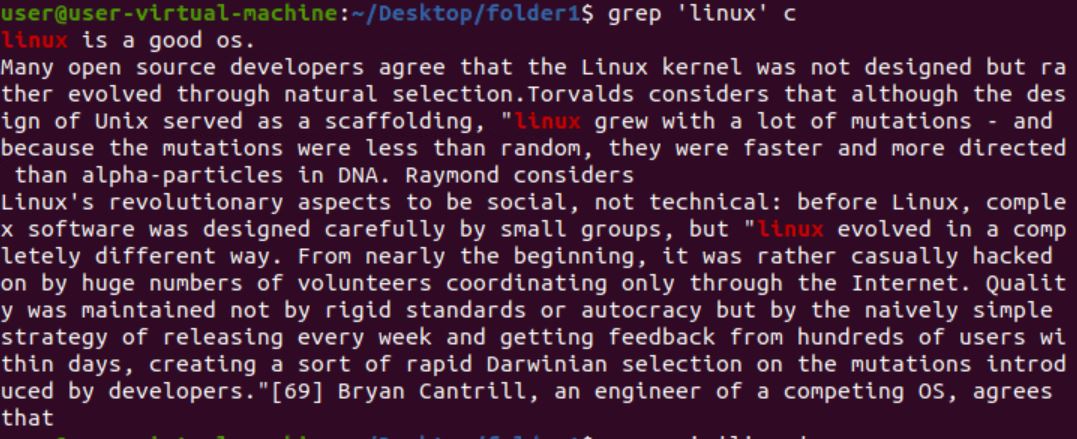
**Ubuntu commands :**

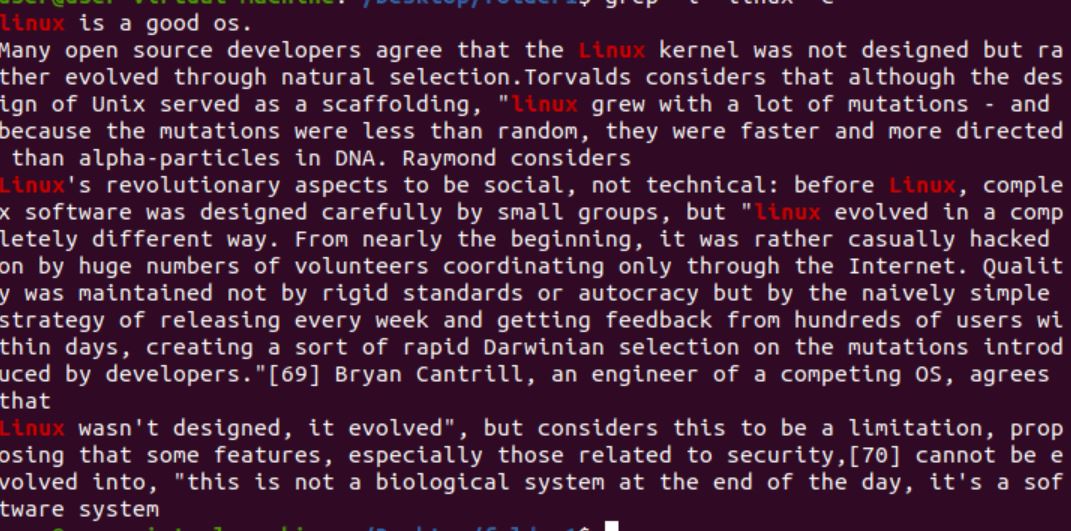
**1)Grep ‘word’ filename :**

-This command is used to fetch the given words along with the line it is present.



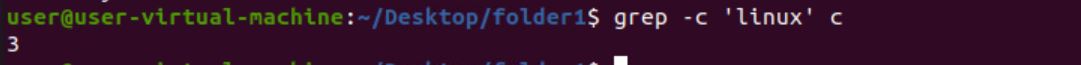
**2)Grep -i ‘word’ filename :**

-This command is used to fetch the given words along with the line it is present without any case sensitive.



**3)Grep –c ‘word’ filename :**

-This command is used to fetch the given word count of the line which start with the given word.



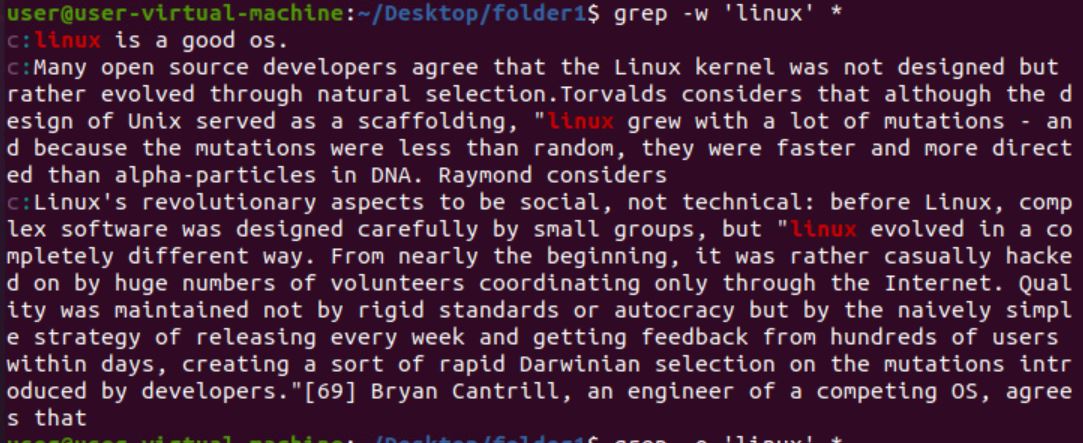
**4)Grep –l ‘word’ \* :**

-This command is used to fetch the file in which the word is present.



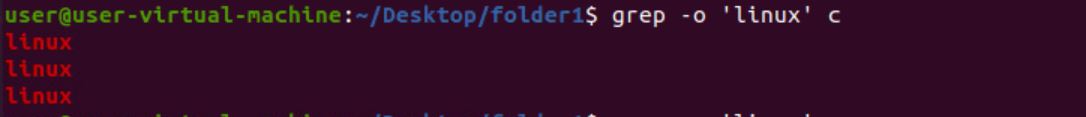
**5)Grep –w ‘word’ filename :**

-This command is used to fetch the full-fledged word.



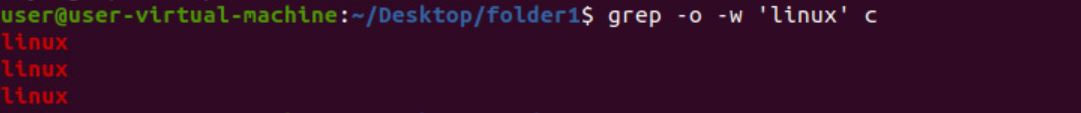
**6)Grep –o ‘word’ filename :**

-This command is used to fetch only the word not the full line.



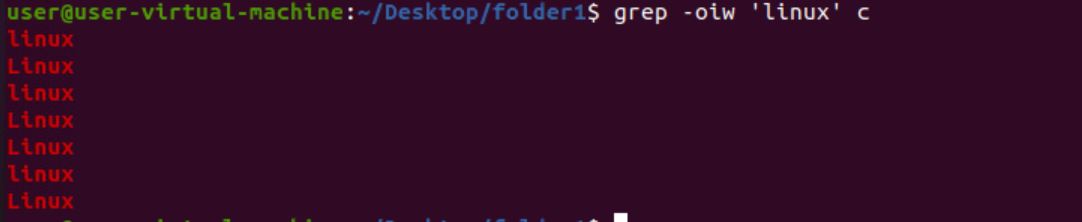
**7)Grep -o –w ‘word’ filename :**

-This command is used to fetch only the word not the full line.



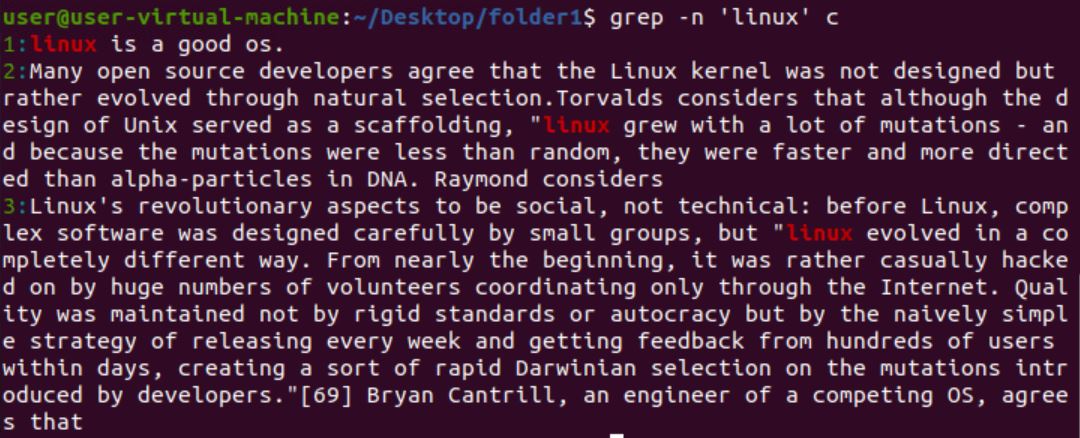
**8)Grep -oiw ‘word’ filename :**

-This command is used to fetch only the word not the full line irrespective of cases.



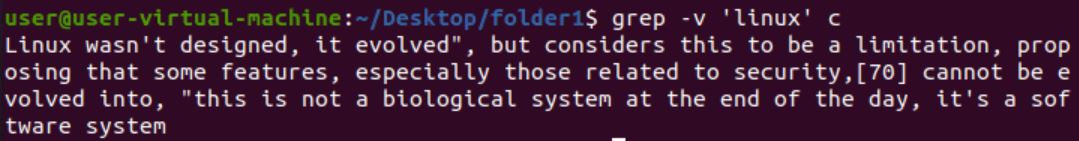
**9)Grep -n ‘word’ filename :**

-This command is used to fetch the word along with the line number.



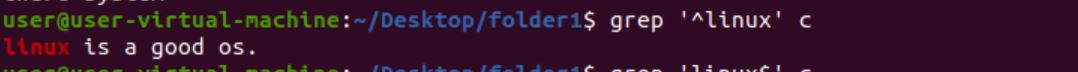
**9)Grep -v ‘word’ filename :**

-This command is used to fetch line in which the word is not present.



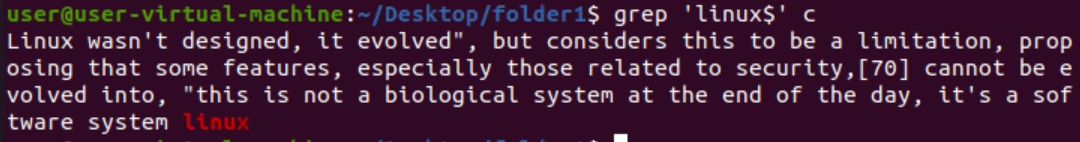
**11)grep ‘^linux’ filename :**

-This command is used to fetch the line using first word of the line.



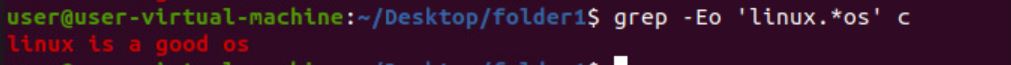
**12)grep ‘linux$’ filename :**

-This command is used to fetch the line using last word of the line.



**13)grep –E ‘linux.|\*os’ filename :**

-This command is used to fetch the line using first and last word of the line.

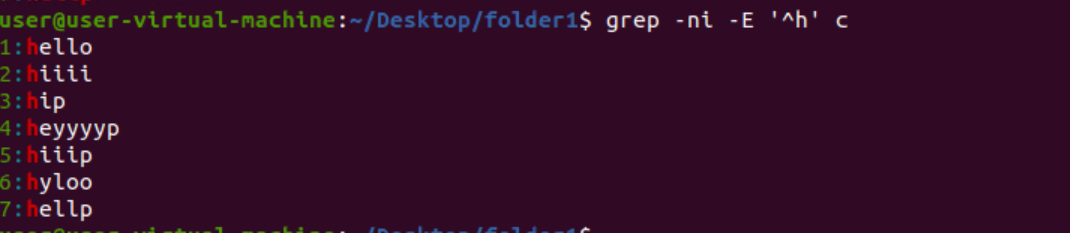


**REGULAR EXPRESSIONS :**

**regular expression syntax:**

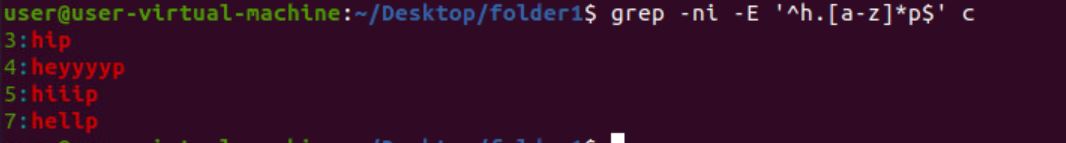
* **[ ]:**

**-Matches any one of a set characters**



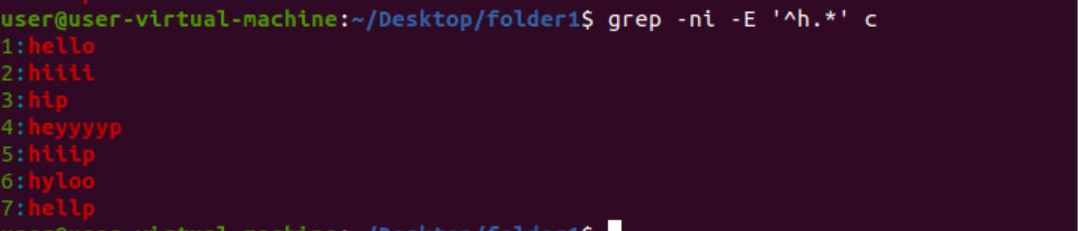
* **[ ] with hyphen:**

**-Matches any one of a range characters**



* **^:**

**-The pattern following it must occur at the beginning of each line**



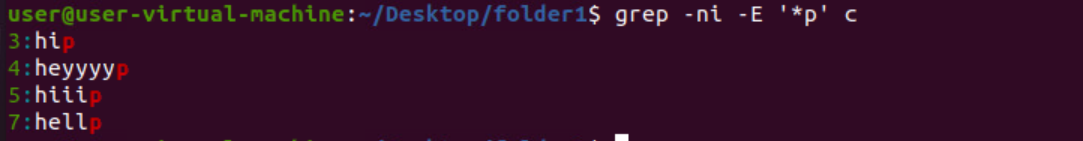
* **^ with [ ] :**

**-The pattern must not contain any character in the set specified**



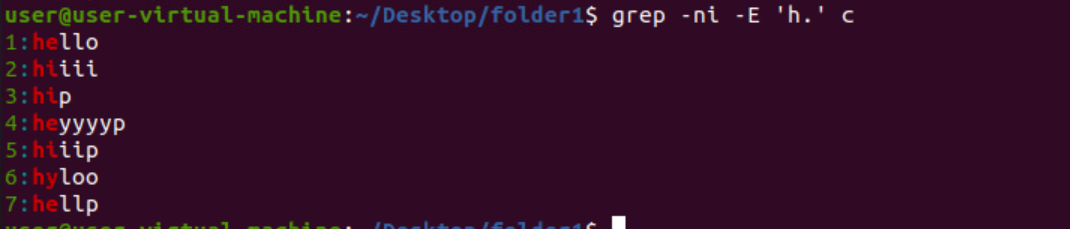
* **$:**

**-The pattern preceding it must occur at the end of each line**



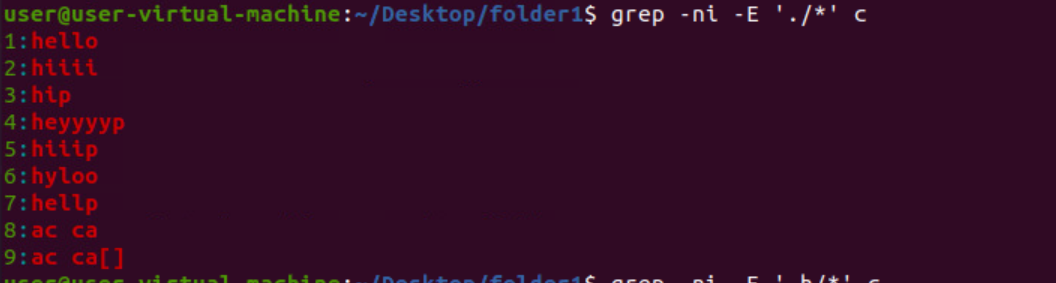
* **. (dot):**

**-Matches any one character**



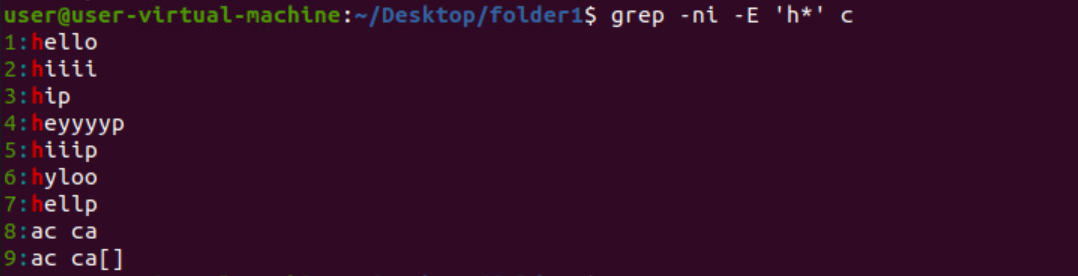
* **\ (backslash):**

**-Ignores the special meaning of the character following it.**



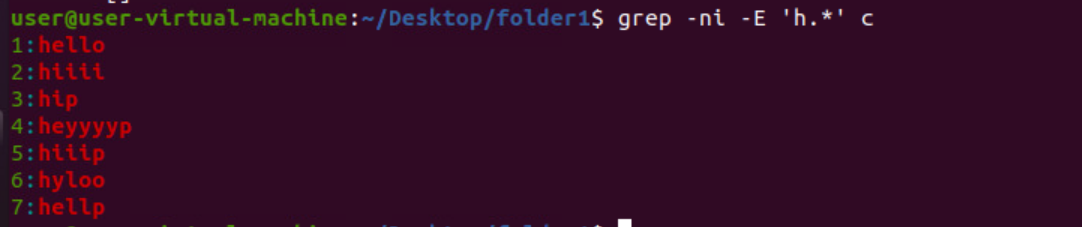
* **\*:**

**-zero or more occurrences of the previous character**

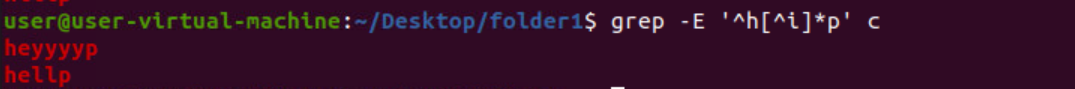


* **(dot).\*:**

**-Nothing or any numbers of characters.**

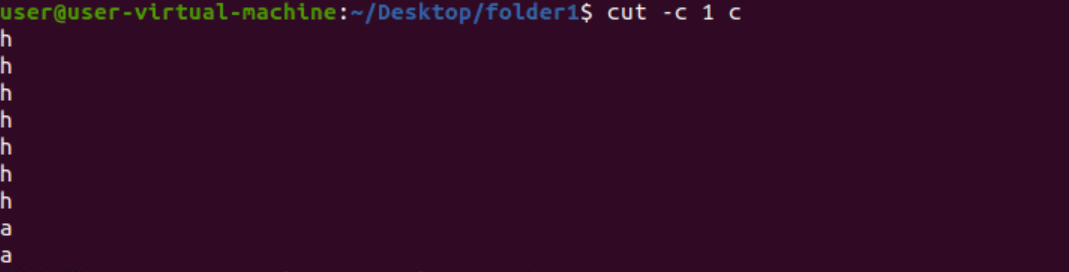


#Task – To eliminate a occurrence of given character in between the given starting and ending character

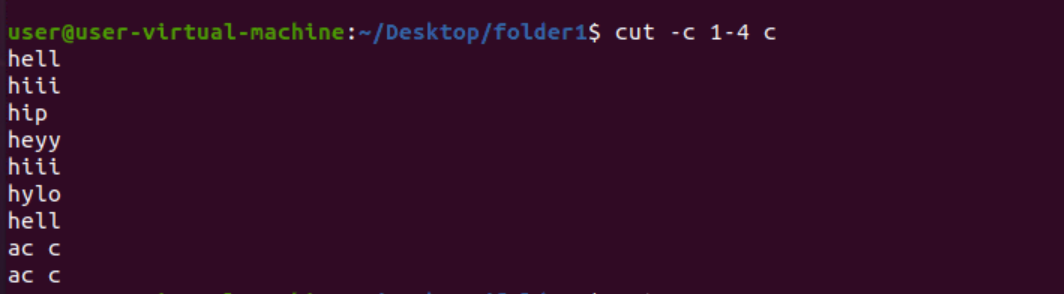


**Cut commands:**

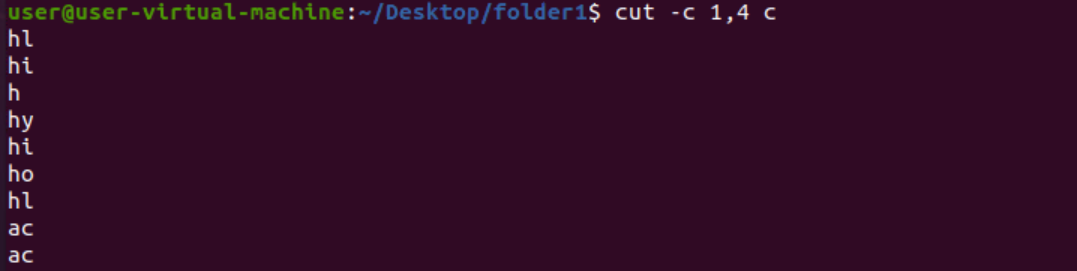
**cut -c 1 file name** - it prints the all first character.



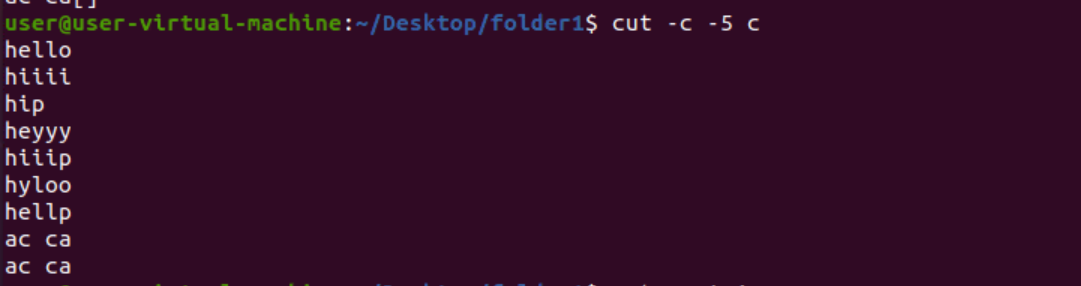
**cut -c 1-4 file name** - it prints the all character from 1 to 4



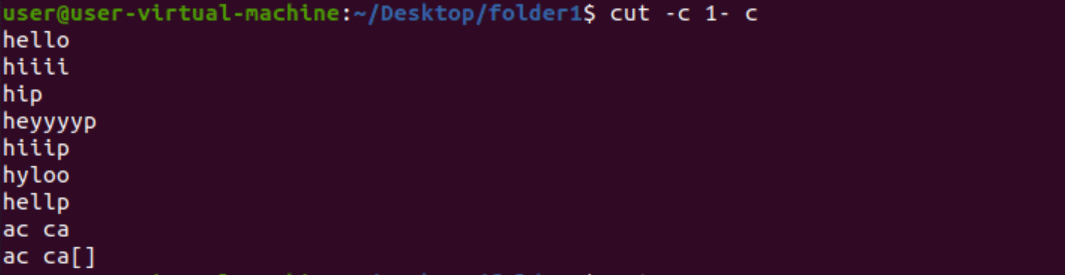
**cut -c 1,4 file name** - it prints the all 1,4 character.



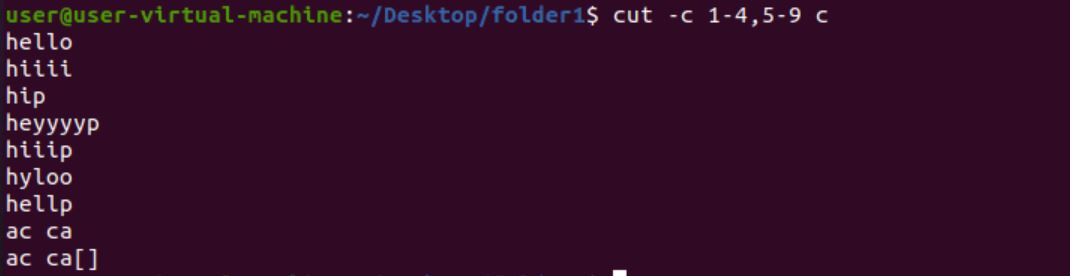
**cut -c -5 file name** - it prints the all character from first -5 .



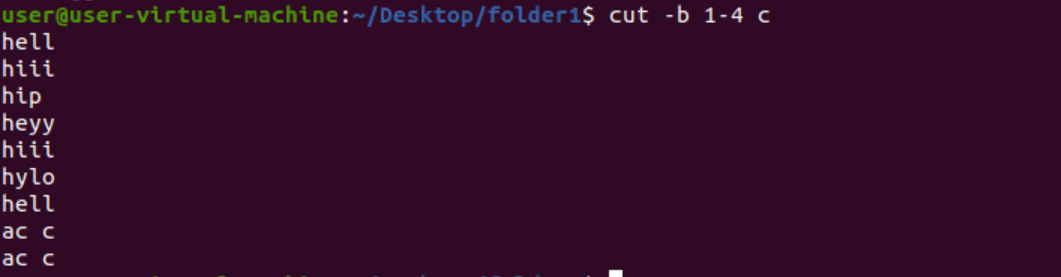
**cut -c 1- file name** - it prints the all character from 1 -last.



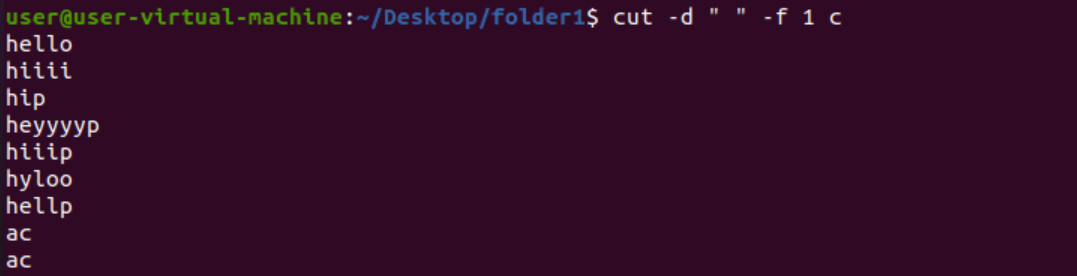
**cut -c 1-4,5-9 file name** - it prints the all 1-4 and 5-9 character.



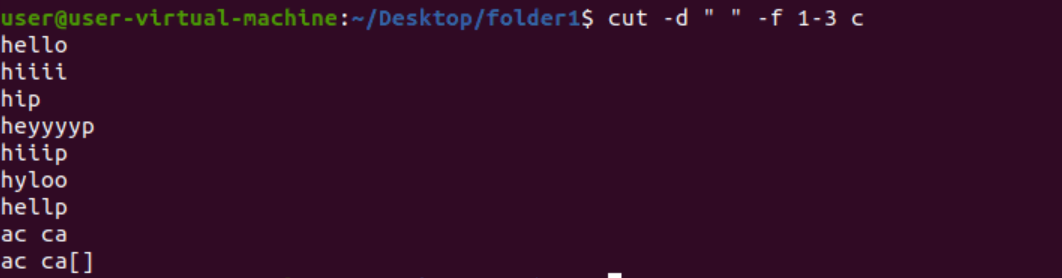
**cut -b 1-4 file name** - it prints the all character from 1 to 4 as same as cut but using address.



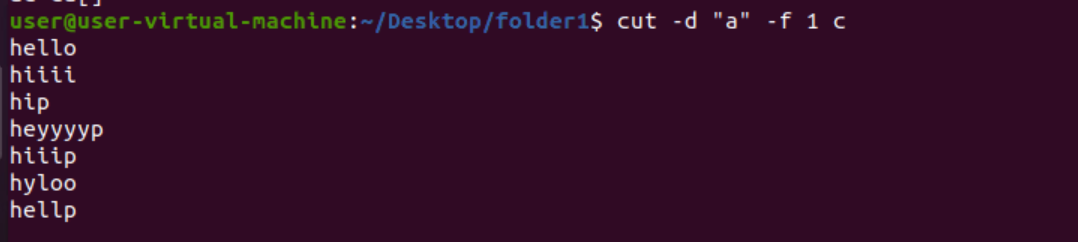
**cut -d " " -f 1 file name** - it prints the characters before the first space.



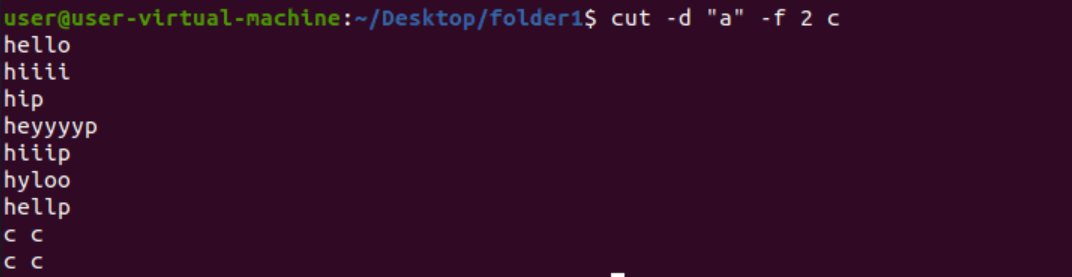
**cut -d " " -f 1-3 file name** - it prints the characters before the first to third space.



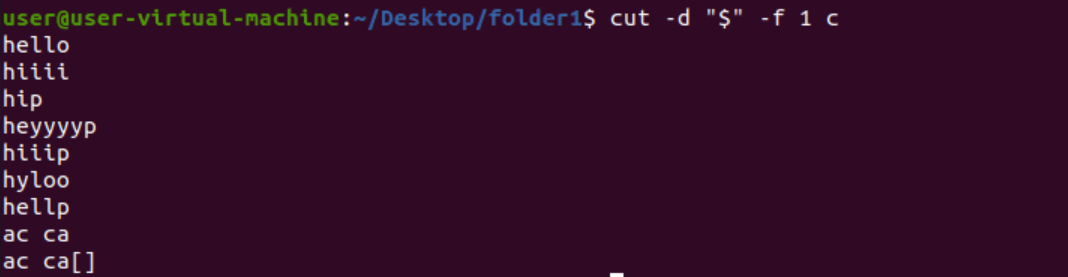
**cut -d "a" -f 1file name** - it prints the characters before the a.



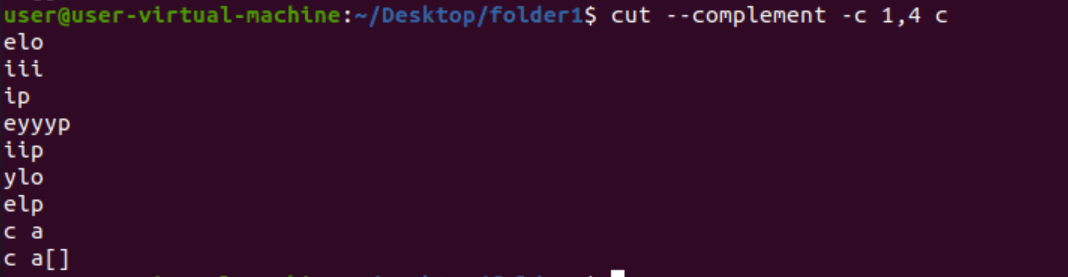
**cut -d "a" -f 2file name** - it prints the characters before the second a.



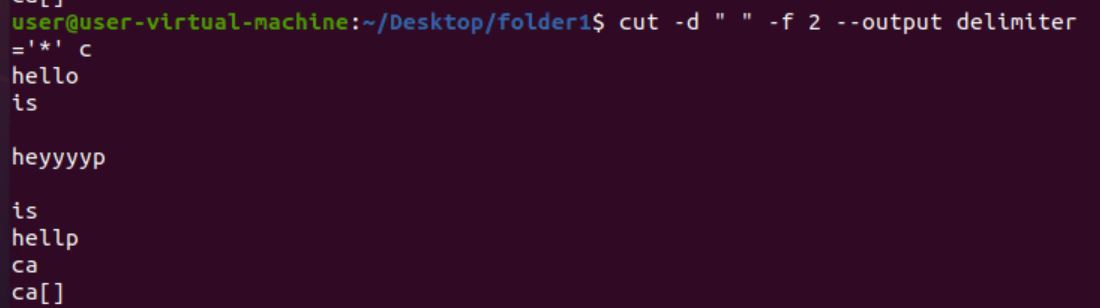
**cut -d "$" -f 1 file name** - it prints the characters before the $.



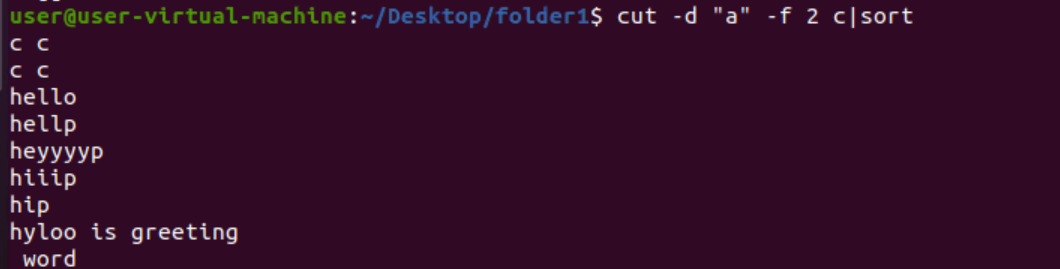
**cut --complement -c 1,4 file name** - it prints all characters except 1,4.



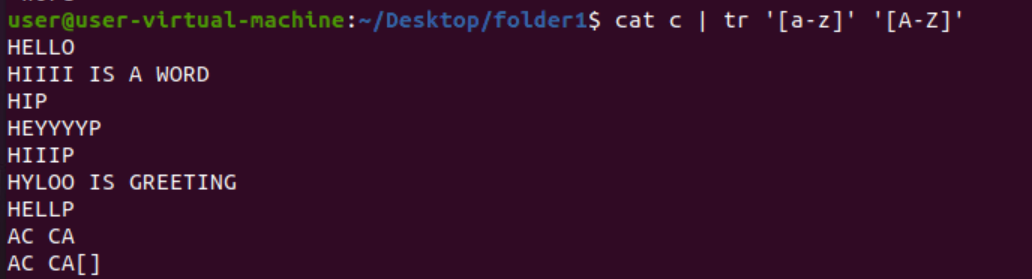
**cut -d " " -f 2 --output-delimiter='\*' file name** - it replaces the spaces with \* till the second space.



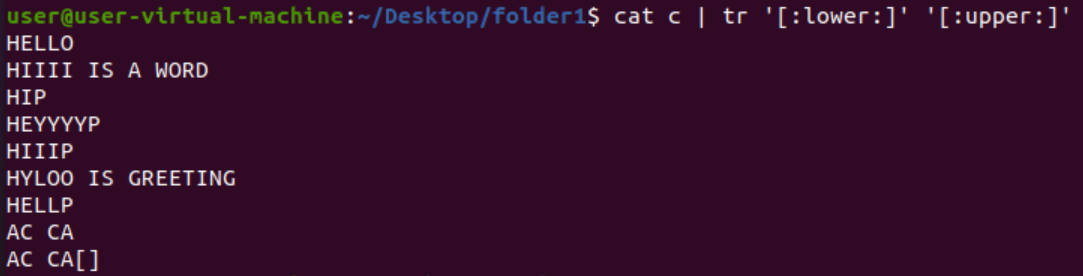
**cat filename | cut -d " " -f 1 file name | sort -** it makes the multiple commands to use at a time



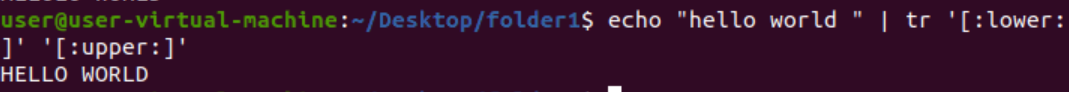
**cat filename | tr '[a-z]' '[A-Z]'** - it will convert every lower case into upper case.



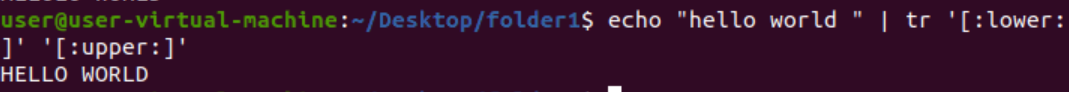
**cat filename | tr '[:lower:]' '[:upper:]'** - it will convert every lower case into upper case.



**echo "hello world" | tr '[:lower:]' '[:upper:]'** - it will convert every lower case into upper case.



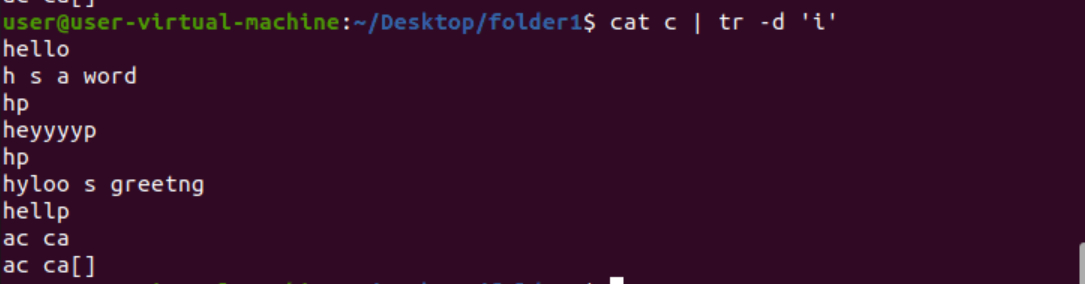
**echo "hello world" | tr ' ' '/n'** - it will convert a space into new line.



**cat filename | tr -s ‘[:space:]’ ' '** -it will squeeze all the spaces.



**cat filename | tr -d ' I '** - it will delete i



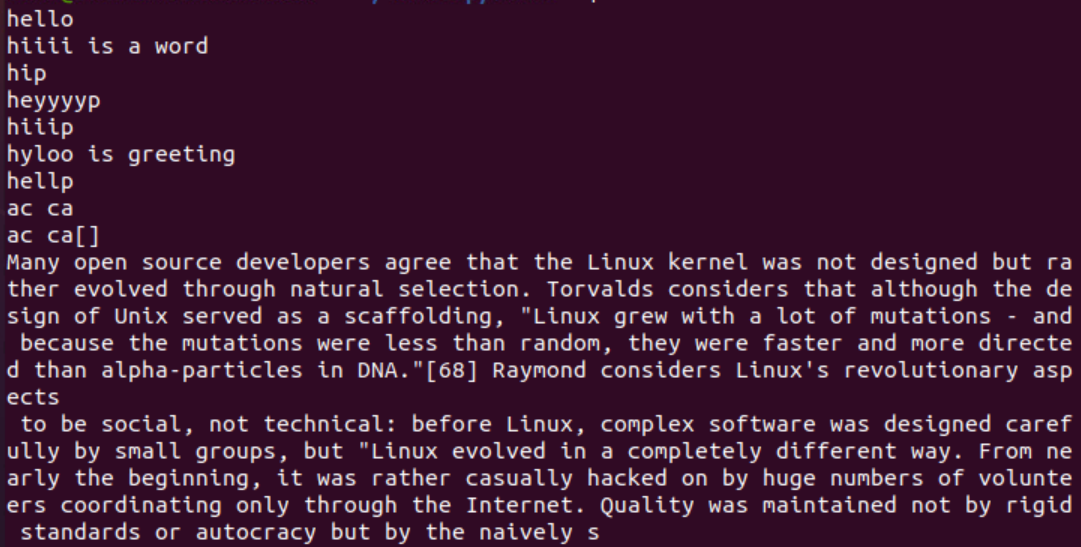
**More commands:**

**more filename –** it will show as page by page = b for previous page enter for line-by-line space for full page iteration.

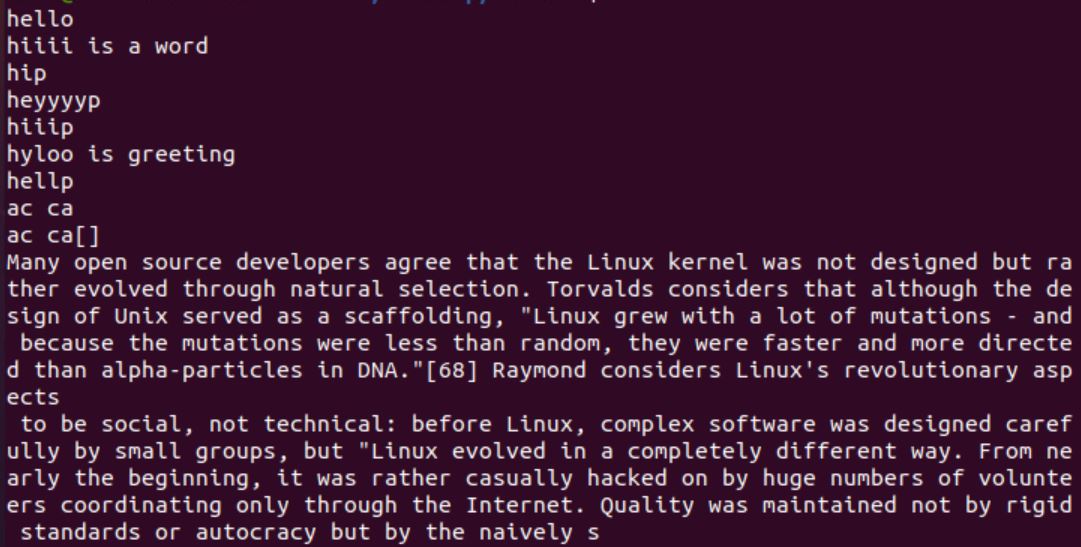


**more -p filename -** it will clear the page and print next data.





**more -c filename** - it will replace the data in the same page.



**more -s filename** - it will eliminate all the extra spaces.

