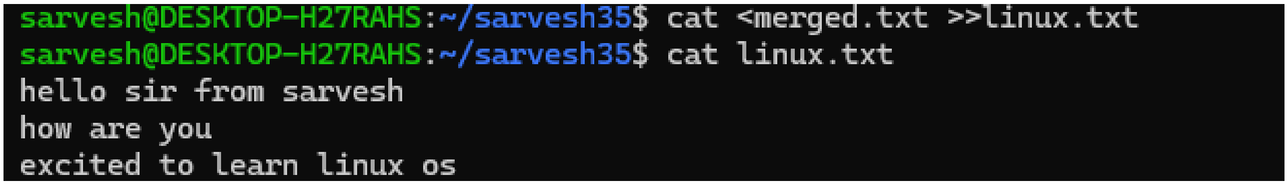
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat -n merged.txt
     1  hello sir from sarvesh
     2  how are you
     3  excited to learn linux os
```

16. View a file's contents with non-printable characters shown as `^` using `cat -v`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat > text.txt
Hello^MWorld
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat -v text.txt
Hello^MWorld
```

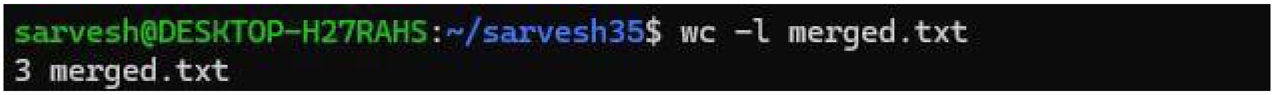
17. Use `cat` with redirection to create a file and append content in one step.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat <merged.txt >>linux.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat linux.txt
hello sir from sarvesh
how are you
excited to learn linux os
```

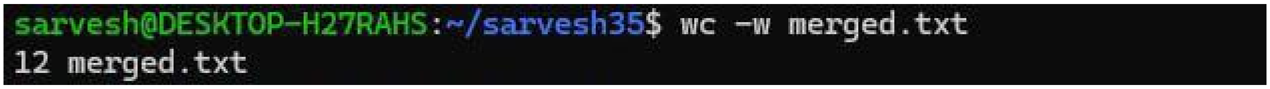
`wc`

18. Count the number of lines in `data.txt` using `wc -l`.



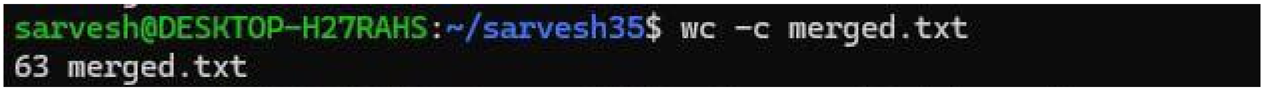
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc -l merged.txt
3 merged.txt
```

19. Use `wc -w` to count the words in `notes.txt`.



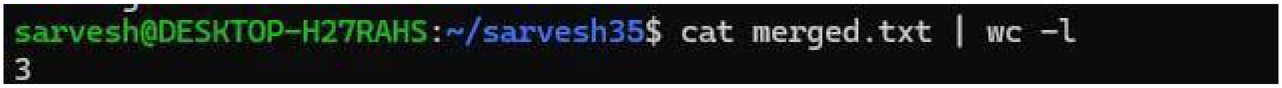
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc -w merged.txt
12 merged.txt
```

20. Calculate the number of characters in `notes.txt` using `wc -c`.



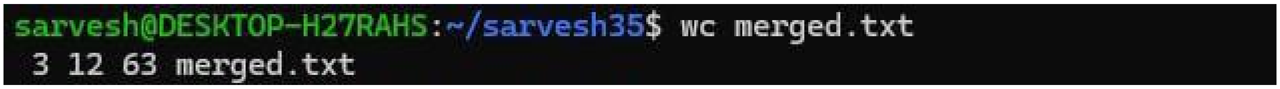
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc -c merged.txt
63 merged.txt
```

21. Combine `cat` and `wc` to count lines in a file displayed on the terminal.



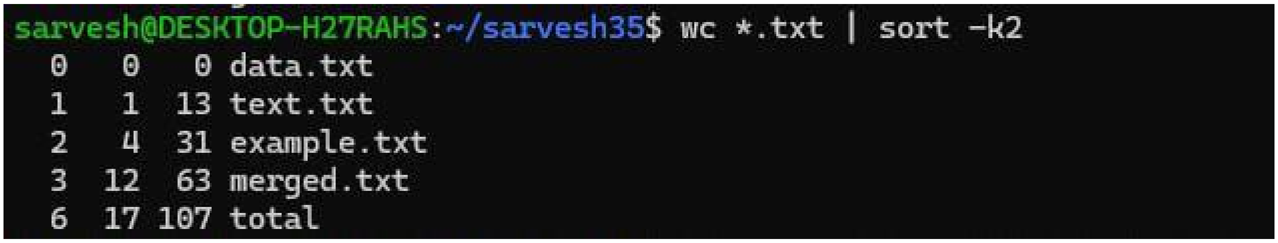
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat merged.txt | wc -l
3
```

22. Use `wc` to display the total number of lines, words, and characters in a file.



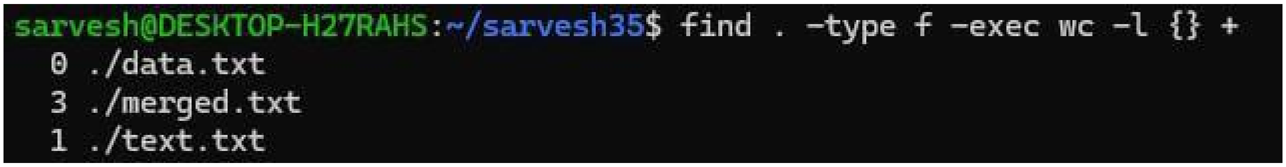
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc merged.txt
 3 12 63 merged.txt
```

23. Apply `wc` to all `.txt` files in the current directory and sort the results by word count.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc *.txt | sort -k2
  0   0   0 data.txt
  1   1  13 text.txt
  2   4  31 example.txt
  3  12  63 merged.txt
  6  17 107 total
```

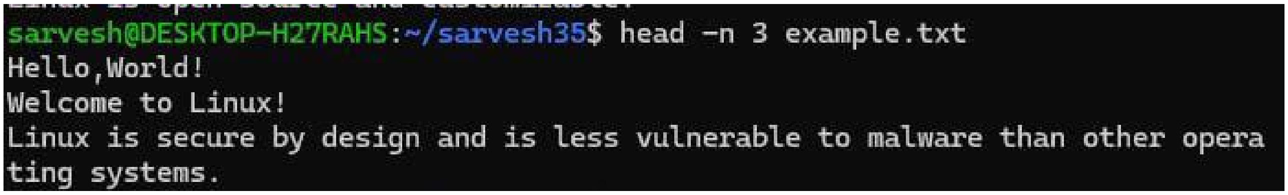
24. Count the lines in a directory's files using `find` and `wc`



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -exec wc -l {} +
  0 ./data.txt
  3 ./merged.txt
  1 ./text.txt
```

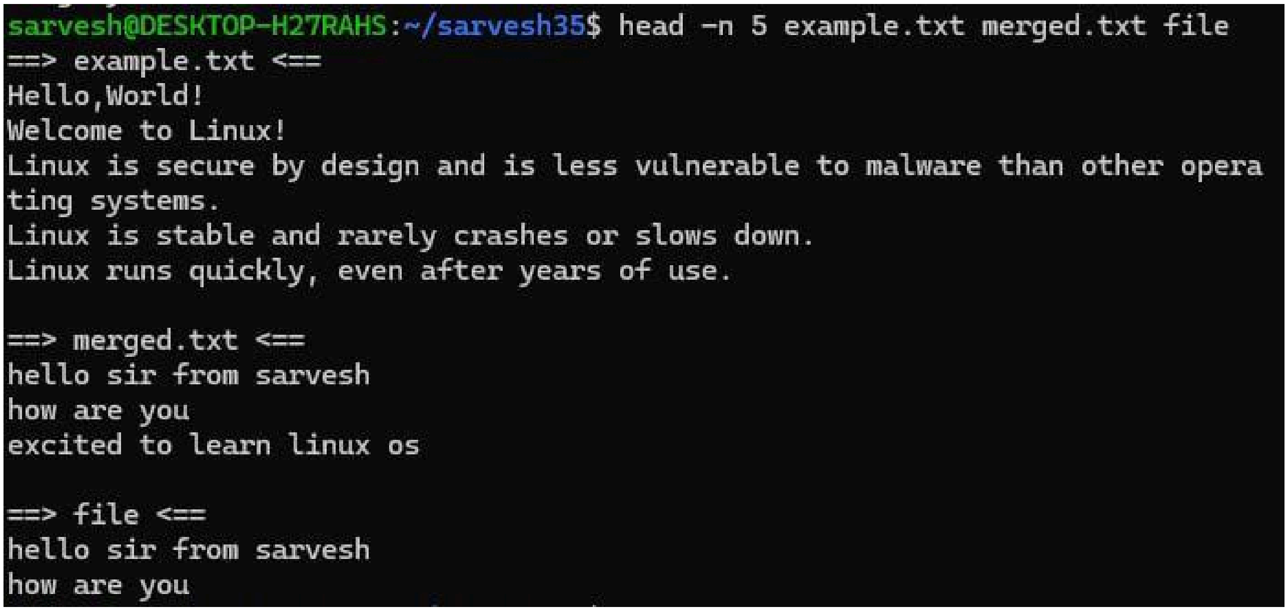
head

25. Display the first 3 lines of `example.txt` using `head -n 3`.



```
Linux is open source and customizable.
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ head -n 3 example.txt
Hello,World!
Welcome to Linux!
Linux is secure by design and is less vulnerable to malware than other opera
ting systems.
```

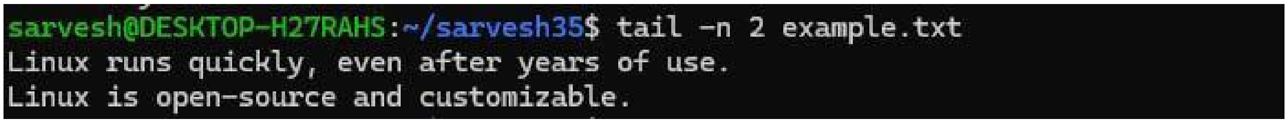
26. Use `head` with multiple files to show their first 5 lines.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ head -n 5 example.txt merged.txt file
==> example.txt <==
Hello,World!
Welcome to Linux!
Linux is secure by design and is less vulnerable to malware than other opera
ting systems.
Linux is stable and rarely crashes or slows down.
Linux runs quickly, even after years of use.

==> merged.txt <==
hello sir from sarvesh
how are you
excited to learn linux os

==> file <==
hello sir from sarvesh
how are you
```

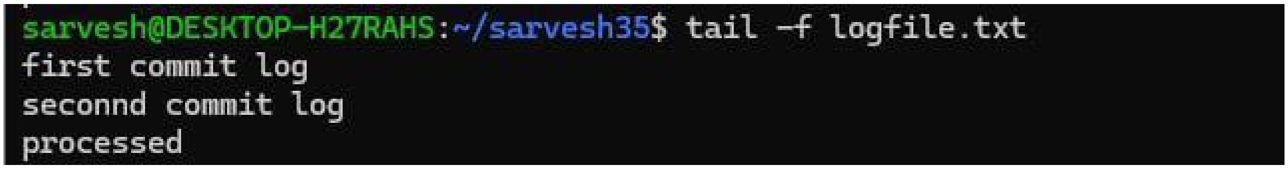
`tail`

27. View the last 2 lines of `logfile.txt` using `tail`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ tail -n 2 example.txt
Linux runs quickly, even after years of use.
Linux is open-source and customizable.
```

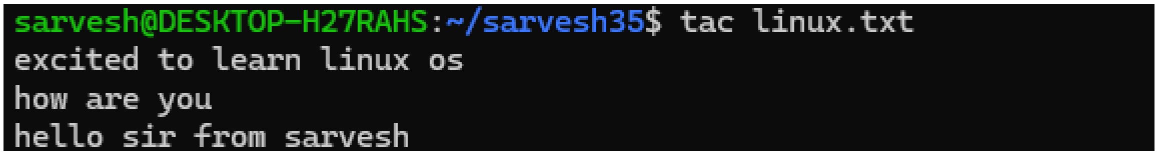
28. Monitor a log file in real time using `tail -f logfile.txt`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ tail -f logfile.txt
first commit log
seconnd commit log
processed
```

`tac`

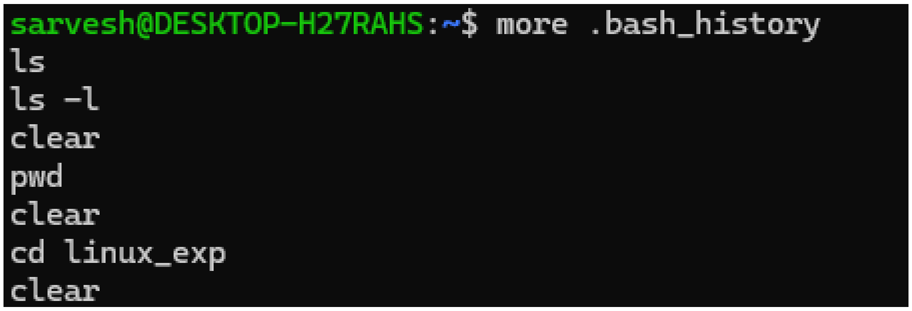
29. Reverse the lines in `data.txt` using `tac`.



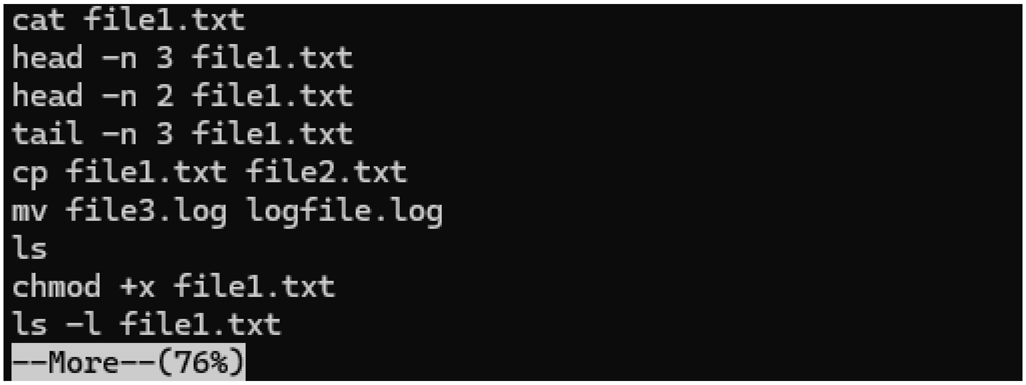
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ tac linux.txt
excited to learn linux os
how are you
hello sir from sarvesh
```

`more` and `less`

30. Use `more` to view a large file and scroll through it line by line.

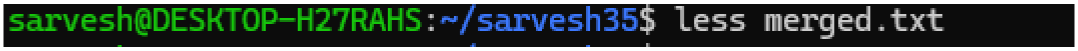


```
sarvesh@DESKTOP-H27RAHS:~$ more .bash_history
ls
ls -l
clear
pwd
clear
cd linux_exp
clear
```

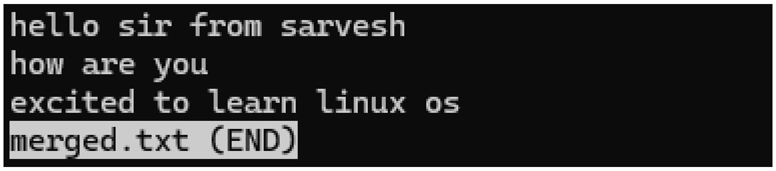


```
cat file1.txt
head -n 3 file1.txt
head -n 2 file1.txt
tail -n 3 file1.txt
cp file1.txt file2.txt
mv file3.log logfile.log
ls
chmod +x file1.txt
ls -l file1.txt
--More--(76%)
```

31. Navigate backward and forward in a file using less.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ less merged.txt
```

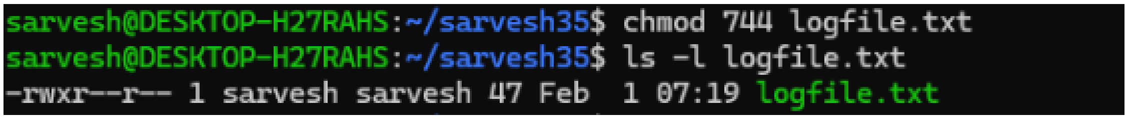


```
hello sir from sarvesh
how are you
excited to learn linux os
merged.txt (END)
```

q

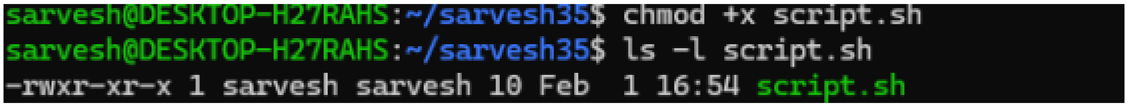
3. File Permissions (chmod)

32. Use chmod 744 to give read/write/execute permissions to the owner and read-only permissions to others.



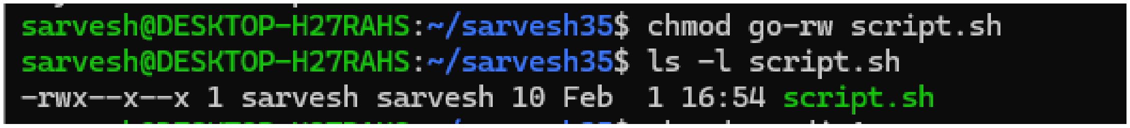
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod 744 logfile.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l logfile.txt
-rwxr--r-- 1 sarvesh sarvesh 47 Feb  1 07:19 logfile.txt
```

33. Change the permissions of script.sh to make it executable by everyone.



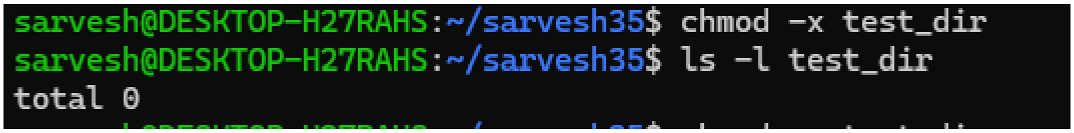
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod +x script.sh
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l script.sh
-rwxr-xr-x 1 sarvesh sarvesh 10 Feb  1 16:54 script.sh
```

34. Set file1.txt to have read/write permissions for the owner only using symbolic notation.



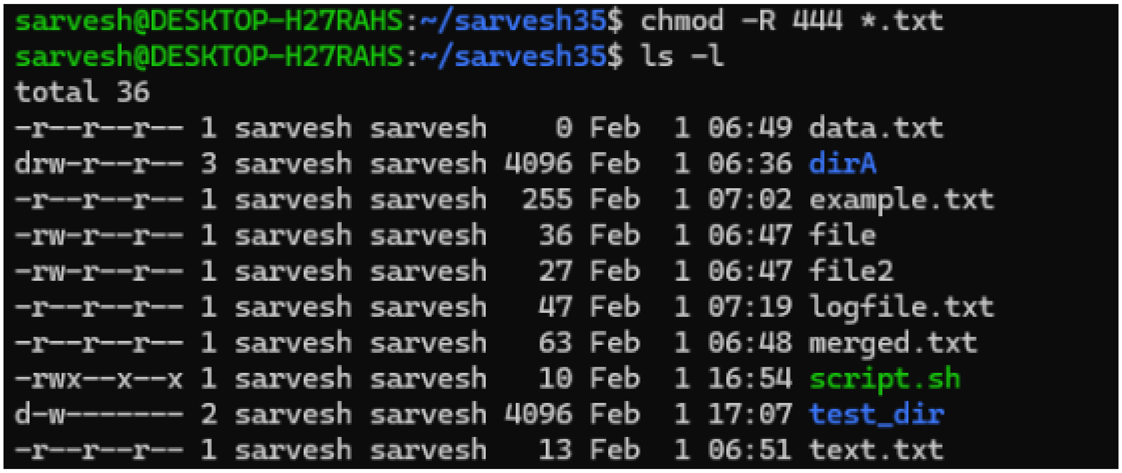
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod go-rw script.sh
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l script.sh
-rwx--x--x 1 sarvesh sarvesh 10 Feb  1 16:54 script.sh
```

35. Remove execute permissions from a directory test_dir using chmod.



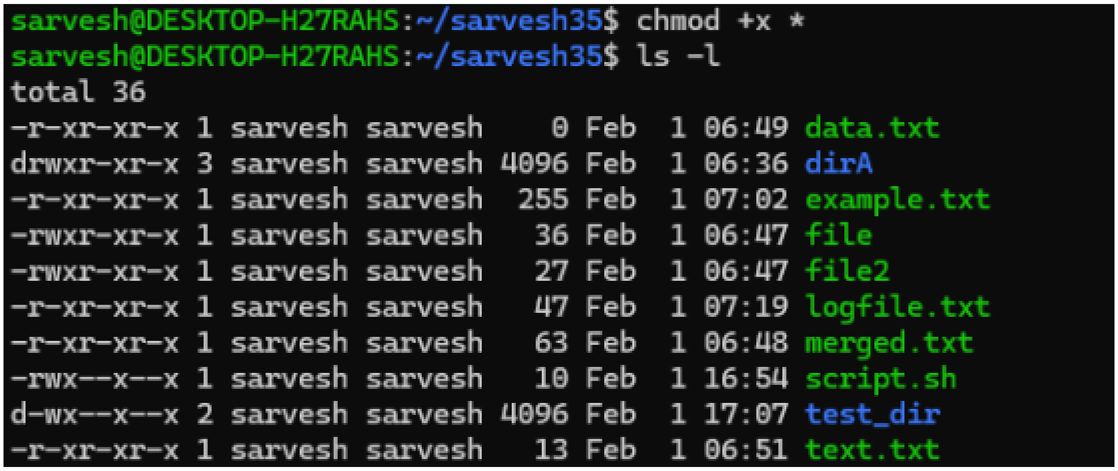
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod -x test_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l test_dir
total 0
```

36. Use `chmod` recursively to set read-only permissions for all `.txt` files in a directory.



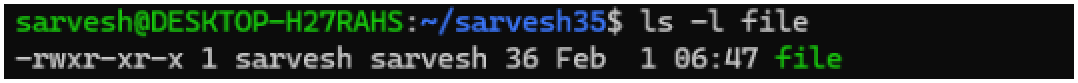
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod -R 444 *.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l
total 36
-r--r--r-- 1 sarvesh sarvesh    0 Feb  1 06:49 data.txt
drw-r--r-- 3 sarvesh sarvesh 4096 Feb  1 06:36 dirA
-r--r--r-- 1 sarvesh sarvesh  255 Feb  1 07:02 example.txt
-rw-r--r-- 1 sarvesh sarvesh   36 Feb  1 06:47 file
-rw-r--r-- 1 sarvesh sarvesh   27 Feb  1 06:47 file2
-r--r--r-- 1 sarvesh sarvesh   47 Feb  1 07:19 logfile.txt
-r--r--r-- 1 sarvesh sarvesh   63 Feb  1 06:48 merged.txt
-rwx--x--x 1 sarvesh sarvesh   10 Feb  1 16:54 script.sh
d-w------- 2 sarvesh sarvesh 4096 Feb  1 17:07 test_dir
-r--r--r-- 1 sarvesh sarvesh   13 Feb  1 06:51 text.txt
```

37. Add execute permissions to all files in a directory using `chmod +x *`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod +x *
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l
total 36
-r-xr-xr-x 1 sarvesh sarvesh    0 Feb  1 06:49 data.txt
drwxr-xr-x 3 sarvesh sarvesh 4096 Feb  1 06:36 dirA
-r-xr-xr-x 1 sarvesh sarvesh  255 Feb  1 07:02 example.txt
-rwxr-xr-x 1 sarvesh sarvesh   36 Feb  1 06:47 file
-rwxr-xr-x 1 sarvesh sarvesh   27 Feb  1 06:47 file2
-r-xr-xr-x 1 sarvesh sarvesh   47 Feb  1 07:19 logfile.txt
-r-xr-xr-x 1 sarvesh sarvesh   63 Feb  1 06:48 merged.txt
-rwx--x--x 1 sarvesh sarvesh   10 Feb  1 16:54 script.sh
d-wx--x--x 2 sarvesh sarvesh 4096 Feb  1 17:07 test_dir
-r-xr-xr-x 1 sarvesh sarvesh   13 Feb  1 06:51 text.txt
```

38. Display the permissions of `file.txt` using `ls -l` and interpret them.

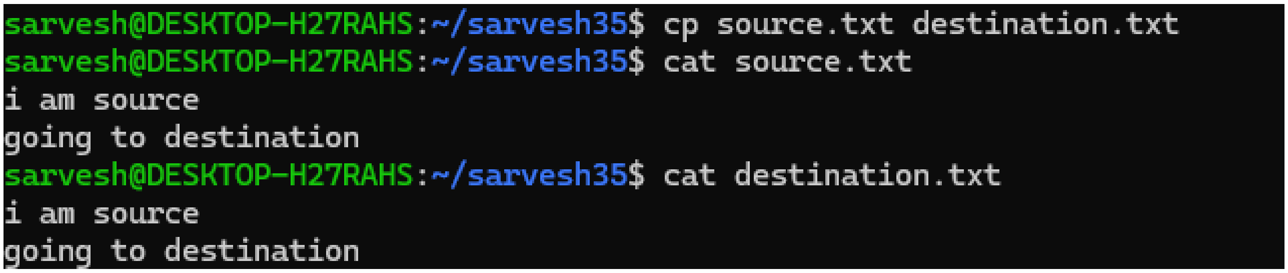


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l file
-rwxr-xr-x 1 sarvesh sarvesh 36 Feb  1 06:47 file
```

4. File and Directory Copying and Moving

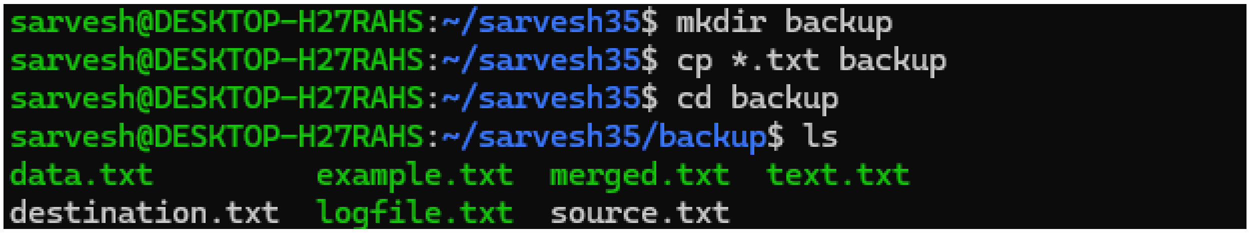
(cp, mv) cp

39. Copy a file `source.txt` to `destination.txt` using `cp`.



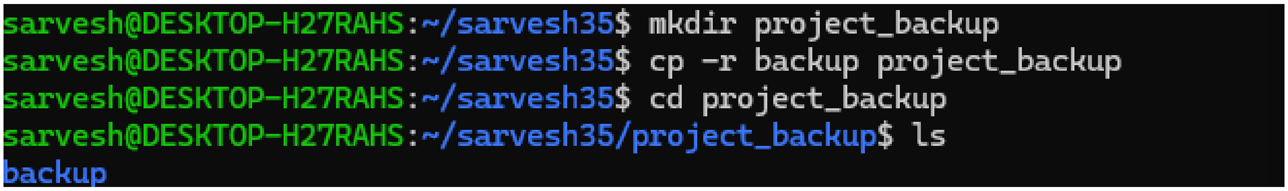
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp source.txt destination.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat source.txt
i am source
going to destination
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat destination.txt
i am source
going to destination
```

40. Copy all `.txt` files from the current directory to `backup/` using `cp`.



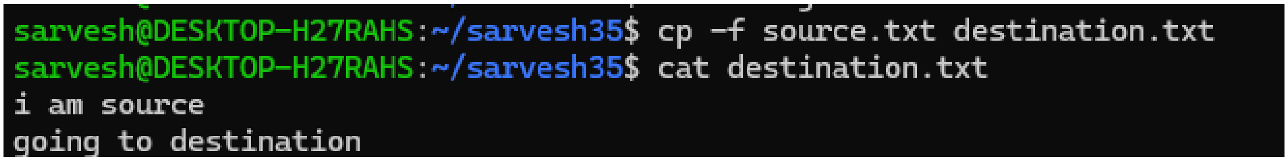
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mkdir backup
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp *.txt backup
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cd backup
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/backup$ ls
data.txt         example.txt  merged.txt  text.txt
destination.txt  logfile.txt  source.txt
```

41. Use `cp -r` to copy an entire directory `project` to `project_backup`.



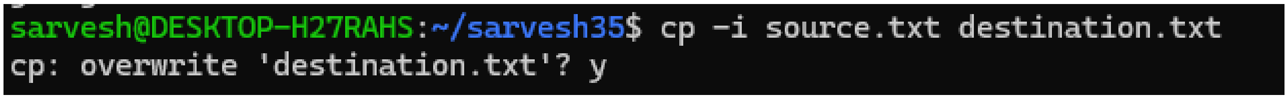
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mkdir project_backup
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp -r backup project_backup
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cd project_backup
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/project_backup$ ls
backup
```

42. Overwrite a file without confirmation using `cp`.



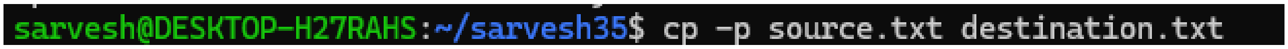
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp -f source.txt destination.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat destination.txt
i am source
going to destination
```

43. Use `cp -i` to enable confirmation before overwriting files.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp -i source.txt destination.txt
cp: overwrite 'destination.txt'? y
```

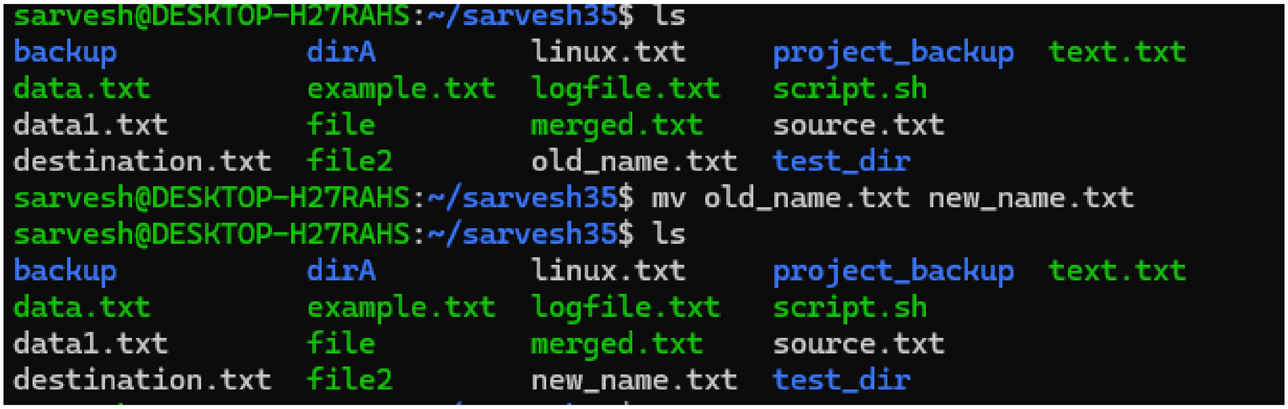
44. Copy files and preserve their attributes using `cp -p`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp -p source.txt destination.txt
```

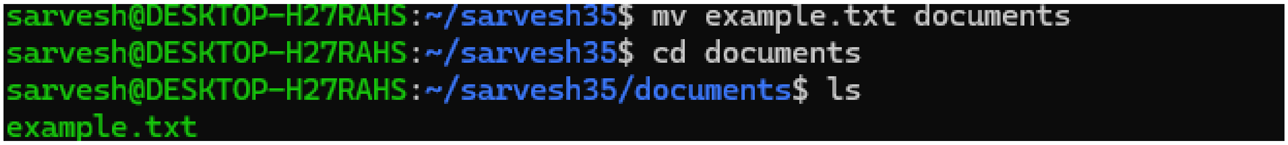
mv

45. Rename `old_name.txt` to `new_name.txt` using `mv`.



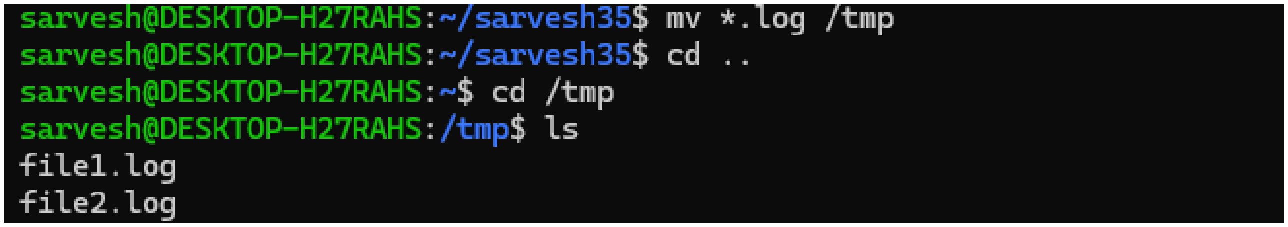
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
backup           dirA          linux.txt     project_backup  text.txt
data.txt         example.txt   logfile.txt   script.sh
data1.txt        file          merged.txt    source.txt
destination.txt  file2         old_name.txt  test_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mv old_name.txt new_name.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
backup           dirA          linux.txt     project_backup  text.txt
data.txt         example.txt   logfile.txt   script.sh
data1.txt        file          merged.txt    source.txt
destination.txt  file2         new_name.txt  test_dir
```

46. Move `example.txt` to the `documents/` directory using `mv`.



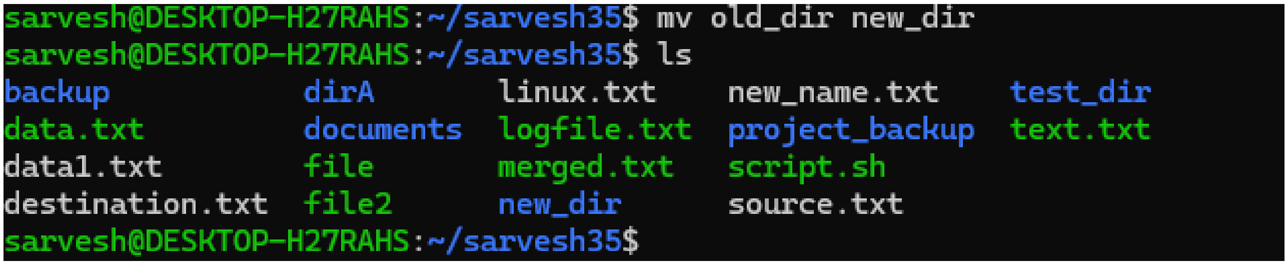
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mv example.txt documents
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cd documents
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/documents$ ls
example.txt
```

47. Use `mv` to relocate all `.log` files to the `/tmp` directory.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mv *.log /tmp
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cd ..
sarvesh@DESKTOP-H27RAHS:~$ cd /tmp
sarvesh@DESKTOP-H27RAHS:/tmp$ ls
file1.log
file2.log
```

48. Move a directory `old_dir` to a new location `new_dir`.

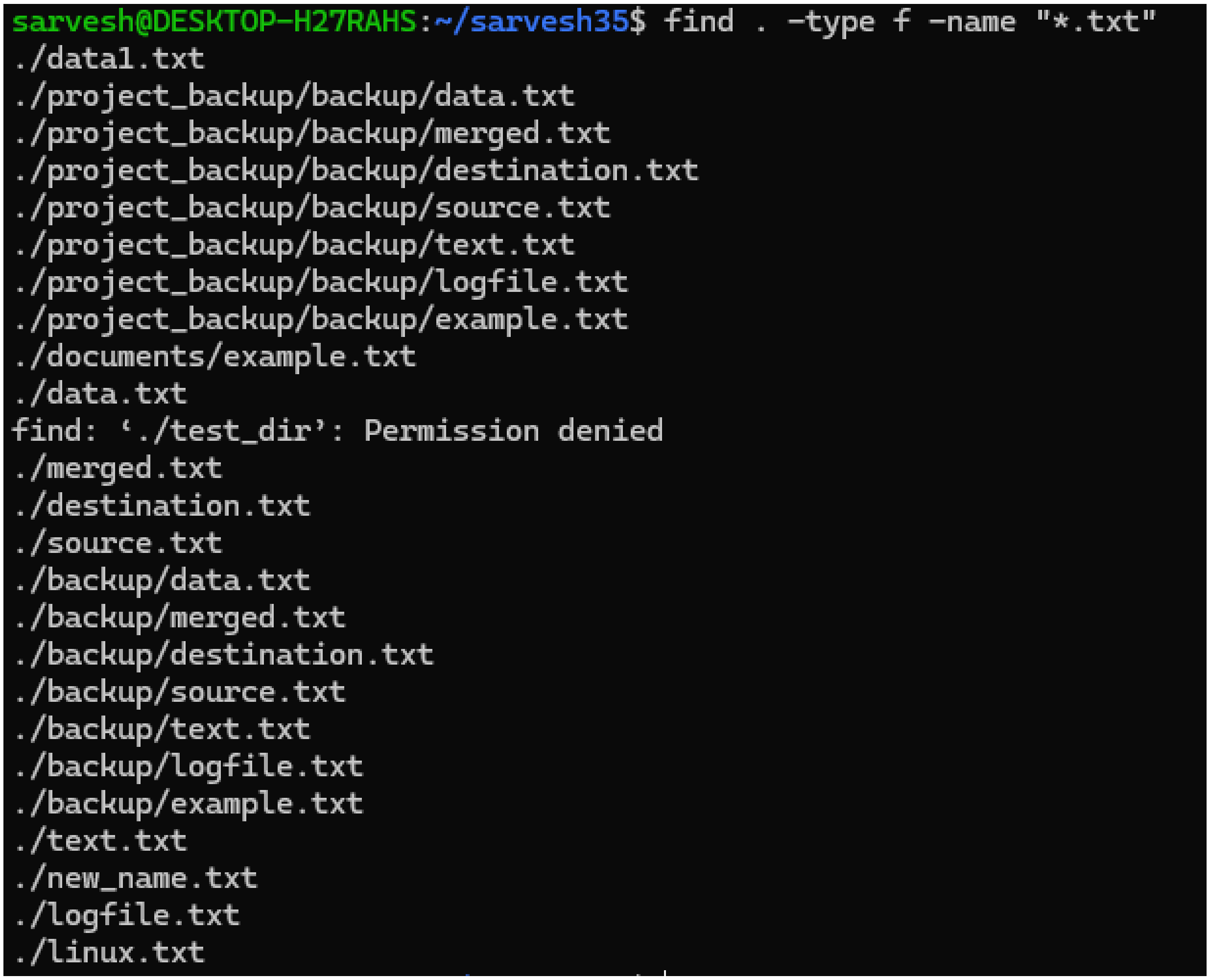


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mv old_dir new_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
backup           dirA       linux.txt    new_name.txt    test_dir
data.txt         documents  logfile.txt  project_backup  text.txt
data1.txt        file       merged.txt   script.sh
destination.txt  file2      new_dir      source.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

5. Searching and History (**`find`**,

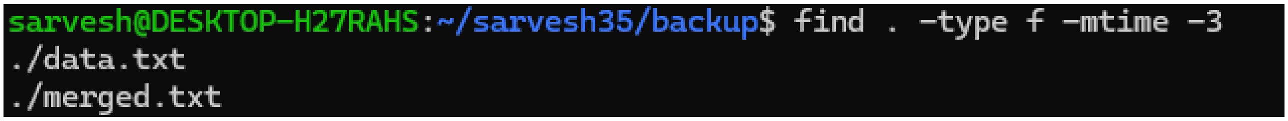
`history`) **`find`**

49. Find all `.txt` files in the current directory using `find`.



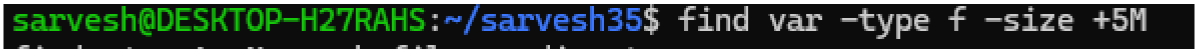
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -name "*.txt"
./data1.txt
./project_backup/backup/data.txt
./project_backup/backup/merged.txt
./project_backup/backup/destination.txt
./project_backup/backup/source.txt
./project_backup/backup/text.txt
./project_backup/backup/logfile.txt
./project_backup/backup/example.txt
./documents/example.txt
./data.txt
find: './test_dir': Permission denied
./merged.txt
./destination.txt
./source.txt
./backup/data.txt
./backup/merged.txt
./backup/destination.txt
./backup/source.txt
./backup/text.txt
./backup/logfile.txt
./backup/example.txt
./text.txt
./new_name.txt
./logfile.txt
./linux.txt
```

50. Locate files modified within the last 3 days using `find`.



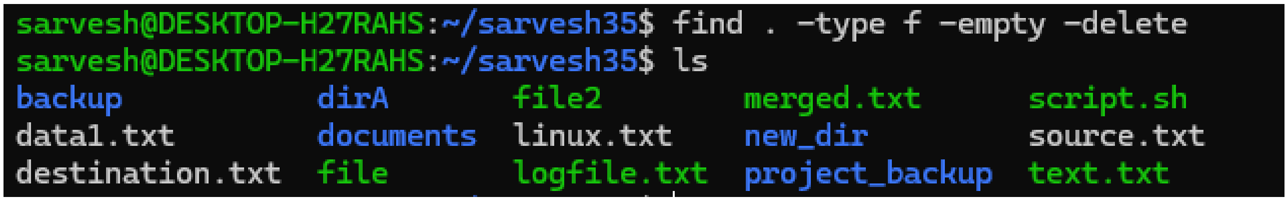
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/backup$ find . -type f -mtime -3
./data.txt
./merged.txt
```

51. Search for files larger than 5 MB in the `/var` directory using `find`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find var -type f -size +5M
```

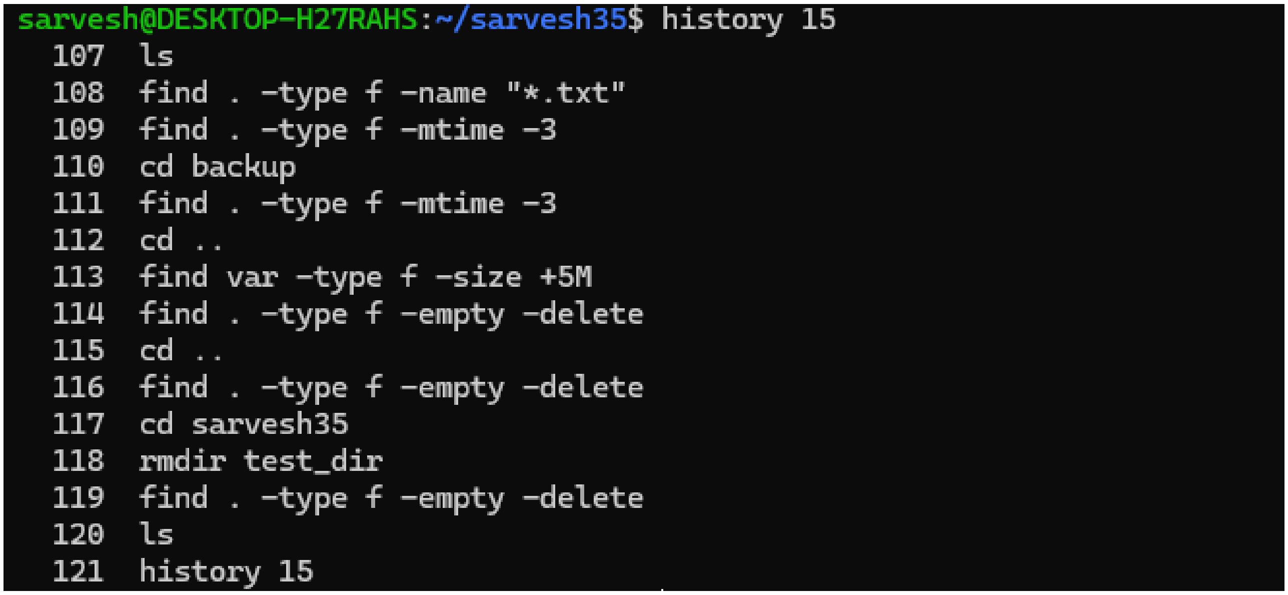
52. Use `find` to locate empty files and delete them.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -empty -delete
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
backup           dirA       file2        merged.txt      script.sh
data1.txt        documents  linux.txt    new_dir         source.txt
destination.txt  file       logfile.txt  project_backup  text.txt
```

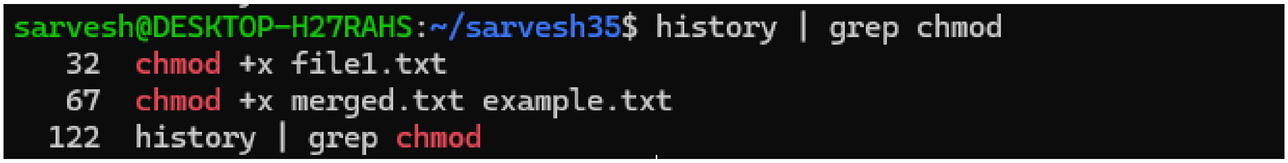
`history`

53. Display the last 15 commands executed using `history`.



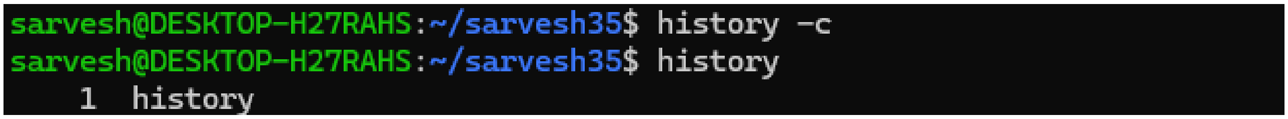
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ history 15
  107  ls
  108  find . -type f -name "*.txt"
  109  find . -type f -mtime -3
  110  cd backup
  111  find . -type f -mtime -3
  112  cd ..
  113  find var -type f -size +5M
  114  find . -type f -empty -delete
  115  cd ..
  116  find . -type f -empty -delete
  117  cd sarvesh35
  118  rmdir test_dir
  119  find . -type f -empty -delete
  120  ls
  121  history 15
```

54. Search your history for commands related to `chmod`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ history | grep chmod
   32  chmod +x file1.txt
   67  chmod +x merged.txt example.txt
  122  history | grep chmod
```

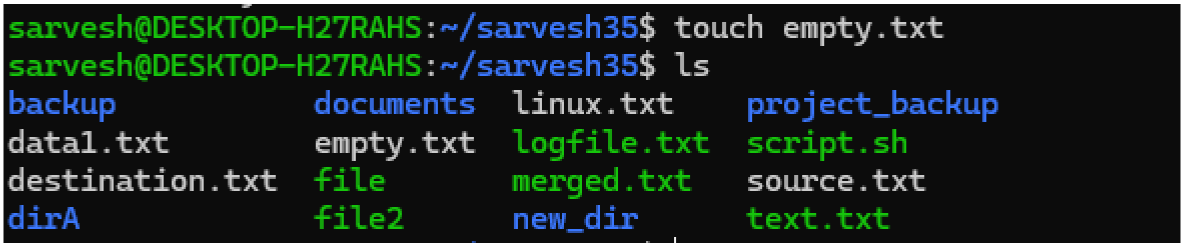
55. Clear the command history for the current session using `history -c`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ history -c
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ history
    1  history
```

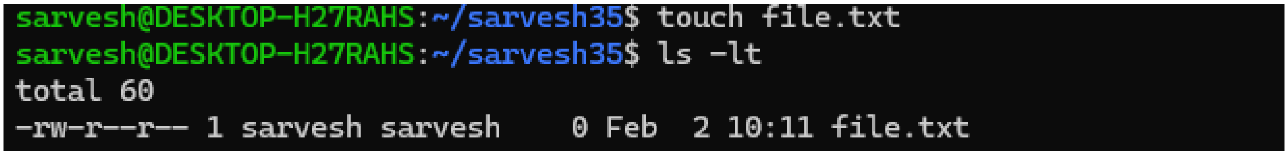
6. Miscellaneous Commands (`touch`, `man`, `clear`)

56. Create an empty file `empty.txt` using `touch`.



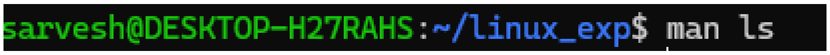
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ touch empty.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
backup           documents  linux.txt    project_backup
data1.txt        empty.txt  logfile.txt  script.sh
destination.txt  file       merged.txt   source.txt
dirA             file2      new_dir      text.txt
```

57. Update the timestamp of `file.txt` using `touch`.

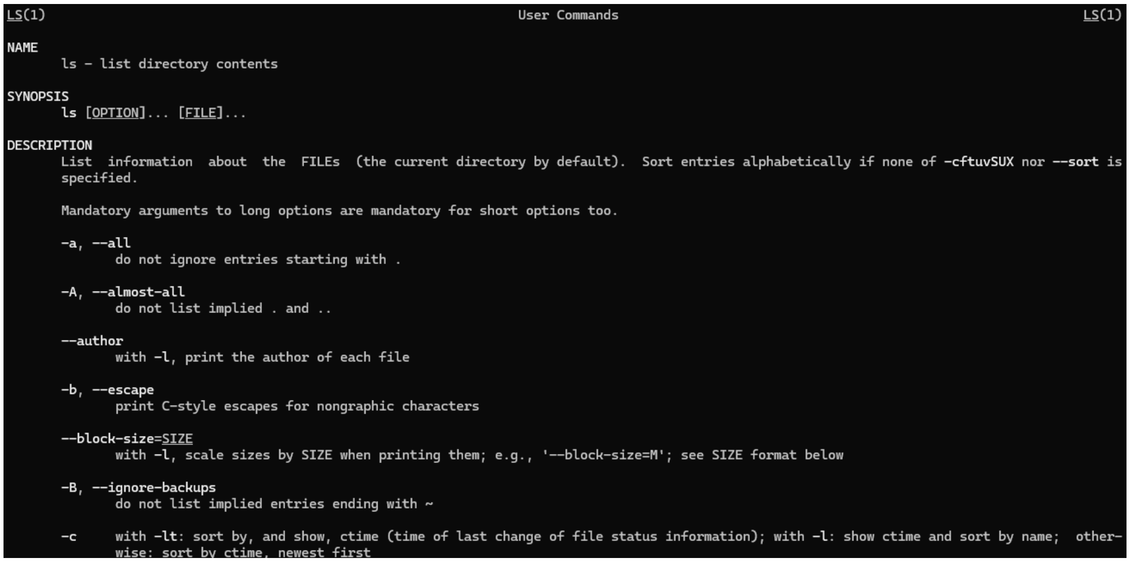


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ touch file.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -lt
total 60
-rw-r--r-- 1 sarvesh sarvesh    0 Feb  2 10:11 file.txt
```

58. Use `man` to open the manual for the `ls` command.

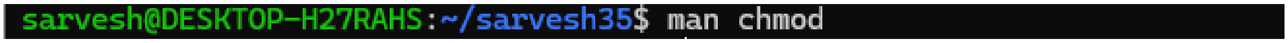


```
sarvesh@DESKTOP-H27RAHS:~/linux_exp$ man ls
```

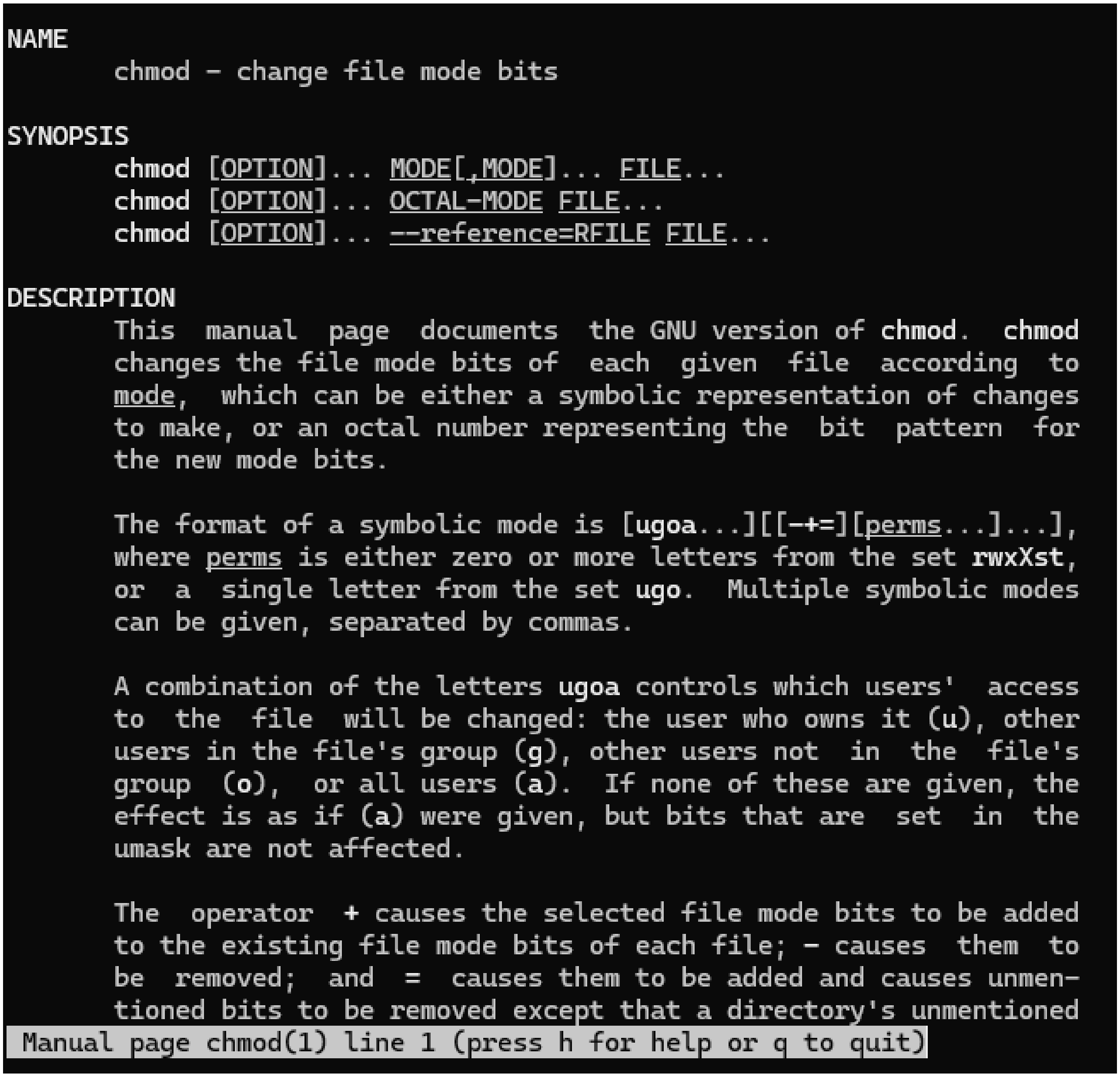


```
LS(1)                                   User Commands                                   LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the  FILEs  (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX nor --sort is
       specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last change of file status information); with -l: show ctime and sort by name;  other-
              wise: sort by ctime, newest first
```

59. Find out more about the `chmod` command using `man`

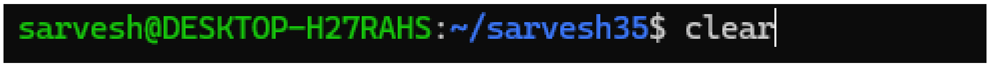


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ man chmod
```



```
NAME
       chmod - change file mode bits

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This  manual  page  documents  the GNU version of chmod.  chmod
       changes the file mode bits of  each  given  file  according  to
       mode,  which can be either a symbolic representation of changes
       to make, or an octal number representing the  bit  pattern  for
       the new mode bits.

       The format of a symbolic mode is [ugoa...][[-+=][perms...]...],
       where perms is either zero or more letters from the set rwxXst,
       or  a  single letter from the set ugo.  Multiple symbolic modes
       can be given, separated by commas.

       A combination of the letters ugoa controls which users'  access
       to  the  file  will be changed: the user who owns it (u), other
       users in the file's group (g), other users not  in  the  file's
       group  (o),  or all users (a).  If none of these are given, the
       effect is as if (a) were given, but bits that are  set  in  the
       umask are not affected.

       The  operator  + causes the selected file mode bits to be added
       to the existing file mode bits of each file; - causes  them  to
       be  removed;  and  =  causes them to be added and causes unmen-
       tioned bits to be removed except that a directory's unmentioned
 Manual page chmod(1) line 1 (press h for help or q to quit)
```

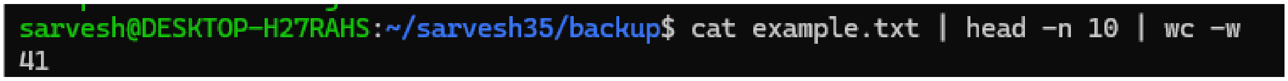
60. Clear the terminal screen using `clear`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ clear
```

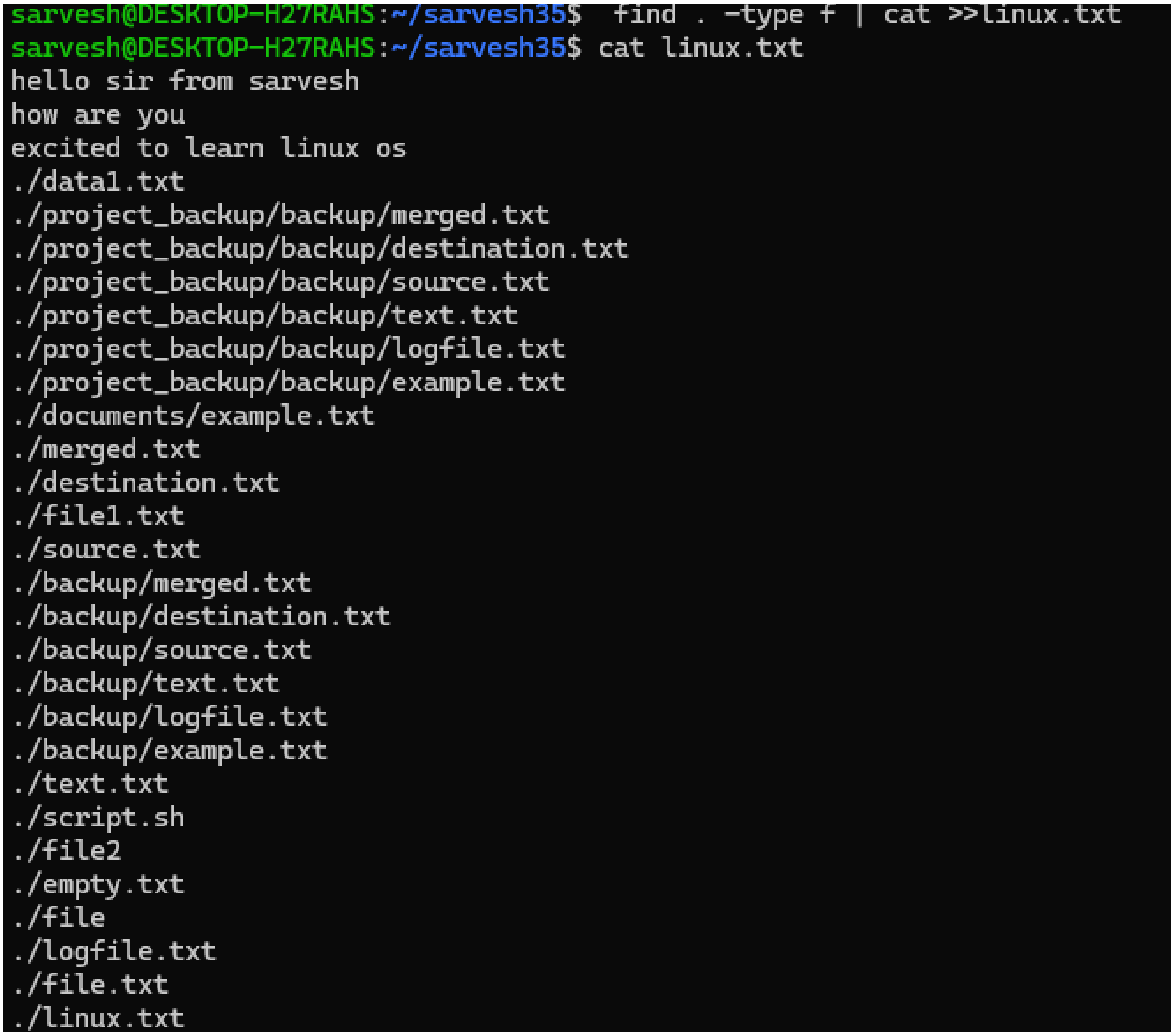
7. Combined Tasks (Focus on `cat`, `wc`, `chmod`, `cp`, `mv`)

61. Combine `cat` and `wc` to count the number of words in the first 10 lines of `data.txt`.



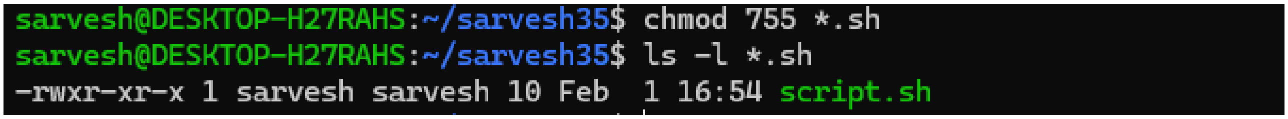
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35/backup$ cat example.txt | head -n 10 | wc -w
41
```

62. Use `find` to locate all `.txt` files and append their contents to `merged.txt` using `cat`.



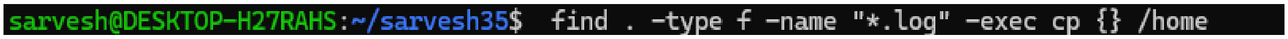
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$  find . -type f | cat >>linux.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat linux.txt
hello sir from sarvesh
how are you
excited to learn linux os
./data1.txt
./project_backup/backup/merged.txt
./project_backup/backup/destination.txt
./project_backup/backup/source.txt
./project_backup/backup/text.txt
./project_backup/backup/logfile.txt
./project_backup/backup/example.txt
./documents/example.txt
./merged.txt
./destination.txt
./file1.txt
./source.txt
./backup/merged.txt
./backup/destination.txt
./backup/source.txt
./backup/text.txt
./backup/logfile.txt
./backup/example.txt
./text.txt
./script.sh
./file2
./empty.txt
./file
./logfile.txt
./file.txt
./linux.txt
```

63. Use `chmod` to set permissions of all `.sh` files in a directory to `755`.



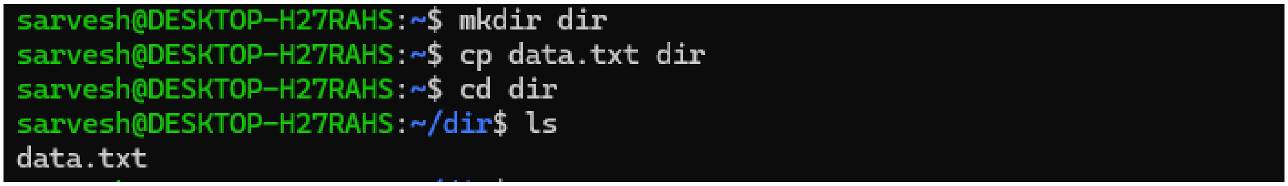
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod 755 *.sh
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l *.sh
-rwxr-xr-x 1 sarvesh sarvesh 10 Feb  1 16:54 script.sh
```

64. Combine `cp` and `find` to copy all `.log` files to a new directory.



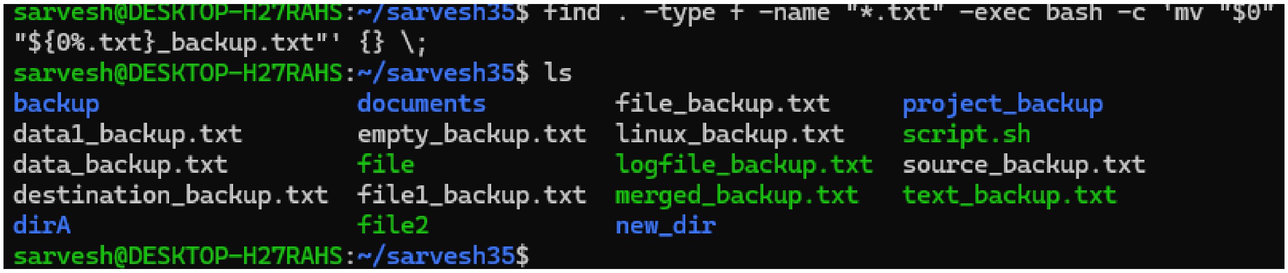
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$  find . -type f -name "*.log" -exec cp {} /home
```

65. Create a directory, copy a file into it, and then move it to another location.



```
sarvesh@DESKTOP-H27RAHS:~$ mkdir dir
sarvesh@DESKTOP-H27RAHS:~$ cp data.txt dir
sarvesh@DESKTOP-H27RAHS:~$ cd dir
sarvesh@DESKTOP-H27RAHS:~/dir$ ls
data.txt
```

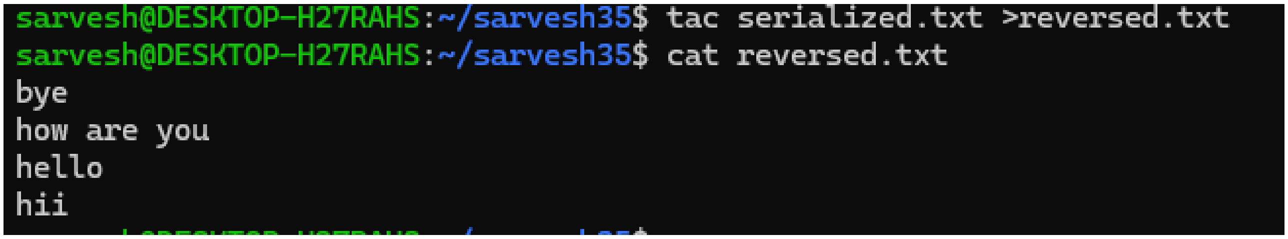
66. Use `mv` to rename all `.txt` files in a directory by adding a `_backup` suffix.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -name "*.txt" -exec bash -c 'mv "$0"
"${0%.txt}_backup.txt"' {} \;
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls
backup                  documents          file_backup.txt     project_backup
data1_backup.txt        empty_backup.txt   linux_backup.txt    script.sh
data_backup.txt         file               logfile_backup.txt  source_backup.txt
destination_backup.txt  file1_backup.txt   merged_backup.txt   text_backup.txt
dirA                    file2              new_dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

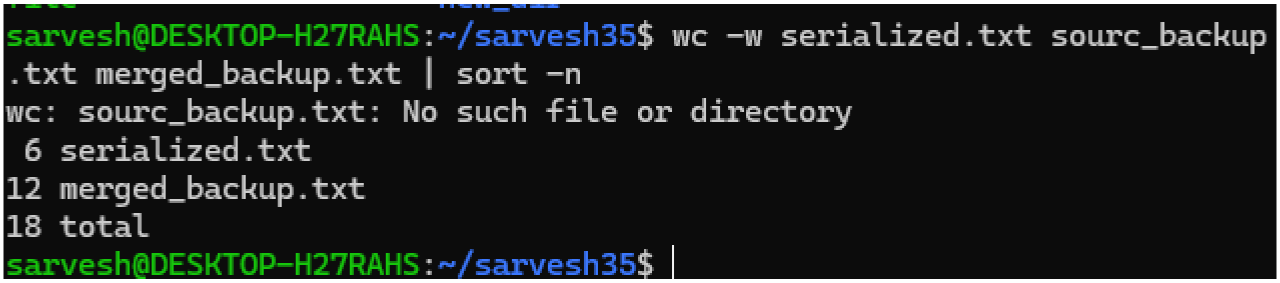
8. Additional Exercises

67. Reverse the contents of `data.txt` and save the result into `reversed.txt`.



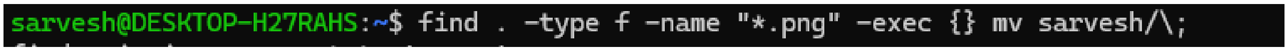
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ tac serialized.txt >reversed.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat reversed.txt
bye
how are you
hello
hii
```

68. Count the total words in multiple files and sort the result.



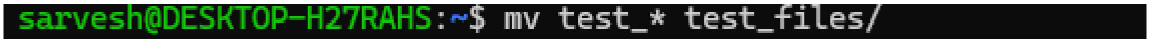
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc -w serialized.txt sourc_backup
.txt merged_backup.txt | sort -n
wc: sourc_backup.txt: No such file or directory
 6 serialized.txt
12 merged_backup.txt
18 total
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ 
```

69. Copy only `.png` files from a directory to another directory.

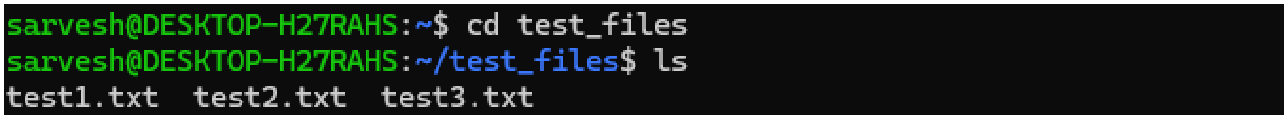


```
sarvesh@DESKTOP-H27RAHS:~$ find . -type f -name "*.png" -exec {} mv sarvesh/\;
```

70. Move all files starting with `test_` to a folder named `test_files`.

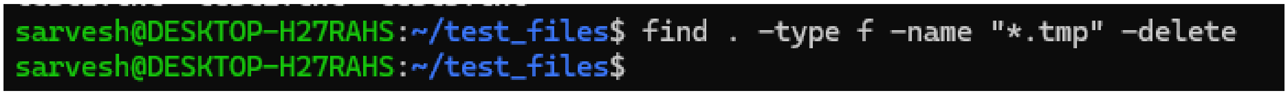


```
sarvesh@DESKTOP-H27RAHS:~$ mv test_* test_files/
```



```
sarvesh@DESKTOP-H27RAHS:~$ cd test_files
sarvesh@DESKTOP-H27RAHS:~/test_files$ ls
test1.txt  test2.txt  test3.txt
```

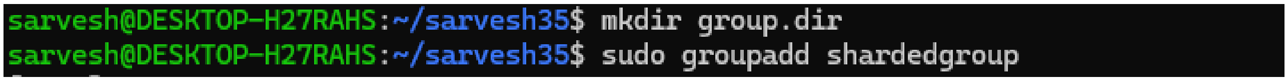
71. Find and delete all `.tmp` files from the current directory using `find`.



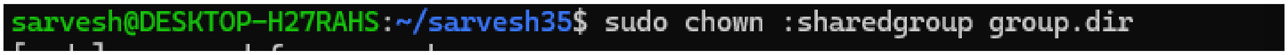
```
sarvesh@DESKTOP-H27RAHS:~/test_files$ find . -type f -name "*.tmp" -delete
sarvesh@DESKTOP-H27RAHS:~/test_files$
```

9. Advanced Challenges

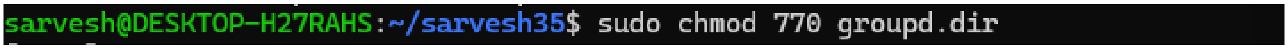
72. Use chmod to set group ownership and permissions for a shared directory.



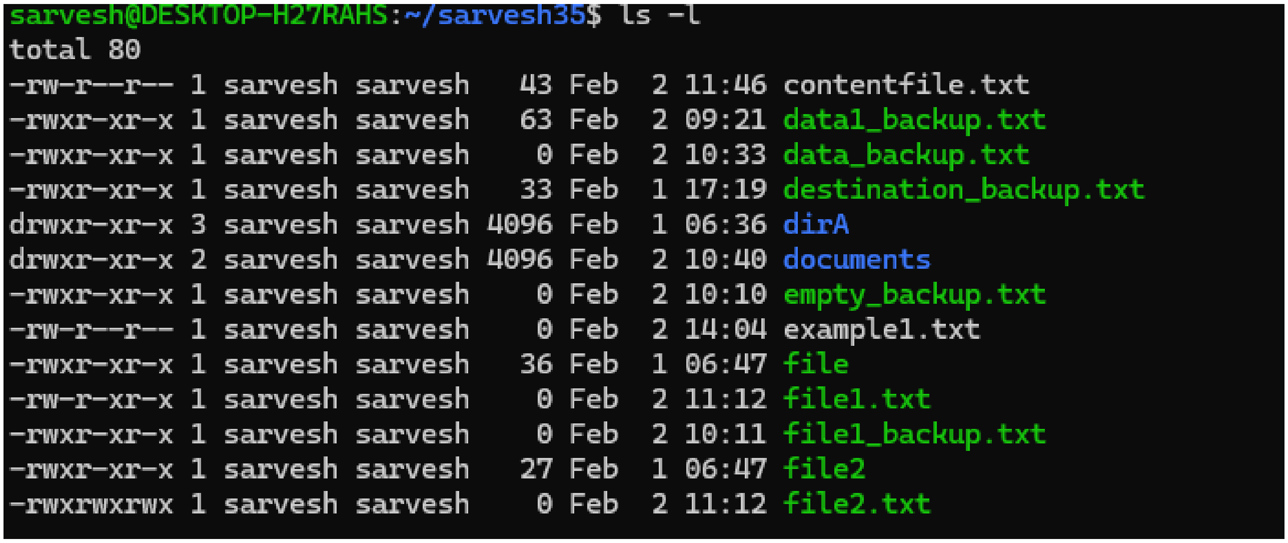
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ mkdir group.dir
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ sudo groupadd shardedgroup
```



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ sudo chown :sharedgroup group.dir
```

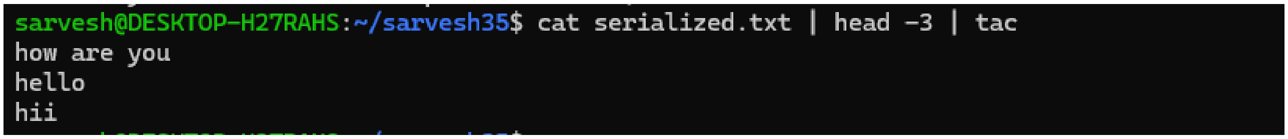


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ sudo chmod 770 groupd.dir
```



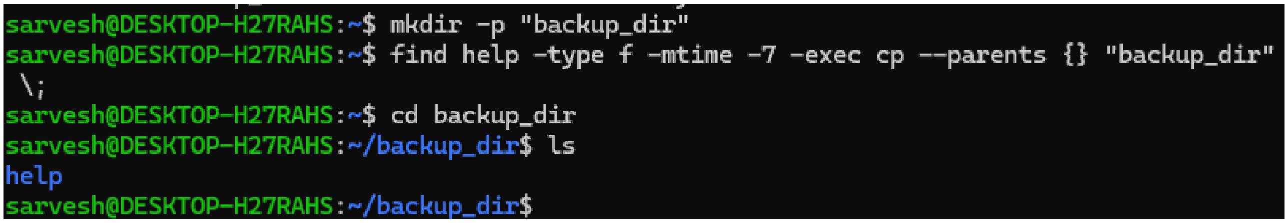
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -l
total 80
-rw-r--r-- 1 sarvesh sarvesh   43 Feb  2 11:46 contentfile.txt
-rwxr-xr-x 1 sarvesh sarvesh   63 Feb  2 09:21 data1_backup.txt
-rwxr-xr-x 1 sarvesh sarvesh    0 Feb  2 10:33 data_backup.txt
-rwxr-xr-x 1 sarvesh sarvesh   33 Feb  1 17:19 destination_backup.txt
drwxr-xr-x 3 sarvesh sarvesh 4096 Feb  1 06:36 dirA
drwxr-xr-x 2 sarvesh sarvesh 4096 Feb  2 10:40 documents
-rwxr-xr-x 1 sarvesh sarvesh    0 Feb  2 10:10 empty_backup.txt
-rw-r--r-- 1 sarvesh sarvesh    0 Feb  2 14:04 example1.txt
-rwxr-xr-x 1 sarvesh sarvesh   36 Feb  1 06:47 file
-rw-r-xr-x 1 sarvesh sarvesh    0 Feb  2 11:12 file1.txt
-rwxr-xr-x 1 sarvesh sarvesh    0 Feb  2 10:11 file1_backup.txt
-rwxr-xr-x 1 sarvesh sarvesh   27 Feb  1 06:47 file2
-rwxrwxrwx 1 sarvesh sarvesh    0 Feb  2 11:12 file2.txt
```

73. Combine cat, tac, and head to display the first 3 lines of a file in reverse order.



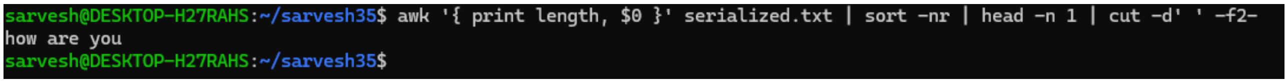
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat serialized.txt | head -3 | tac
how are you
hello
hii
```

74. Create a backup script that uses cp and find to archive files modified within the last 7 days.



```
sarvesh@DESKTOP-H27RAHS:~$ mkdir -p "backup_dir"
sarvesh@DESKTOP-H27RAHS:~$ find help -type f -mtime -7 -exec cp --parents {} "backup_dir"
 \;
sarvesh@DESKTOP-H27RAHS:~$ cd backup_dir
sarvesh@DESKTOP-H27RAHS:~/backup_dir$ ls
help
sarvesh@DESKTOP-H27RAHS:~/backup_dir$
```

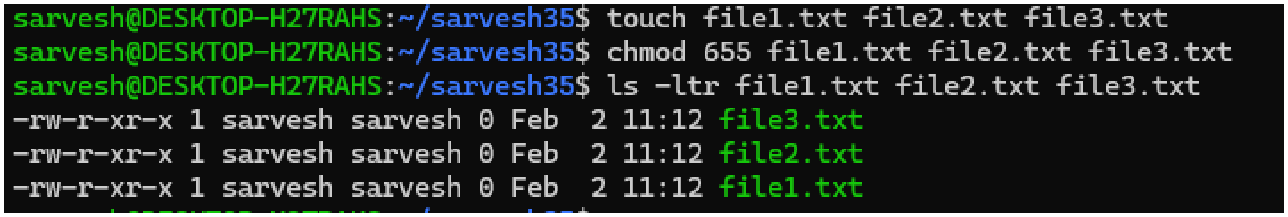
75. Write a command to display only the longest line in a file using wc and sort.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ awk '{ print length, $0 }' serialized.txt | sort -nr | head -n 1 | cut -d' ' -f2-
how are you
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

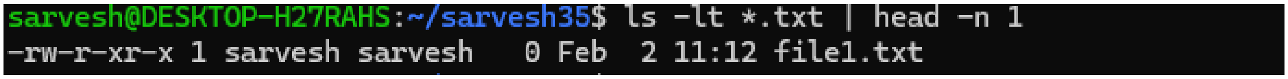
10. Comprehensive Practical Tasks

76. Create multiple files using touch and change their permissions using chmod.



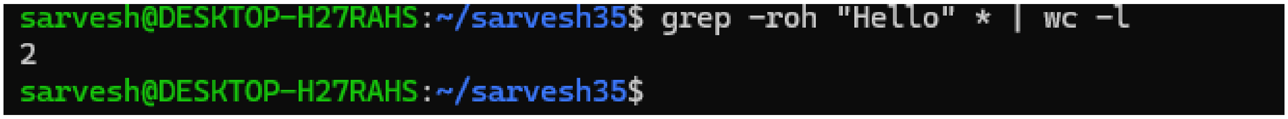
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ touch file1.txt file2.txt file3.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod 655 file1.txt file2.txt file3.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -ltr file1.txt file2.txt file3.txt
-rw-r-xr-x 1 sarvesh sarvesh 0 Feb  2 11:12 file3.txt
-rw-r-xr-x 1 sarvesh sarvesh 0 Feb  2 11:12 file2.txt
-rw-r-xr-x 1 sarvesh sarvesh 0 Feb  2 11:12 file1.txt
```

77. Write a command to find the most recent `.txt` file in a directory.



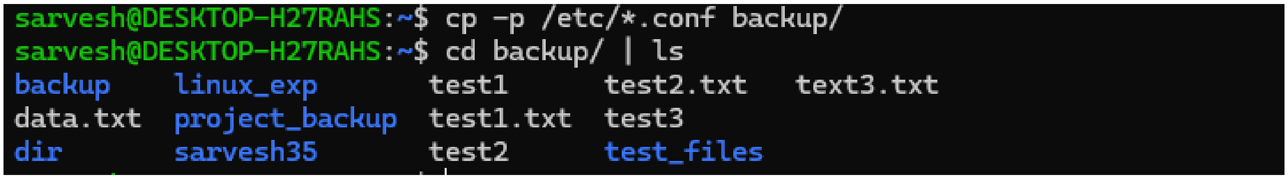
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -lt *.txt | head -n 1
-rw-r-xr-x 1 sarvesh sarvesh   0 Feb  2 11:12 file1.txt
```

78. Create a pipeline to search for a term in multiple files and count the occurrences.



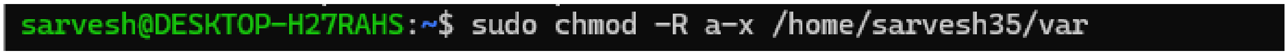
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ grep -roh "Hello" * | wc -l
2
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

79. Copy all `.conf` files from `/etc` to a `backup/` directory while preserving permissions.



```
sarvesh@DESKTOP-H27RAHS:~$ cp -p /etc/*.conf backup/
sarvesh@DESKTOP-H27RAHS:~$ cd backup/ | ls
backup    linux_exp       test1      test2.txt   text3.txt
data.txt  project_backup  test1.txt  test3
dir       sarvesh35       test2      test_files
```

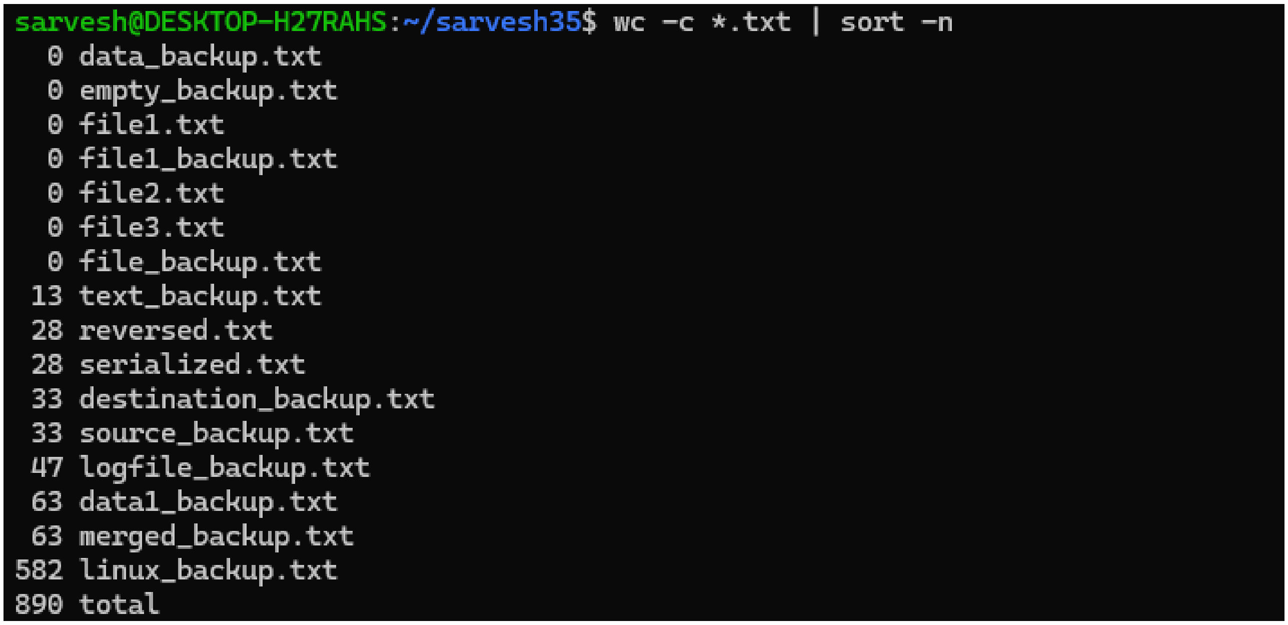
80. Use `chmod` to remove all execute permissions from a directory and its subdirectories.



```
sarvesh@DESKTOP-H27RAHS:~$ sudo chmod -R a-x /home/sarvesh35/var
```

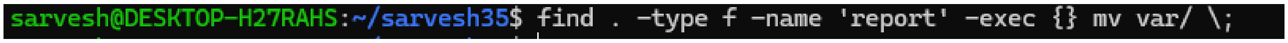
11. Additional Variations

81. Count the total characters in a directory's files and sort by file size.



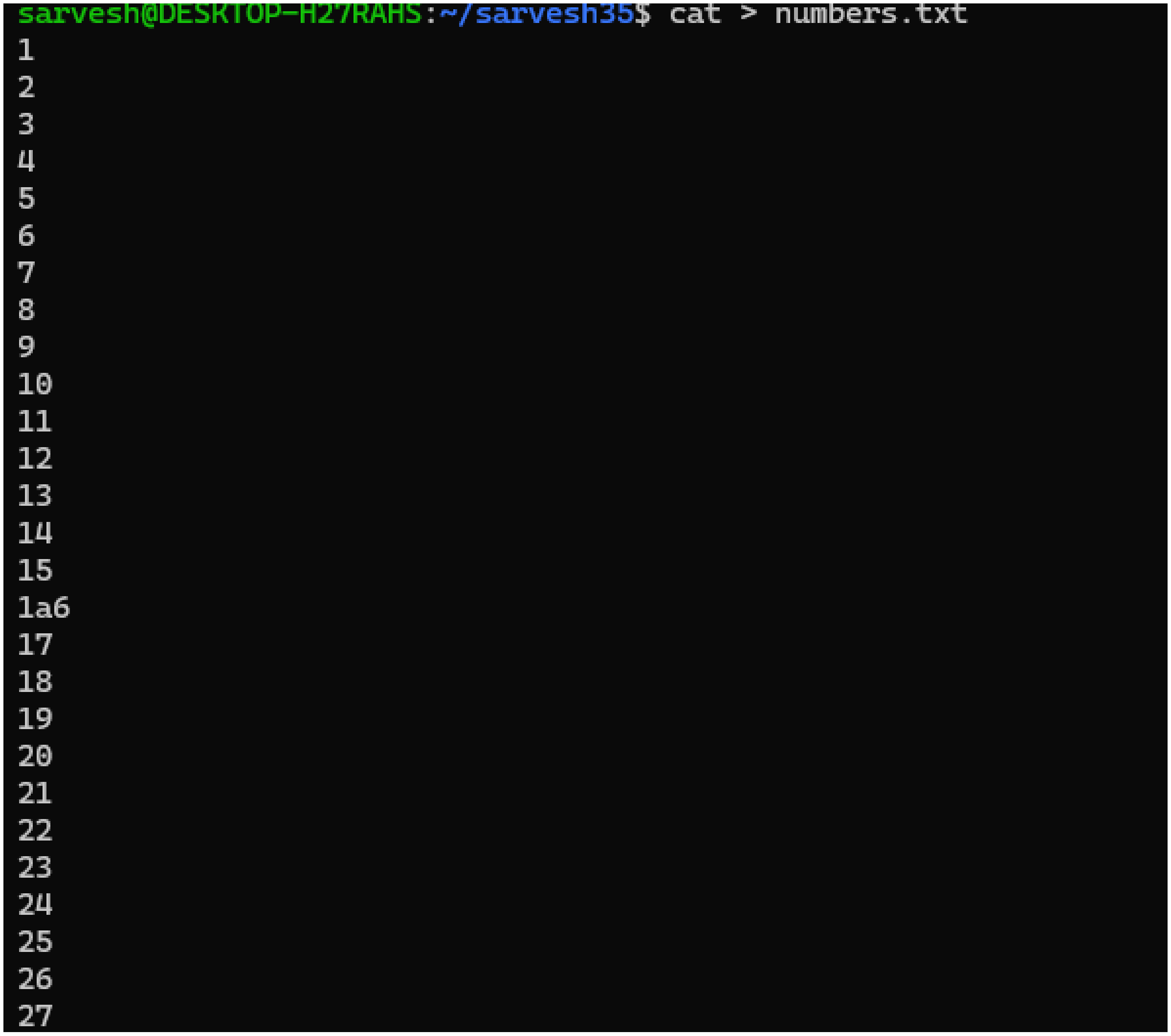
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ wc -c *.txt | sort -n
  0 data_backup.txt
  0 empty_backup.txt
  0 file1.txt
  0 file1_backup.txt
  0 file2.txt
  0 file3.txt
  0 file_backup.txt
 13 text_backup.txt
 28 reversed.txt
 28 serialized.txt
 33 destination_backup.txt
 33 source_backup.txt
 47 logfile_backup.txt
 63 data1_backup.txt
 63 merged_backup.txt
582 linux_backup.txt
890 total
```

82. Move all files containing the word "report" to a specific directory.

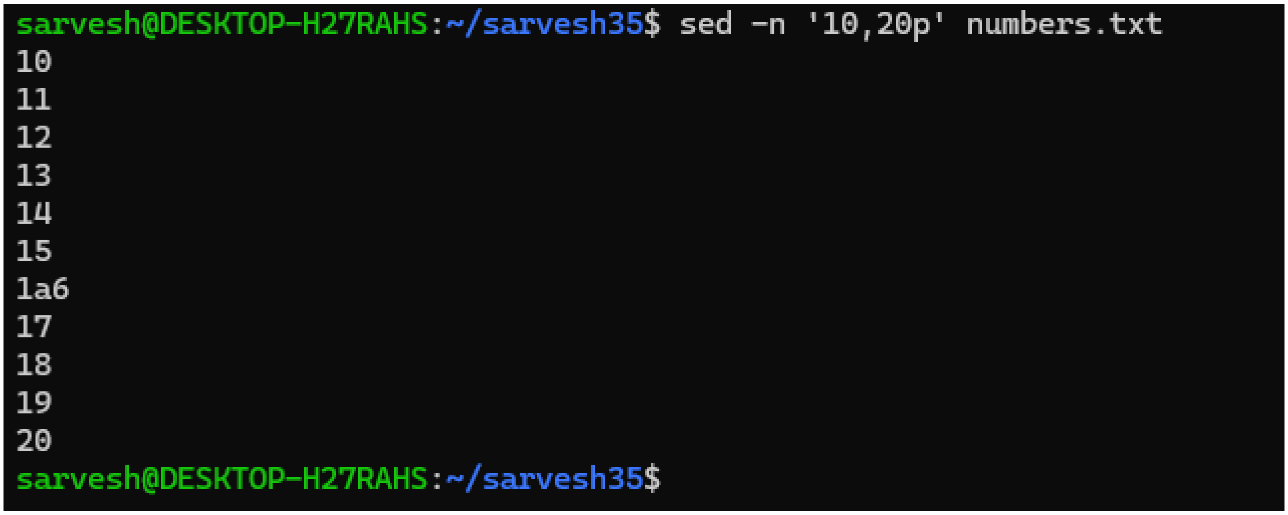


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -name 'report' -exec {} mv var/ \;
```

83. Create a command pipeline to display only the middle lines of a file.

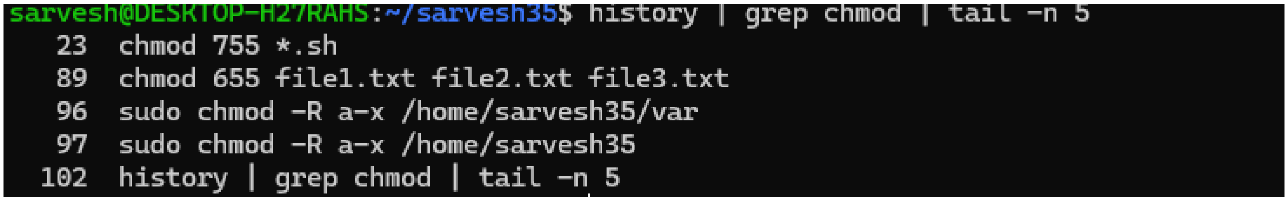


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat > numbers.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
1a6
17
18
19
20
21
22
23
24
25
26
27
```



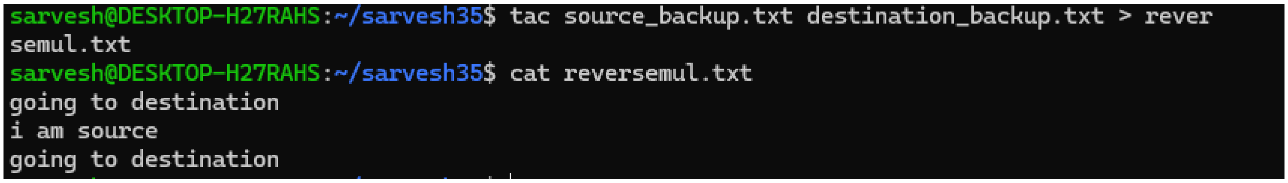
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ sed -n '10,20p' numbers.txt
10
11
12
13
14
15
1a6
17
18
19
20
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

84. Display the last 5 commands related to file permissions from the command history.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ history | grep chmod | tail -n 5
   23  chmod 755 *.sh
   89  chmod 655 file1.txt file2.txt file3.txt
   96  sudo chmod -R a-x /home/sarvesh35/var
   97  sudo chmod -R a-x /home/sarvesh35
  102  history | grep chmod | tail -n 5
```

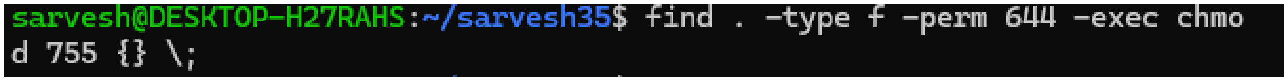
85. Reverse the contents of multiple files and save the output into corresponding new files.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ tac source_backup.txt destination_backup.txt > rever
semul.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat reversemul.txt
going to destination
i am source
going to destination
```

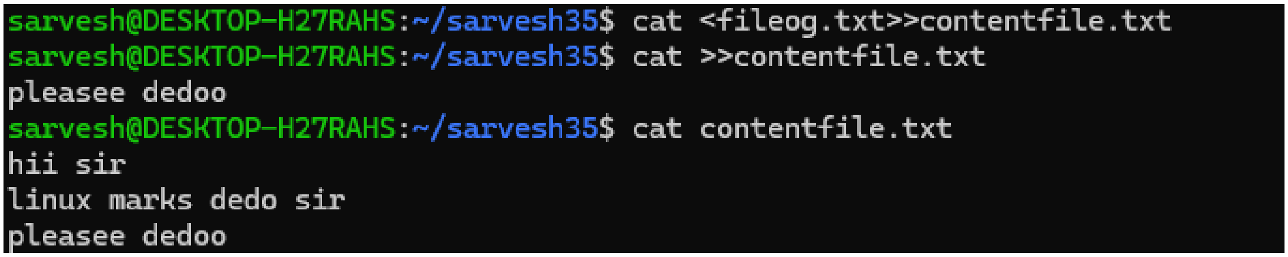
12. Extra Questions

86. Use `find` to locate files with specific permissions and modify them using `chmod`.



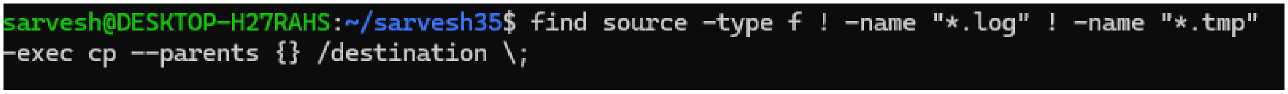
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -perm 644 -exec chmo
d 755 {} \;
```

87. Use `cat` to create a file, display its contents, and then append to it.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat <fileog.txt>>contentfile.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat >>contentfile.txt
pleasee dedoo
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat contentfile.txt
hii sir
linux marks dedo sir
pleasee dedoo
```

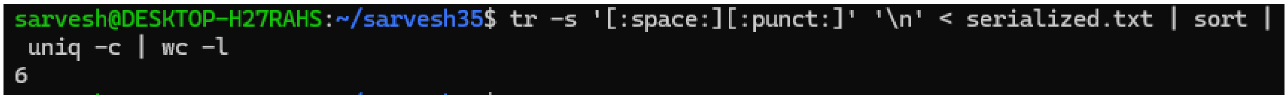
88. Copy files from one directory to another while excluding certain file types.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find source -type f ! -name "*.log" ! -name "*.tmp"
-exec cp --parents {} /destination \;
```

89. Write a script that uses `chmod`, `mv`, and `cp` to manage file backups.

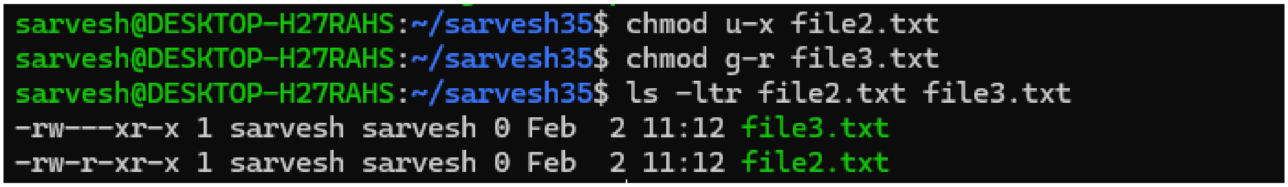
90. Create a pipeline to count the number of unique words in a file.



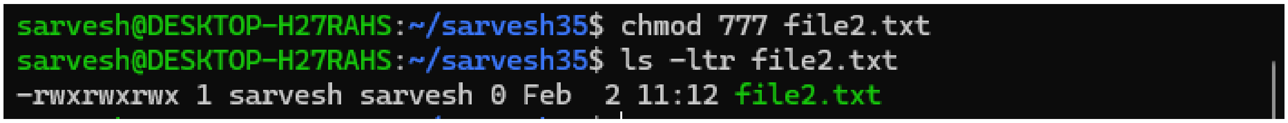
```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ tr -s '[:space:][:punct:]' '\n' < serialized.txt | sort |
 uniq -c | wc -l
6
```

13. Exploring the command

91. Explore the effect of `chmod` symbolic notation by modifying specific bits.

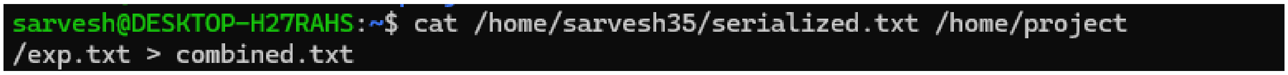


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod u-x file2.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod g-r file3.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -ltr file2.txt file3.txt
-rw---xr-x 1 sarvesh sarvesh 0 Feb  2 11:12 file3.txt
-rw-r-xr-x 1 sarvesh sarvesh 0 Feb  2 11:12 file2.txt
```

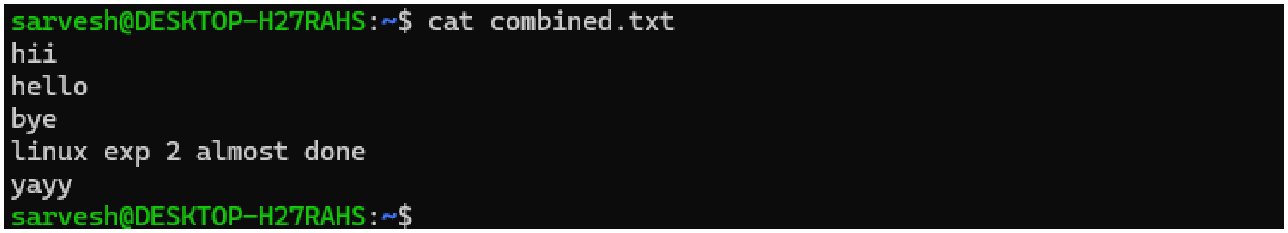


```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ chmod 777 file2.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ ls -ltr file2.txt
-rwxrwxrwx 1 sarvesh sarvesh 0 Feb  2 11:12 file2.txt
```

92. Use `cat` to combine contents from files in different directories.

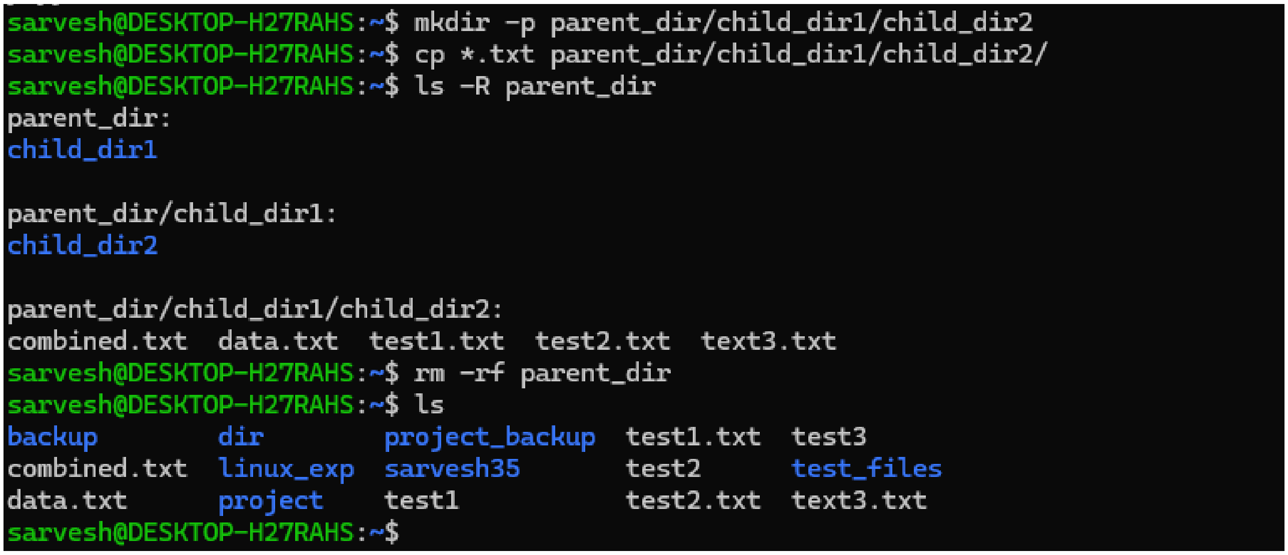


```
sarvesh@DESKTOP-H27RAHS:~$ cat /home/sarvesh35/serialized.txt /home/project
/exp.txt > combined.txt
```



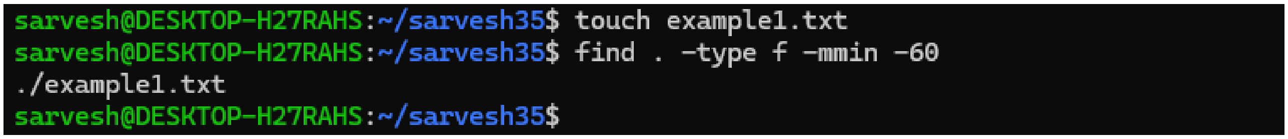
```
sarvesh@DESKTOP-H27RAHS:~$ cat combined.txt
hii
hello
bye
linux exp 2 almost done
yayy
sarvesh@DESKTOP-H27RAHS:~$
```

93. Create a nested directory structure, copy files into it, and then delete it.



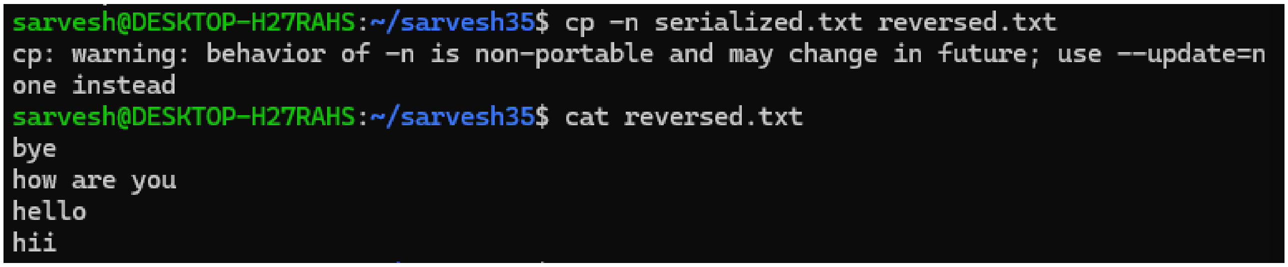
```
sarvesh@DESKTOP-H27RAHS:~$ mkdir -p parent_dir/child_dir1/child_dir2
sarvesh@DESKTOP-H27RAHS:~$ cp *.txt parent_dir/child_dir1/child_dir2/
sarvesh@DESKTOP-H27RAHS:~$ ls -R parent_dir
parent_dir:
child_dir1

parent_dir/child_dir1:
child_dir2

parent_dir/child_dir1/child_dir2:
combined.txt  data.txt  test1.txt  test2.txt  text3.txt
sarvesh@DESKTOP-H27RAHS:~$ rm -rf parent_dir
sarvesh@DESKTOP-H27RAHS:~$ ls
backup        dir        project_backup  test1.txt  test3
combined.txt  linux_exp  sarvesh35       test2      test_files
data.txt      project    test1           test2.txt  text3.txt
sarvesh@DESKTOP-H27RAHS:~$
```

94. Find and display all files modified within the past hour using `find`.



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ touch example1.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ find . -type f -mmin -60
./example1.txt
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$
```

95. Use `cp` with the `-n` flag



```
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cp -n serialized.txt reversed.txt
cp: warning: behavior of -n is non-portable and may change in future; use --update=n
one instead
sarvesh@DESKTOP-H27RAHS:~/sarvesh35$ cat reversed.txt
bye
how are you
hello
hii
```