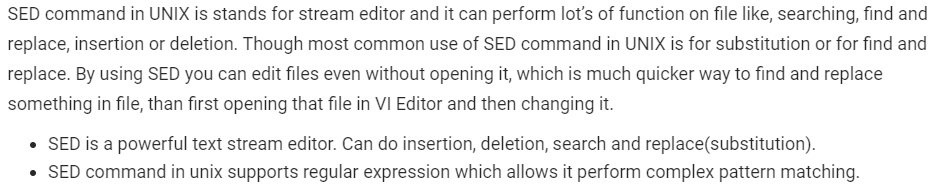
**sed command:**



**Examples:**

The below simple sed command replaces the word “unix” with “linux” in the file. By default, the sed command replaces the first occurrence of the pattern in each line and it won’t replace the second, third…occurrence in the line.

* **$sed 's/unix/linux/' geekfile.txt**

Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word “unix” with “linux” in a line. As below it replaces only second occurrence.

* **sed 's/unix/linux/2' geekfile.txt**

The substitute flag /g (global replacement) specifies the sed command to replace all the occurrences of the string in the line. Below is the example to replace all the occurrences.

* **sed 's/unix/linux/g' geekfile.txt**

Use the combination of /1, /2 etc and /g to replace all the patterns from the nth occurrence of a pattern in a line. The following sed command replaces the third, fourth, fifth… “unix” word with “linux” word in a line. Below is the example to replace from the third occurrence to nth occurrence.

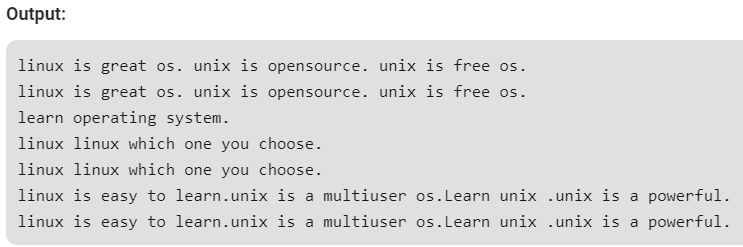
* **sed 's/unix/linux/3g' geekfile.txt**

You can restrict the sed command to replace the string on a specific line number. An example is

* **sed '3 s/unix/linux/' geekfile.txt**

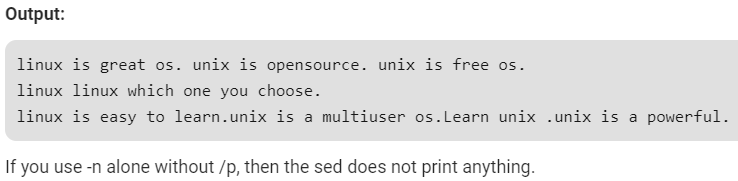
The /p print flag prints the replaced line twice on the terminal. If a line does not have the search pattern and is not replaced, then the /p prints that line only once.

* **sed 's/unix/linux/p' geekfile.txt**



Use the -n option along with the /p print flag to display only the replaced lines. Here the -n option suppresses the duplicate rows generated by the /p flag and prints the replaced lines only one time.

* **sed -n 's/unix/linux/p' geekfile.txt**



To print single line from a file

* **sed -n LINE\_NUMBERp file.txt**
* **sed -n 4p file.txt**

You can specify a range of line numbers to the sed command for replacing a string. Here the sed command replaces the lines with range from 1 to 3.

* **sed '1,3 s/unix/linux/' geekfile.txt**

Here $ indicates the last line in the file. So the sed command replaces the text from second line to last line in the file.

* **sed '2,$ s/unix/linux/' geekfile.txt**

SED command can also be used for deleting lines from a particular file. SED command is used for performing deletion operation without even opening the file

To Delete a particular line, say n in this example

* **sed 'nd' filename.txt**
* **sed '5d' filename.txt**

To Delete a last line

* **sed '$d' filename.txt**

To Delete line from range x to y

* **sed 'x,yd' filename.txt**
* **sed '3,6d' filename.txt**

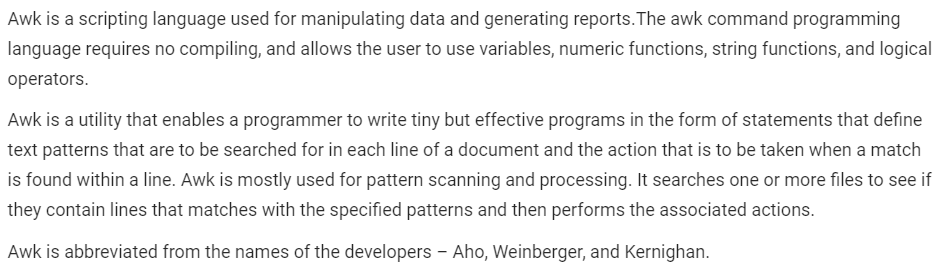
To Delete from nth to last line

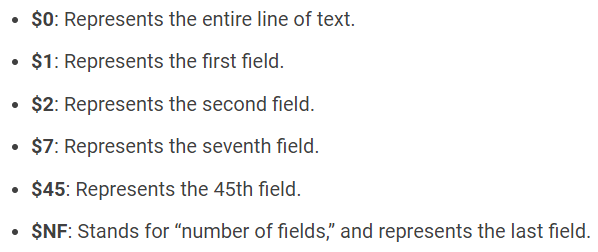
* **sed 'nth,$d' filename.txt**
* **sed '12,$d' filename.txt**

To Delete pattern matching line

* **sed '/pattern/d' filename.txt**
* **sed '/abc/d' filename.txt**

**awk command:**

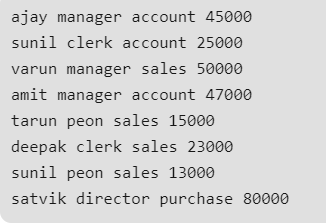




**Examples:**

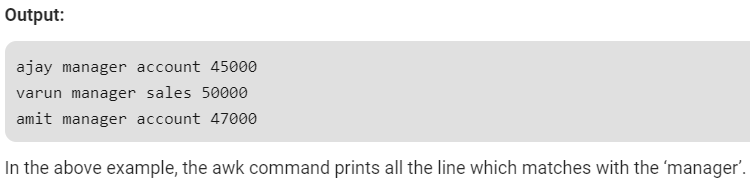
By default Awk prints every line of data from the specified file.

* **awk '{print}' employee.txt**



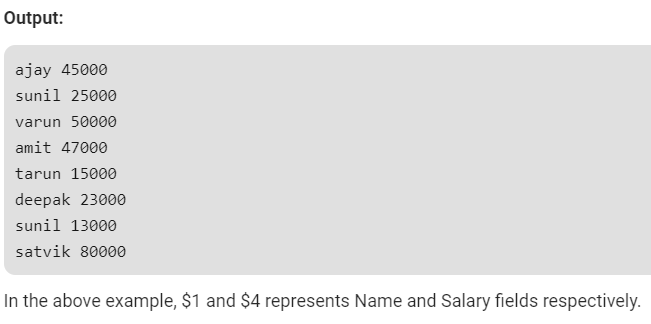
Print the lines which matches with the given pattern.

* **awk '/manager/ {print}' employee.txt**



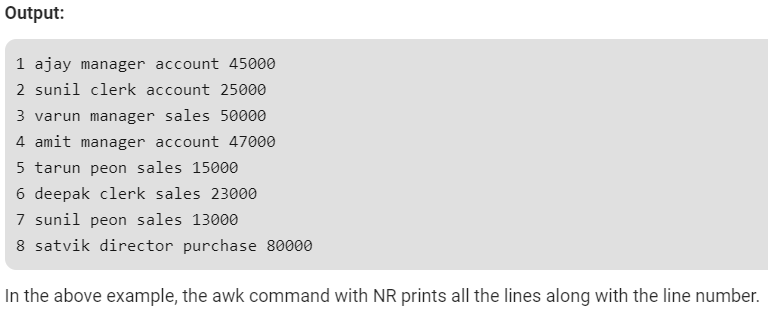
For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

* **awk '{print $1,$4}' employee.txt**



Use of NR built-in variables (Display Line Number)

* **awk '{print NR,$0}' employee.txt**



* **awk '{print $1,$NF}' employee.txt**

