sed Command

sed command

- SED command in Linux stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion.
- Though most common use of SED command in Linux is for substitution or for find and replace.
- By using SED, you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.
- SED is a powerful text stream editor. Can do insertion, deletion, search and replace (substitution).
- SED command in Linux supports regular expression which allows it perform complex pattern matching.

sed command

- Syntax:
- sed OPTIONS... [SCRIPT] [INPUTFILE...]

- Example: Consider the below text file as an input.
- $cat > file2.txt
- Linux is opensource. Linux is free os.

sed command

1. Replacing or substituting string: Sed command is mostly used to replace the text in a file.

- The below simple sed command replaces the word “Linux” with “unix” in the file.

- sed ‘s/Linux/unix/' file2.txt

Output:

- unix is opensource. Linux is free os.

sed command

- Here the “s” specifies the substitution operation.

- The “/” are delimiters.

- The “unix” is the search pattern and the “linux” is the replacement string.

- 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 command

2. Replacing the nth occurrence of a pattern in a line: 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.
- sed 's/unix/linux/2' file2.txt

Output:

- unix is opensource. unix is free os.

sed command

3. Replacing all the occurrence of the pattern in a line.
The substitute flag /g (global replacement) specifies the sed command to replace all the occurrences of the string in the line.

- sed 's/unix/linux/g' file2.txt

Output:
linux is opensource. linux is free os.

sed command

4. Replacing from nth occurrence to all occurrences in a line.

- 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.
- sed 's/unix/linux/2g' geekfile.txt

Output:

```
unix is opensource. linux is free os
```

sed command

5. Parenthesize first character of each word.

This sed example prints the first character of every word in parenthesis.

```
$ echo "Welcome To The Linux Class" | sed
's/\(\b[A-Z]\)/\(\1\)/g’
```

Output: (W)elcome (T)o (T)he (L)inux (C)lass

sed command

6. Replacing string on a specific line number.

- You can restrict the sed command to replace the string on a specific line number.

- An example is
- sed '3 s/unix/linux/' file2.txt

Output:

unix is opensource. unix is free os.

sed command

- 7. Duplicating the replaced line with /p flag.

- 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' file2.txt
  ```

Output:

```
Linux is opensource. unix is free os.
Linux is opensource. unix is free os.
```

sed command

- 8. Printing only the replaced lines.

- 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' file2.txt

Output: unix is opensource. unix is free os.

sed command

- To view particular line from a file.

```
sed –n ‘4p’ file2.txt
```

- To replace word from a file.

```
sed ‘S/oldword/newword/g’ f1
```

(/ is delimiter, S substitute, g globally)

- To delete a particular line from a file.

```
sed ‘4d’ file2.txt (temporarily)
sed –i ‘4d’ file2.txt (permanent)
```

sed command

- To delete empty lines from a file.

```
sed ‘ /^$/d’ file1.txt ( temporarily)
sed –i ‘ /^$/d’ file1.txt (permanent)
```

- To delete range of lines from a file.

```
sed –i ‘6,11-d’ file1.txt
```

- To delete multiple particular lines from a file.

```
sed –i ‘3d;4d;7d;’ file1.txt
```

sed command

- To delete last line from file.

```
sed –i ‘$d’ file1.txt
sed –i ‘1d’ file1.txt
```

- To delete first to last lines from a file.(All lines)

```
sed –i ‘1d,$d’ file1.txt
```

- To delete first and last line from a file.

```
sed –i ‘1d;$d’ file1.txt
```