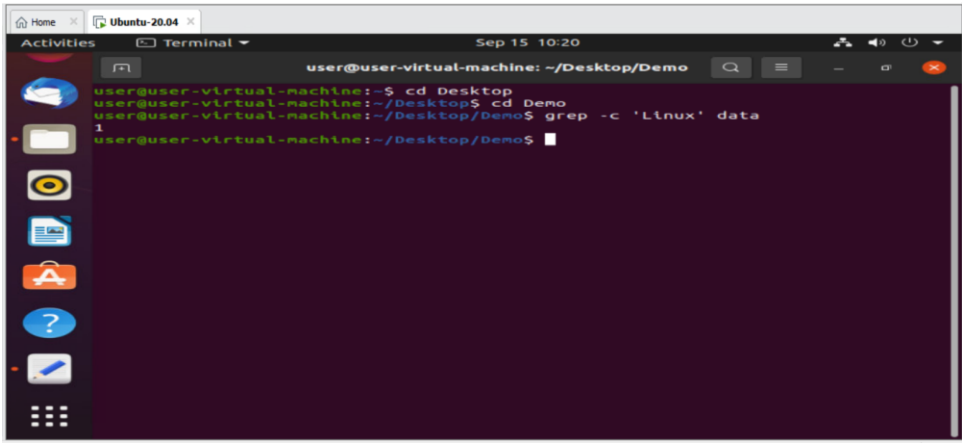
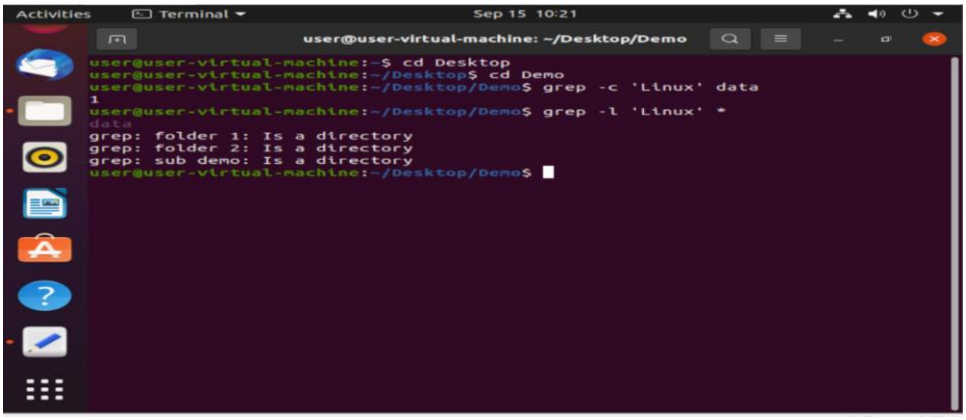
1.Grep – c

It displays in how many lines the particular word(here linux) has occurred.



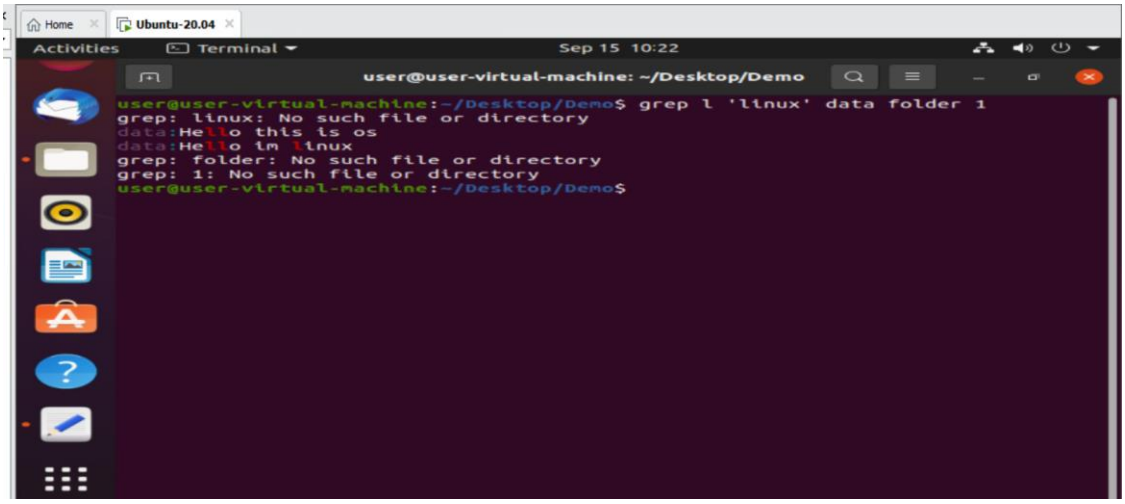
2. Grep – l*

It displays all the folders and directory in which the particular word is present

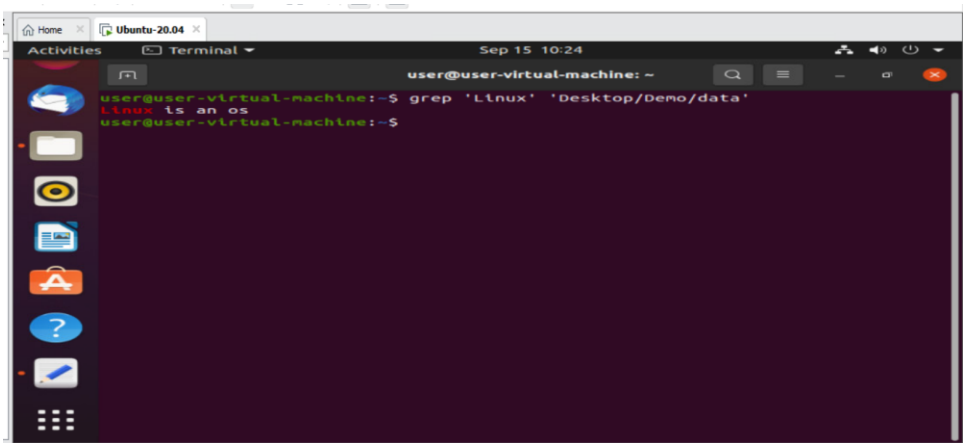


3. Grep – l

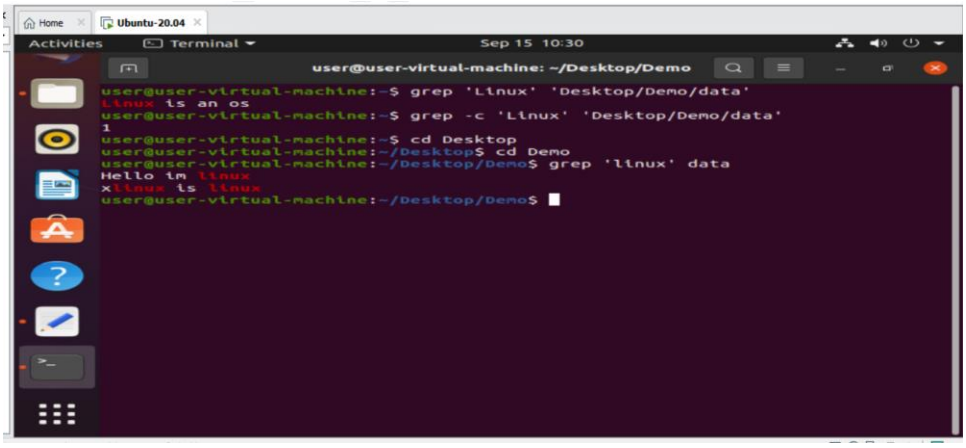
This command is used to display the particular pattern with ignoring case sensitive.



4. Grep ‘pattern name’ ‘directory’

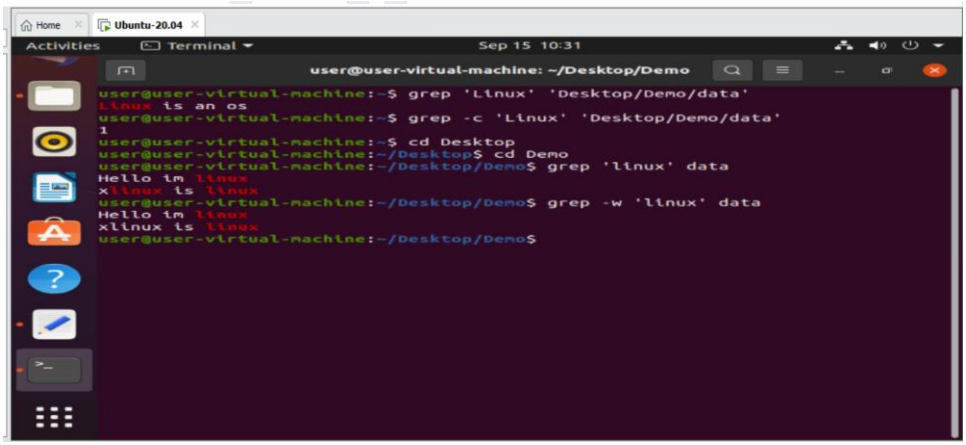


5. Grep ‘pattern name’ ‘filename’



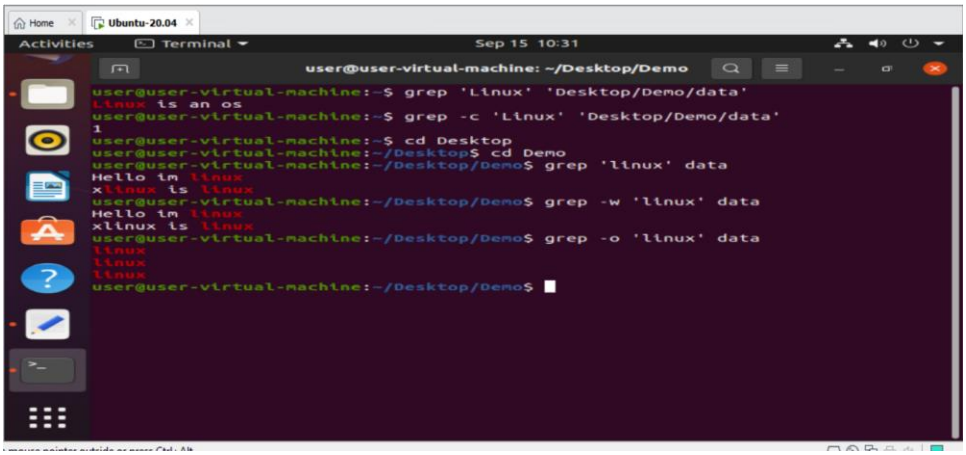
6.Grep –w

This command is used to print the pattern that matched whole word



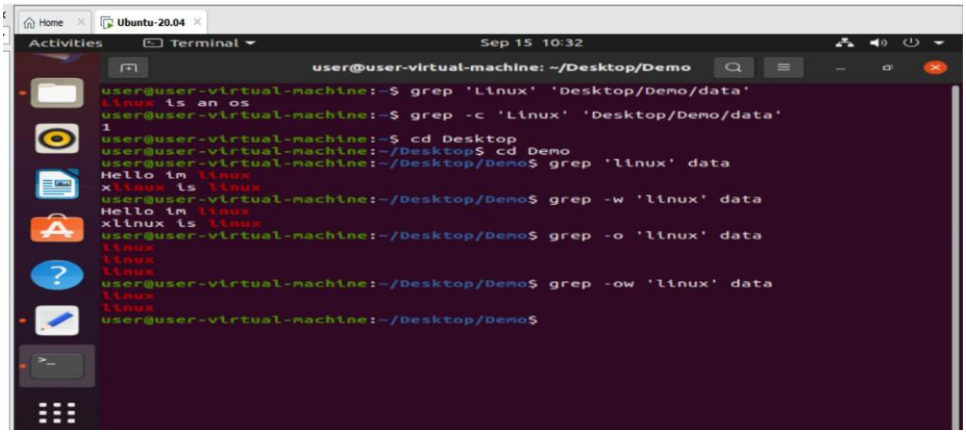
7. Grep –o

This command is used to print only the matched parts of a matching line, with each such part on a separate output line.

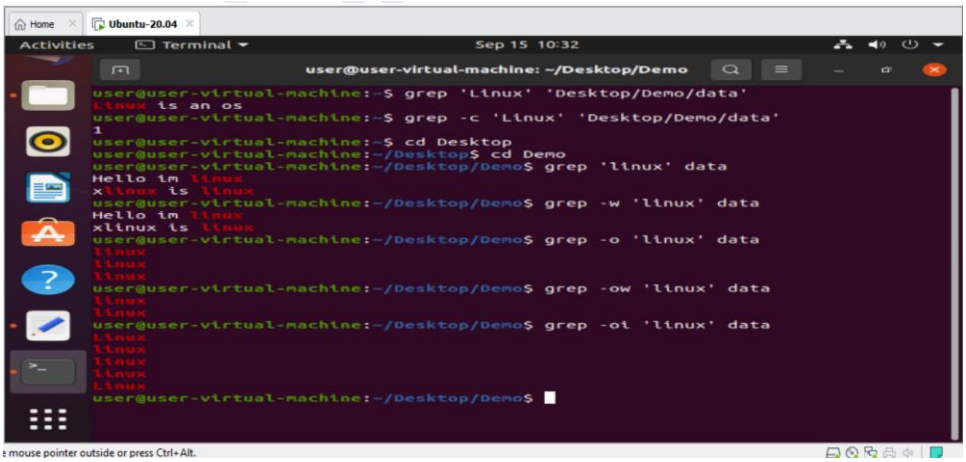


8.Grep –ow

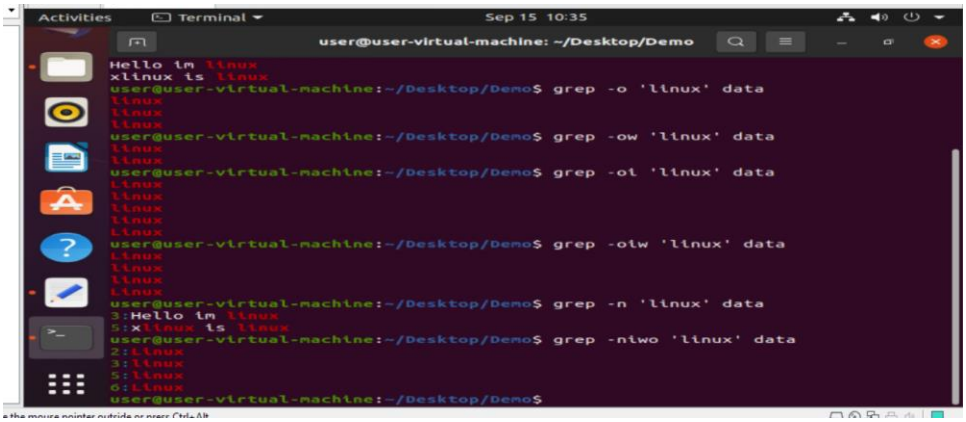
This command is used to print all the specific pattern without any matching or unmatching.



9.Grep –oi

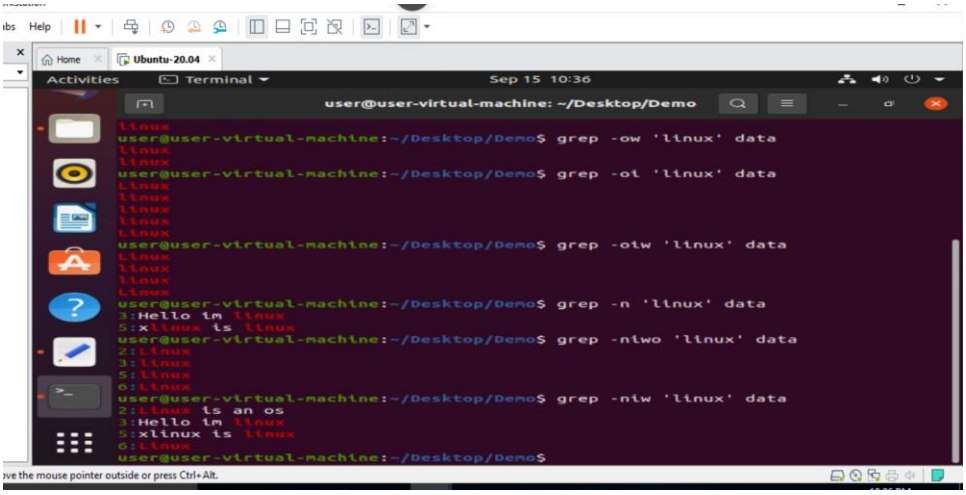


10.Grep –oiw

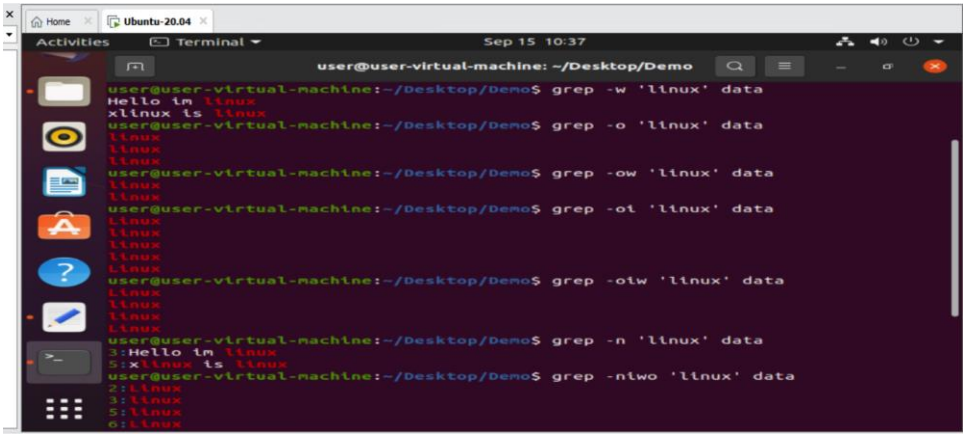


11.Grep –n

This Command is used to print the pattern the comes in the ending of the line with line numbers



12.1Grep –niwo



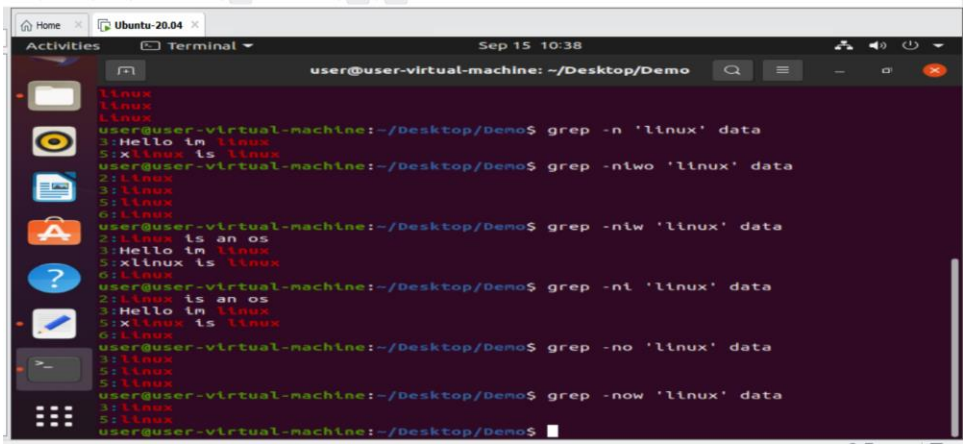
13.Grep –niw

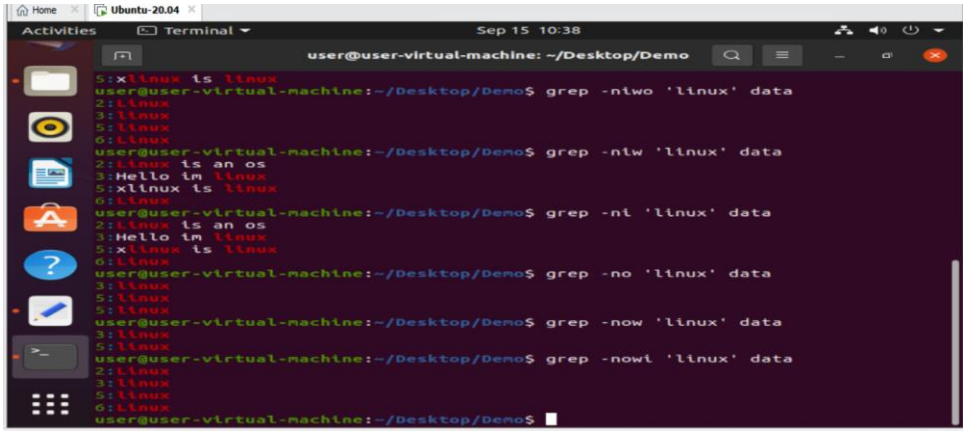
14.Grep –ni

15.Grep –no

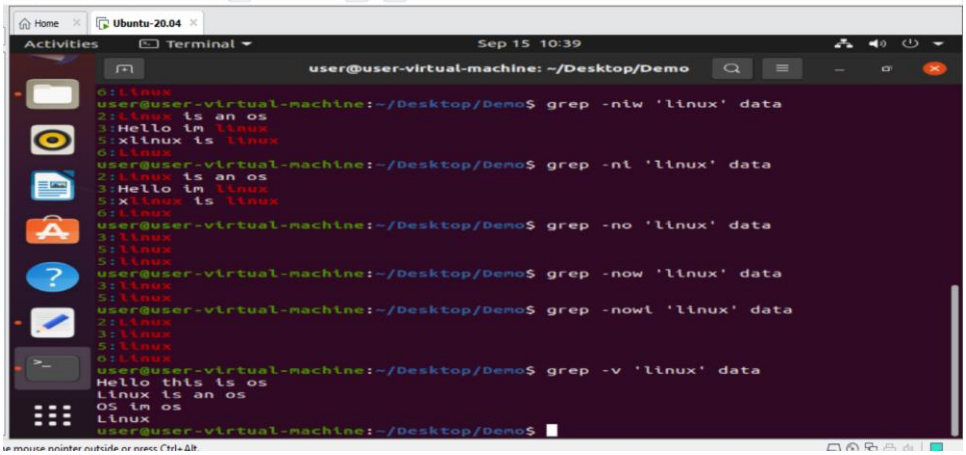
16.Grep –now

17.Grep –nowi

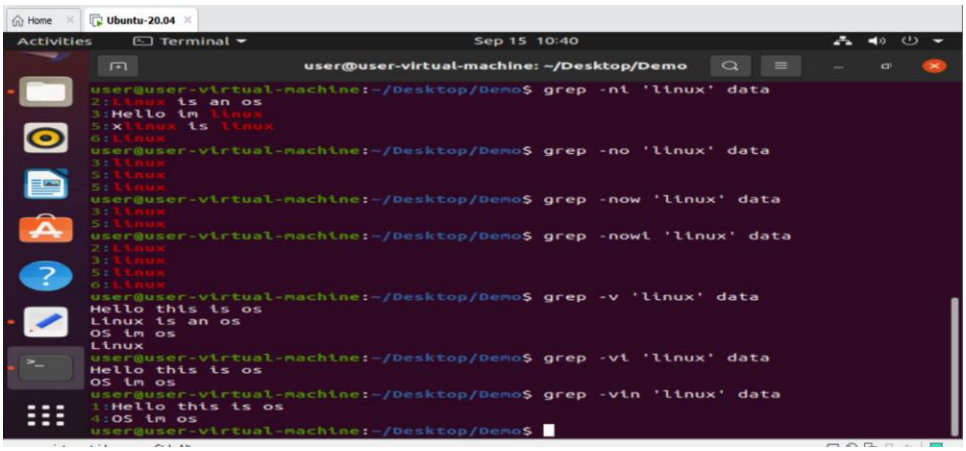




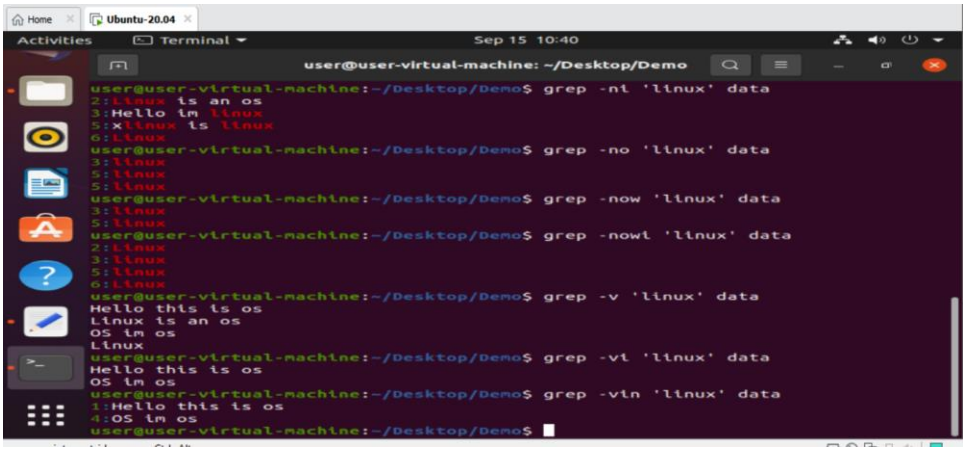
18.Grep –v



19.Grep –vi



20.Grep –vin



21.Grep –vino

22.Grep – ^

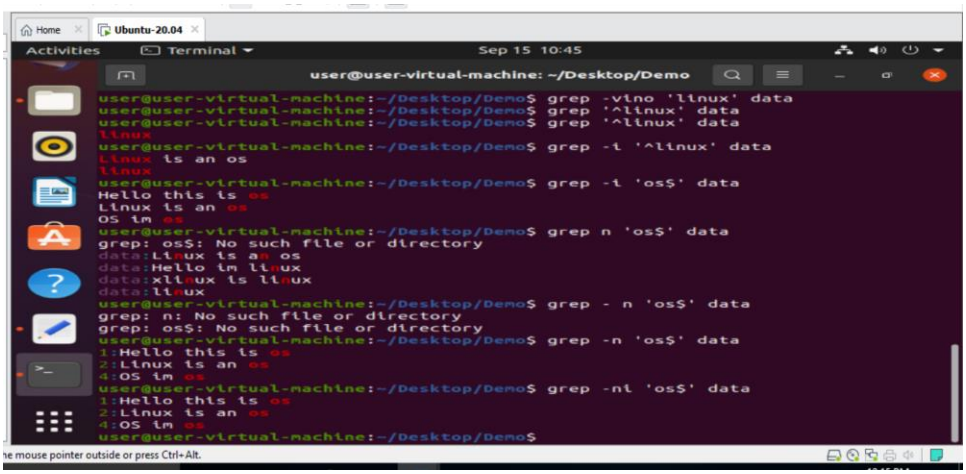
23.Grep – ^ linux

24.Grep – I ^

25.Grep –I $

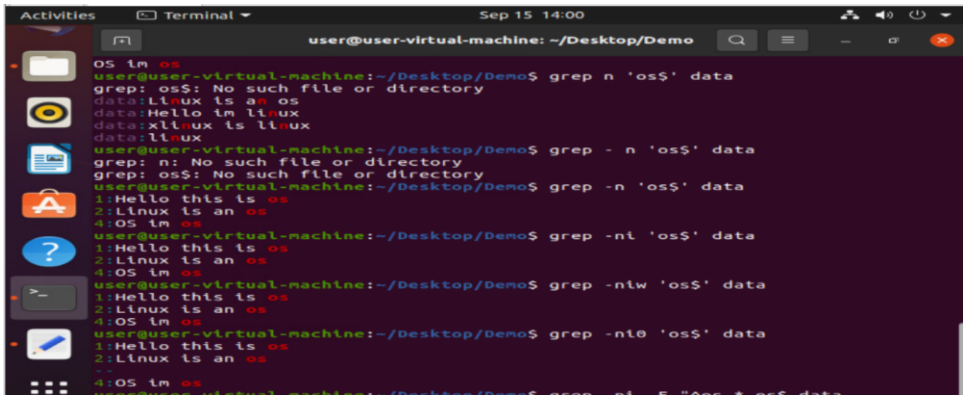
26.Grep – $

27.Grep –n



28.Grep niw

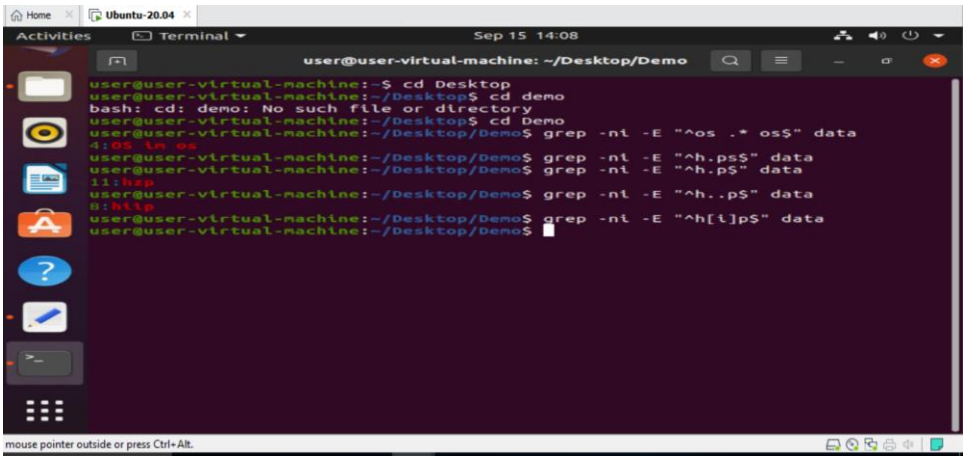
29. Grep nio



30.Grep ^os

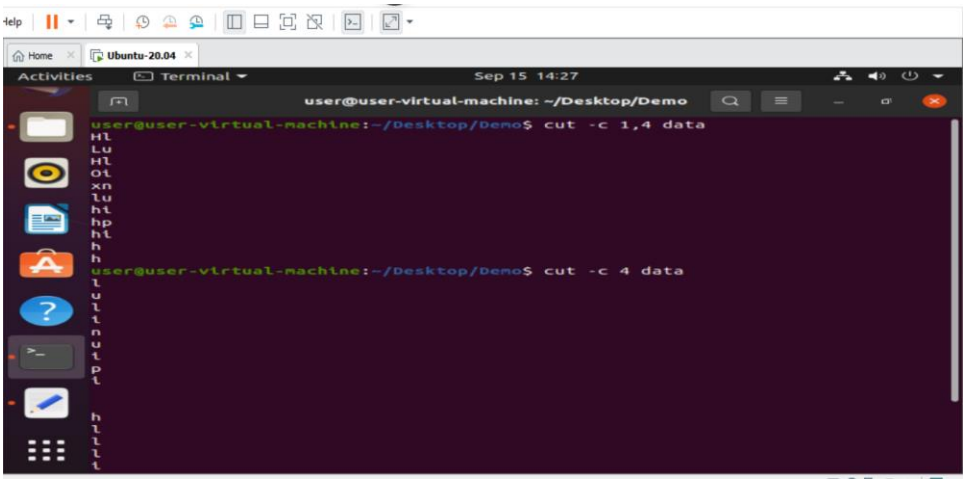
31.Grep h.p$

32. Grep h..p$

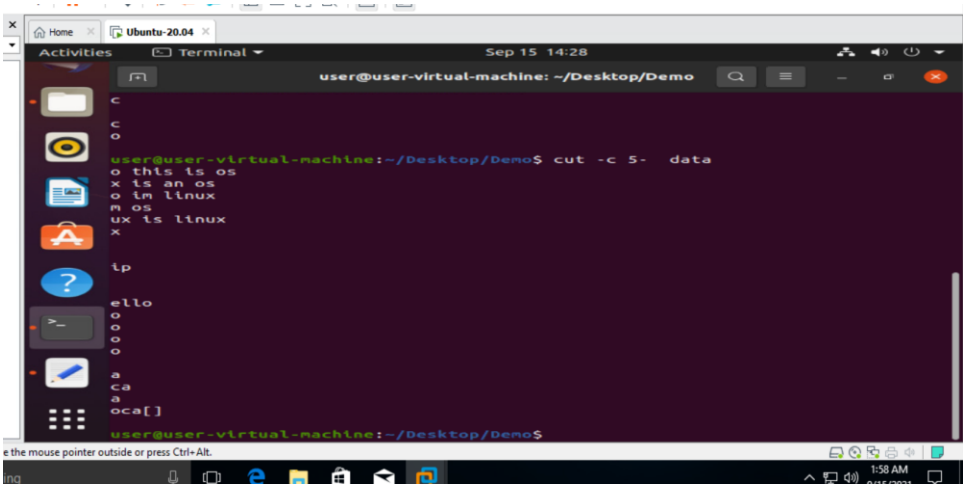


33.$cut -c 1,4 data2

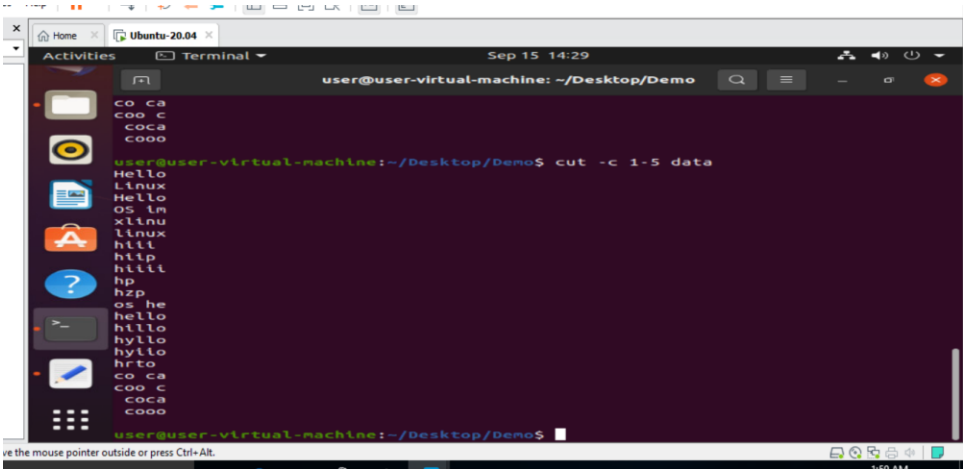
34. $cut -c 4 data2



35. $cut -c 5- data2

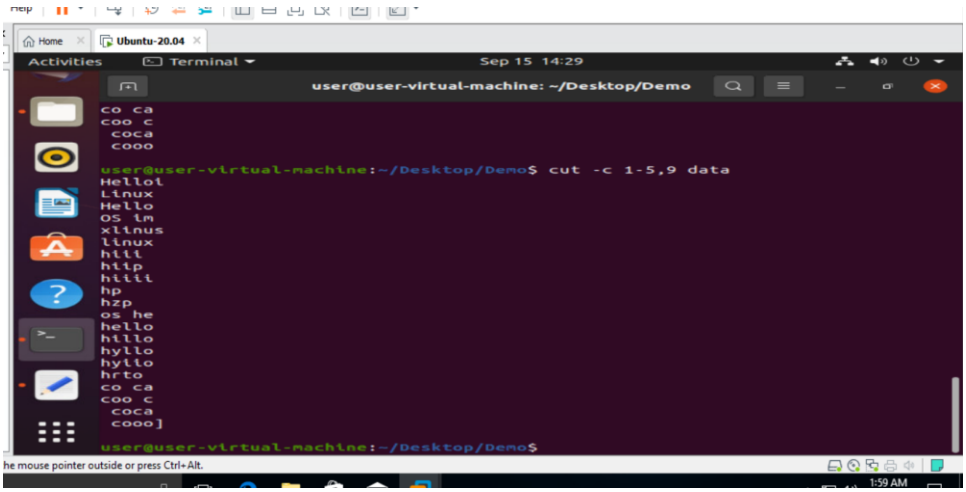


36. $cut -c 1-5 data2

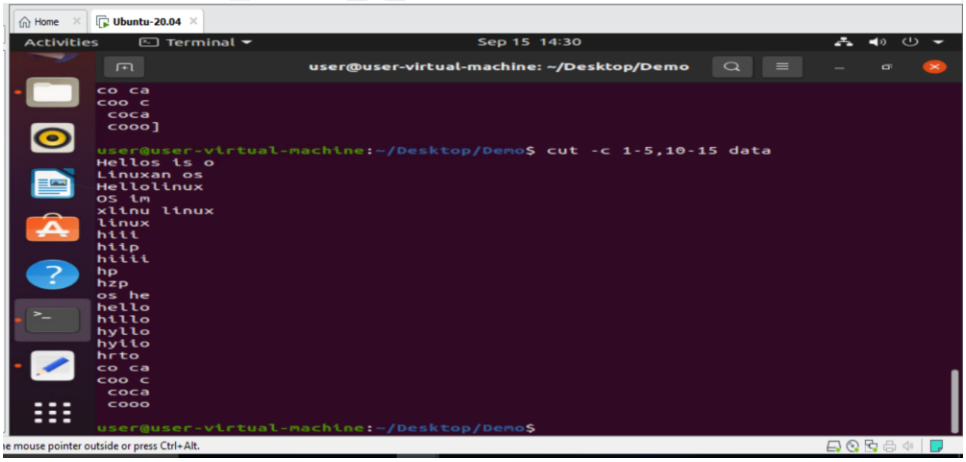


37.$cut -c 4-9 data2

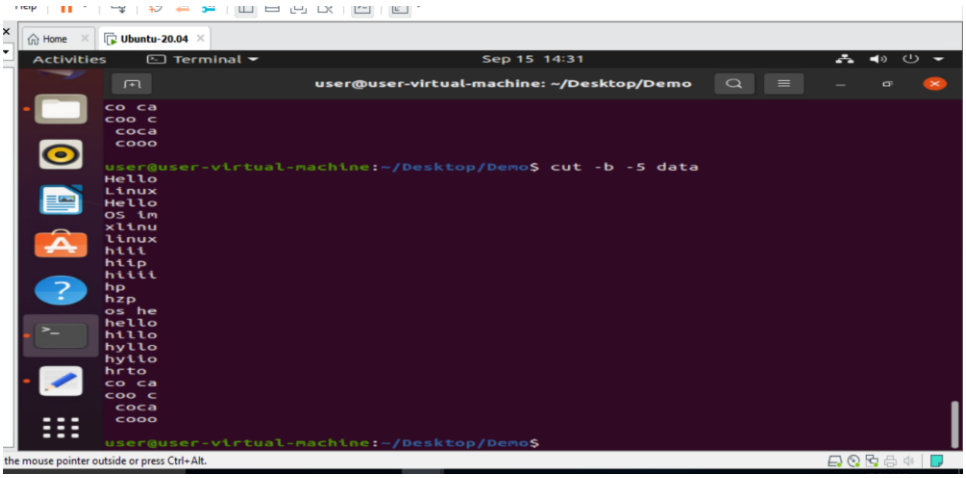
38. $cut -c 1,5-9 data2



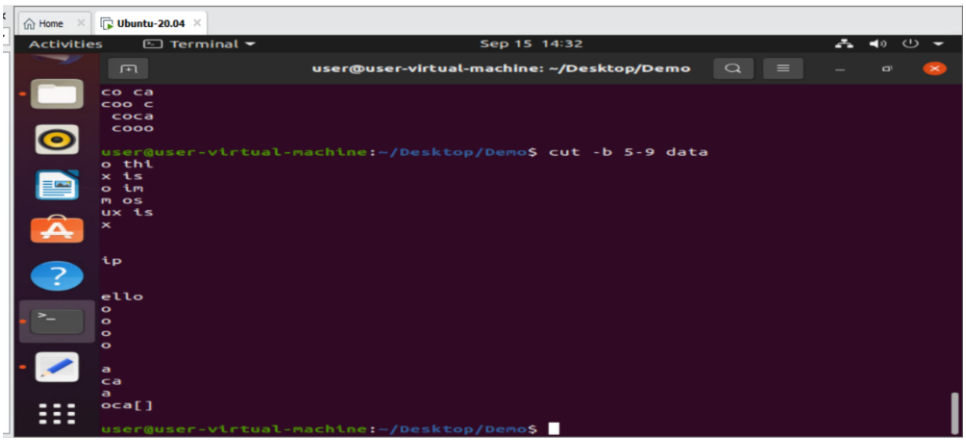
39.$cut -c 1-5 , 10-15 data



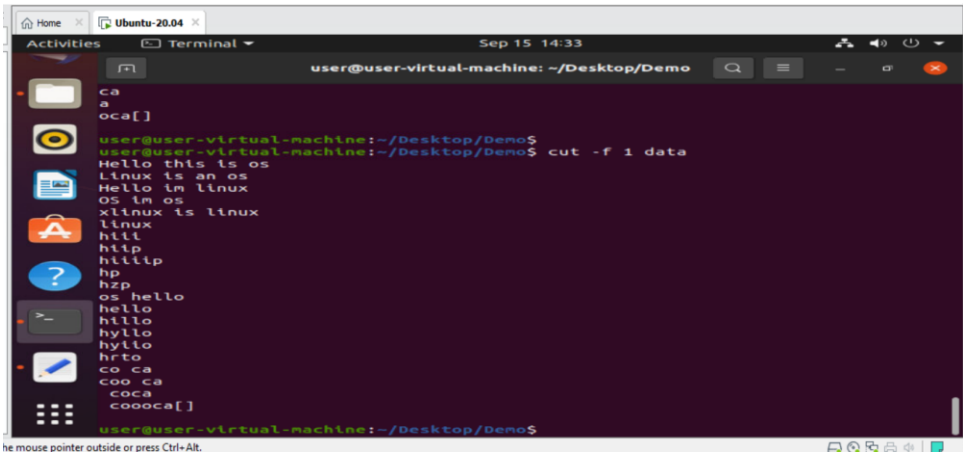
40.$cut -b -5 data



41.Cut –b 5-9 data



42.Cut –f 1 data

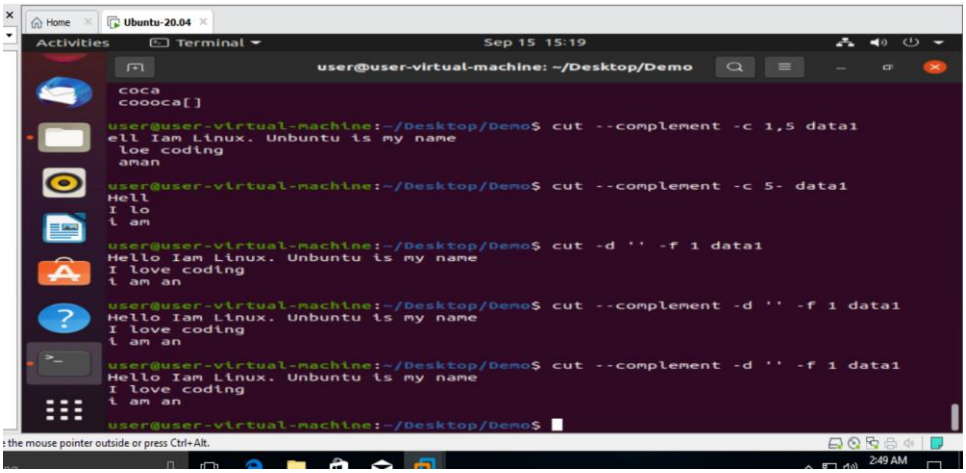


43. $cut –complement –c 1,5 data1 :- This command is used to display the selected column of the file.

44. $cut –complement –c 5- data1 - This command is used to display the the data from 1 column to 4th column

45. Cut –d ‘ ‘ -f 1 data1

46. $cut –complement –d ‘ ‘ -f 1 data1



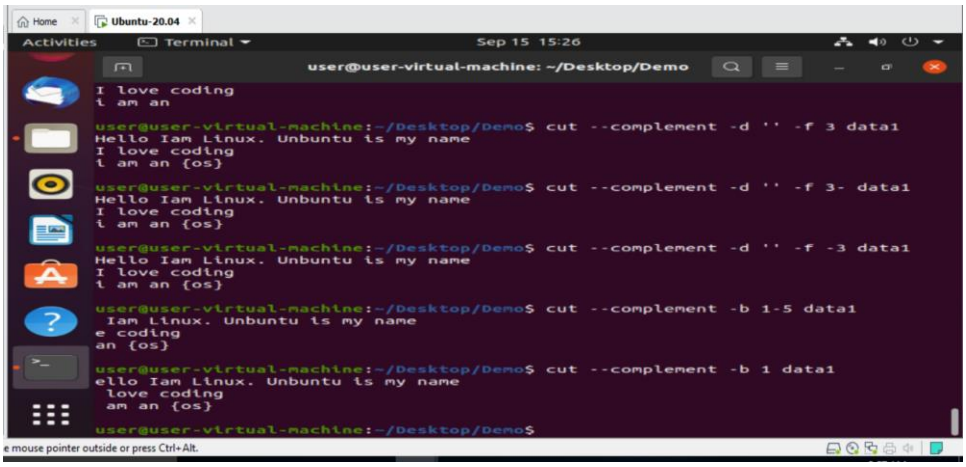
47. $cut –complement –d ‘ ‘ -f 3 data1

48. $cut –complement –d ‘ ‘ -f 3- data1

49. $cut –complement –d ‘ ‘ -f -3 data1

50. $cut –complement –b 1 -5 data1

51. $cut –complement –b 1 data1

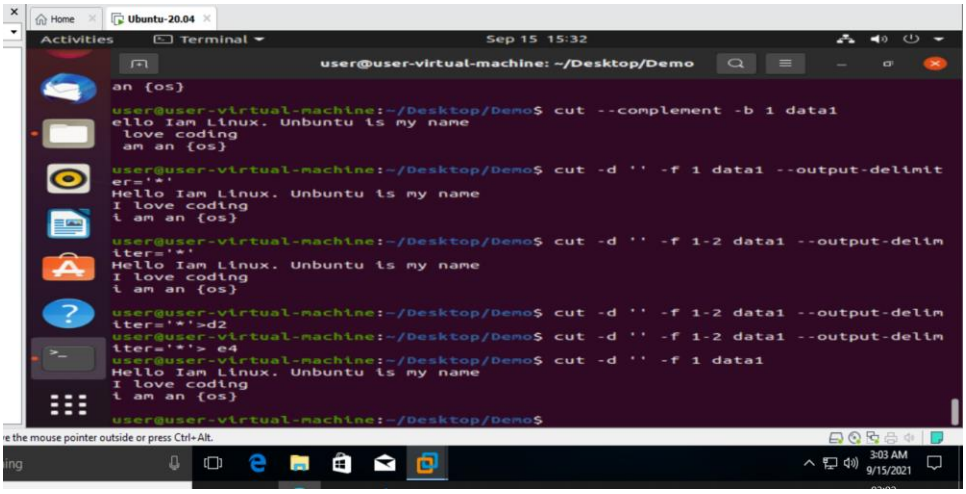


52.$ cut –d ‘ ‘ -f 1 data1 –output-delimiter=”*”

53. $cut –complement –d ‘ ‘ -f 1 data1

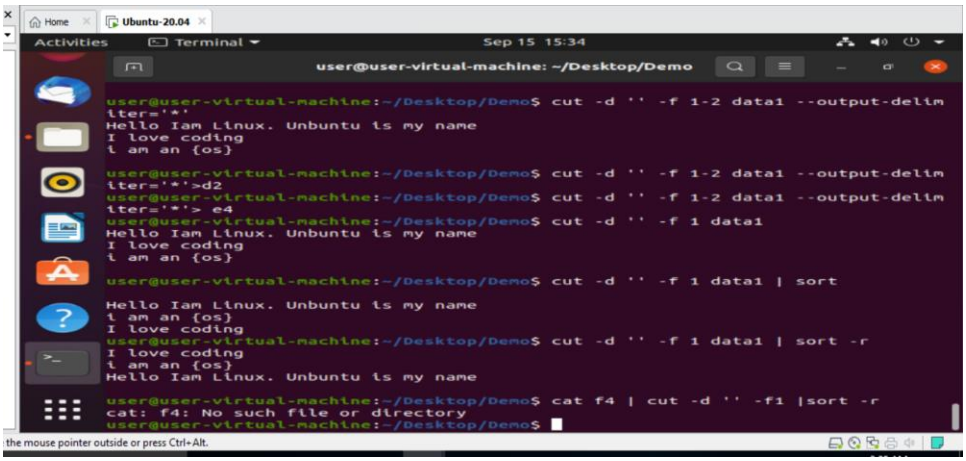
54. $cut –complement –d ‘ ‘ -f 1 data1 –output-delimiter=”*” >d2

55. $cut –complement –d ‘ ‘ -f 1 data1 –output-delimiter=”*” >e4

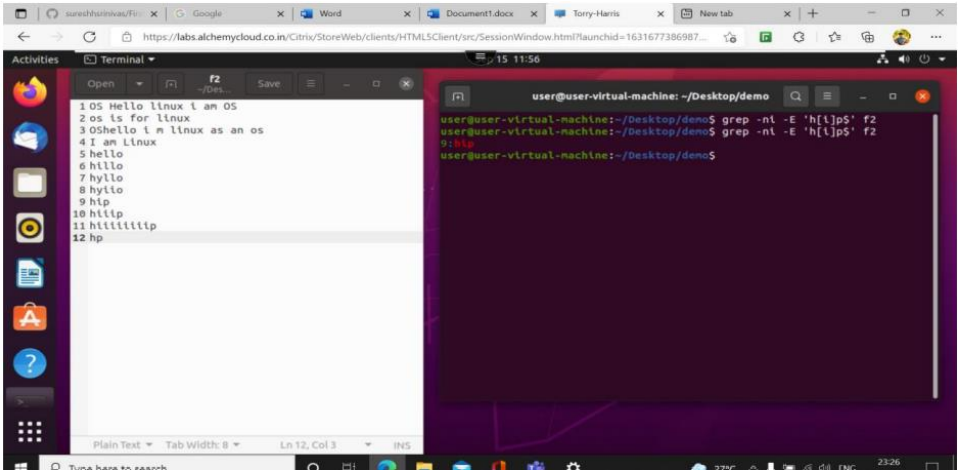


56. Cut –d ‘ ‘ -f 1 data1 | sort

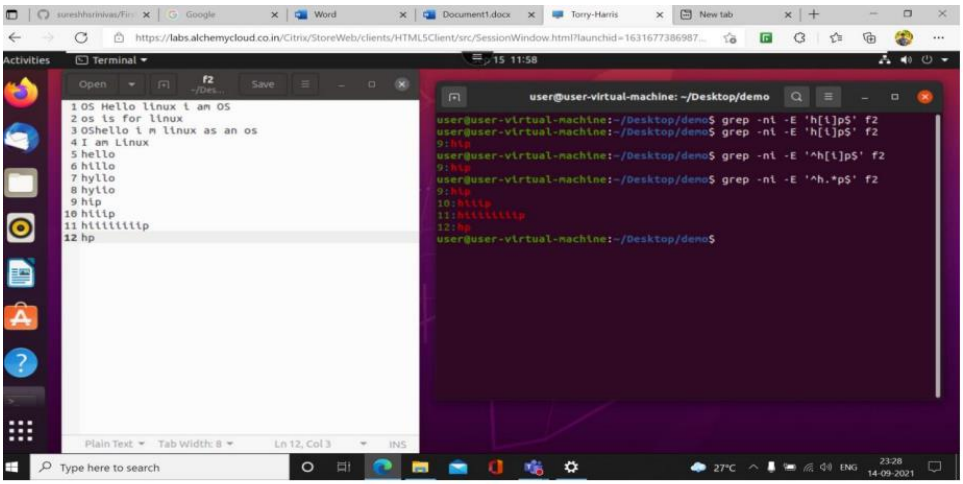
57. Cut –d ‘ ‘ -r 1 data1 | sort -r



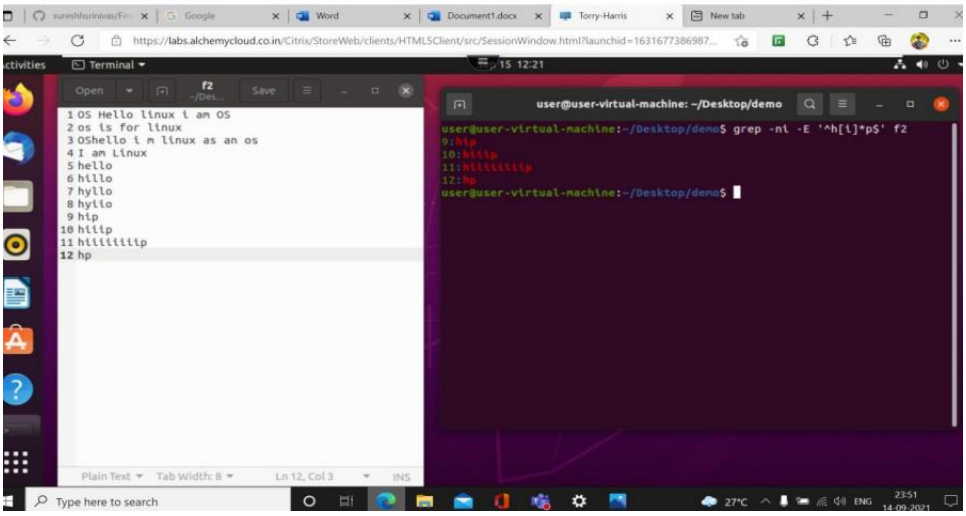
58. **$grep –ni –E ‘^h[I]p$’ f2**



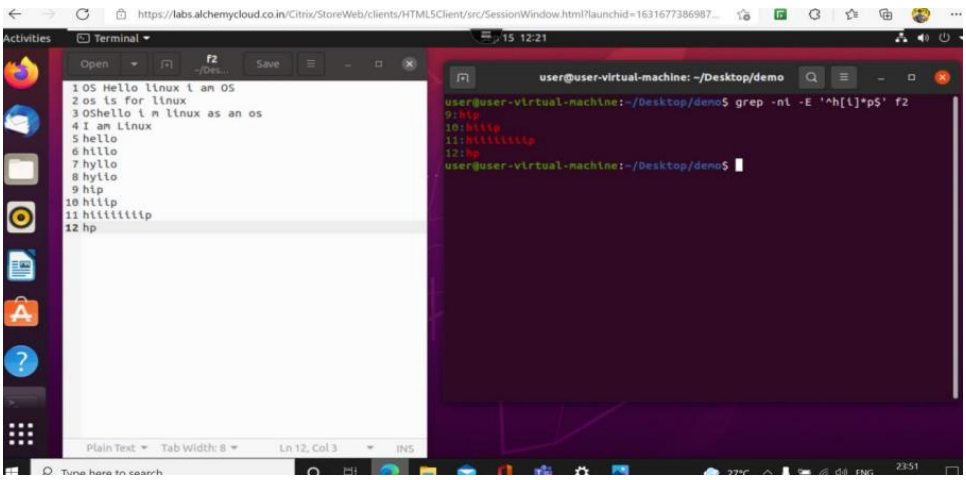
59. **$grep –ni –E ‘^h.*p$’ f2**



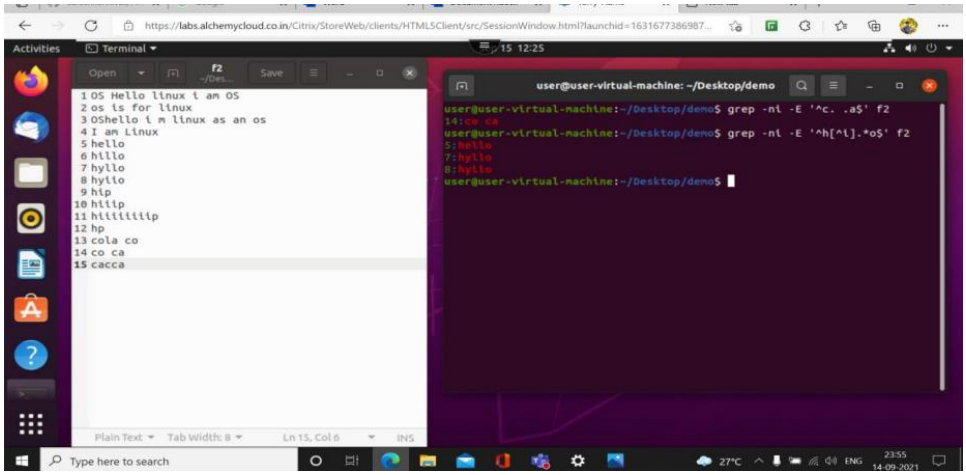
60.$grep –ni –E ‘^h[I]*p$’ f2



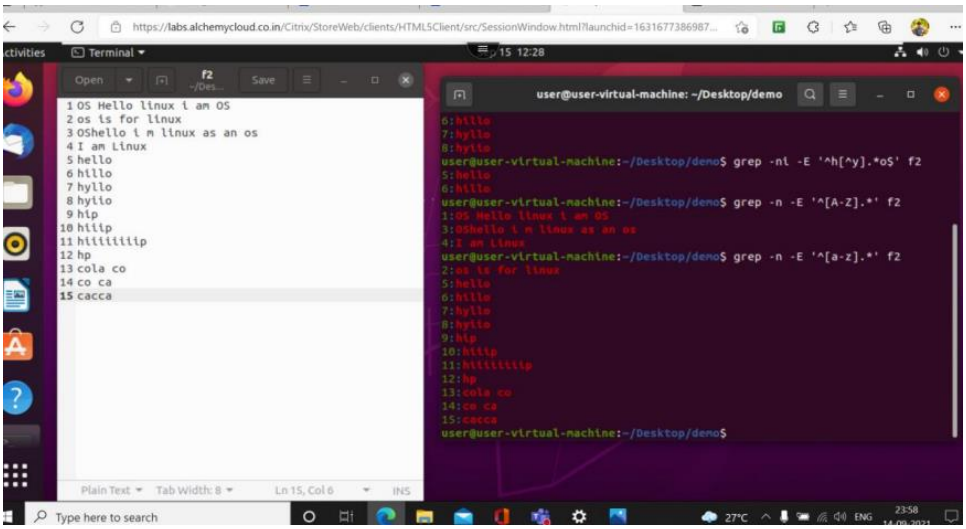
61.grep –ni –E ^h[I]+p$’ f2



62. $grep –ni -E ‘^h[^I].*o$’ f2

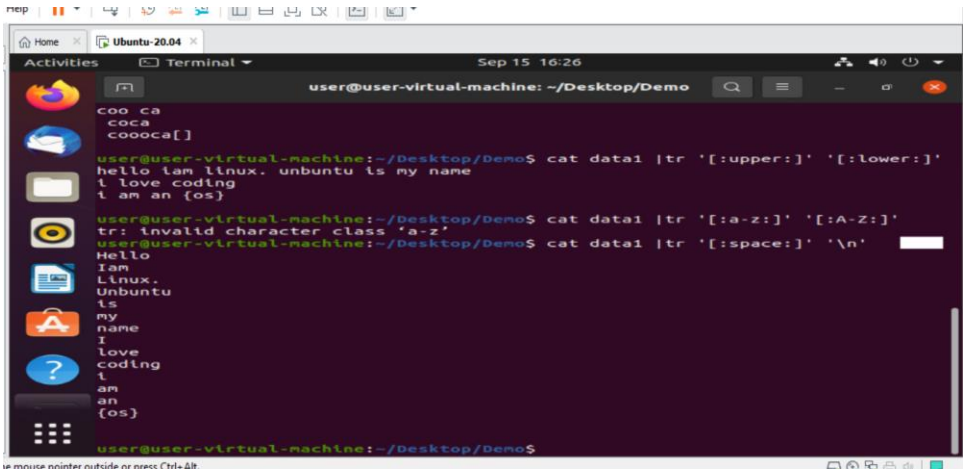


63. $grep –n -E ‘^[a-z].*’ f2



64. Cat data1 | tr '[:upper:]' '[:lower:]'

65. Cat data1 | tr '[space:] '\n'

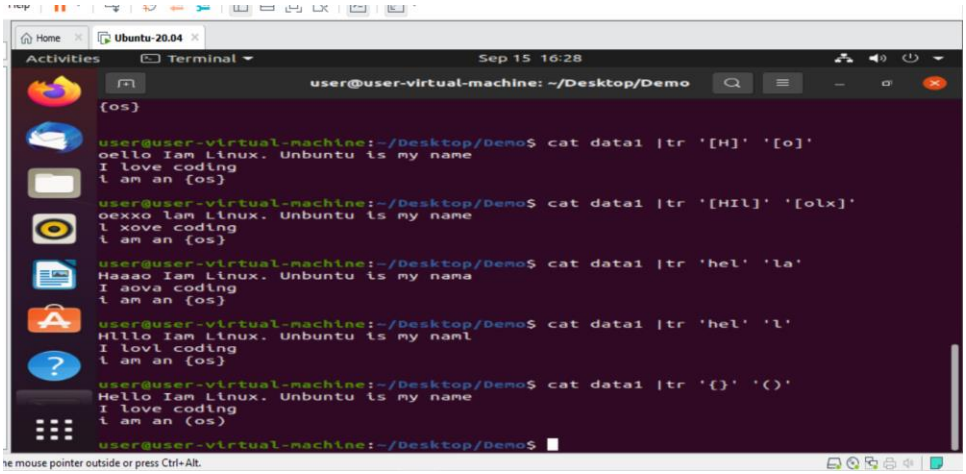


66. Cat data1 | tr '[H]' '[o]'

67. Cat data1 | tr '[Hil]' '[olx]'

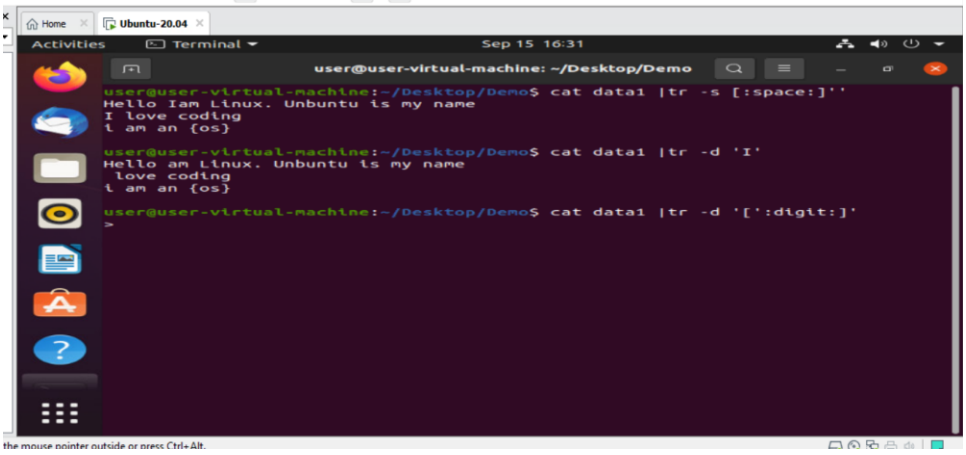
68. Cat data1 | tr 'hel' 'l'

69. Cat data1 | tr '{}' '()'



70. Cat data1 | tr –s [:space:]’ ’ cat data1 | tr –d ‘I’

71. Cat data1 | tr –s [:space:]’ ’ cat data1 | tr –d ‘[:digit:]’

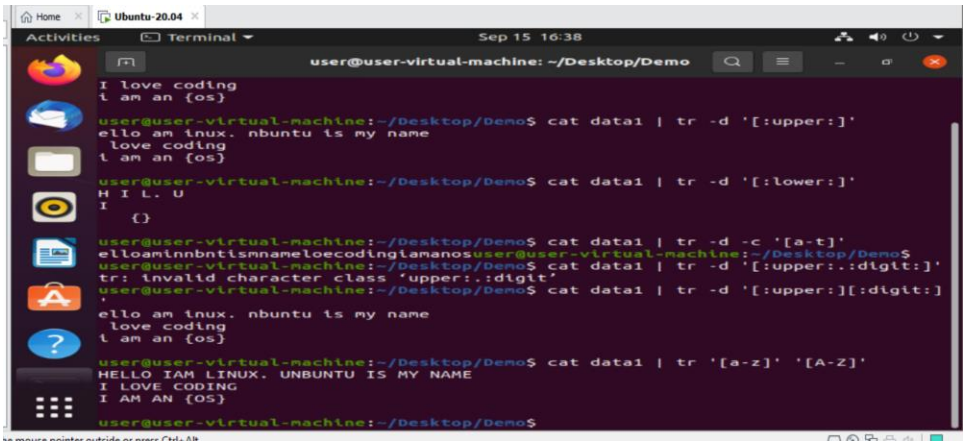


72. Cat data1 | tr –d ‘[:upper:]’

73. Cat data1 | tr –d ‘[:lower:]’

74. Cat data1 | tr –d ‘[:upper:][:digit:]’

75.cat data1 | tr –d -c ‘[a-t]’



Important commands

[ ]: Matches any one of a set characters
[ ] with hyphen: Matches any one of a range characters
^: The pattern following it must occur at the beginning of each line
^ with [ ] : The pattern must not contain any character in the set specified
$: The pattern preceding it must occur at the end of each line
. (dot): Matches any one character
\ (backslash): Ignores the special meaning of the character following it
*: zero or more occurrences of the previous character
(dot).*: Nothing or any numbers of characters.