

# CONTENT

- **sed** to edit an input stream and understand its addressing mechanism
- Line addressing
  - Using multiple instructions
- Context addressing
- Writing selected lines to a file
- Text editing
  - Inserting, changing and deleting lines

# CONTENT

- Substitution
  - The remembered pattern
- Basic regular expressions revisited
  - The repeated pattern
  - Interval regular expression
  - The tagged regular expression
- conclusion

# SUBSTITUTION

- Substitution is the most important feature of sed, and this is one job that sed does exceedingly well

`[address]s/expression1/expression2/flags`

- Just similar to the syntax of substitution in vi editor, we use it in sed also

# SUBSTITUTION

- Substitution is the most important feature of sed, and this is one job that sed does exceedingly well

`[address]s/expression1/expression2/flags`

- Just similar to the syntax of substitution in vi editor, we use it in sed also

```
sed 's/|/:/' emp.lst | head -n 2
```

```
2233:a.k.shukla |gm |sales |12/12/52|6000
```

```
9876:jai sharma |director|production|12/03/50|7000
```

- Only the first instance of | in a line has been replaced.  
We need to use the g (global) flag to replace all the pipes

```
sed 's/|/:/g' emp.lst | head -n 2
```

- We can limit the vertical boundaries too by specifying an address (for first three lines only)  
sed '1,3s/|/:/g' emp.lst
- Replace the word director with member in the first five lines of emp.lst  
sed '1,5s/director/member/' emp.lst

- sed also uses regular expressions for patterns to be substituted
- To replace all occurrence of agarwal, aggarwal and agrawal with simply Agarwal, we have,

```
sed 's/[Aa]gg*[ar][ar]wal/Agarwal/g' emp.lst
```

- We can also use ^ and \$ with the same meaning
- To add 2 prefix to all emp-ids,

```
sed 's/^/2/' emp.lst | head -n 1
```

```
22233 | a.k.shukla | gm | sales | 12/12/52 | 6000
```

- To add .00 suffix to all salary,

```
sed 's/$/.00/' emp.lst | head -n 1
```

```
2233 | a.k.shukla | gm | sales | 12/12/52 | 6000.00
```



# Performing multiple substitutions

```
sed 's/<I>/<EM>/g  
s/<B>/<STRONG>/g  
s/<U>/<EM>/g' form.html
```

An instruction processes the output of the previous instruction, as sed is a stream editor and works on data stream

```
sed 's/<I>/<EM>/g  
s/<EM>/<STRONG>/g' form.html
```

- When a 'g' is used at the end of a substitution instruction, the change is performed globally along the line. Without it, only the left most occurrence is replaced
- When there are a group of instructions to execute, you should place these instructions in a file instead and use sed with the -f option

# Compressing multiple spaces

```
sed 's/*|/|/g' emp.lst | tee empn.lst | head -n 3  
2233|a.k.shukla|g.m|sales|12/12/52|6000  
9876|jai sharma|director|production|12/03/50|7000  
5678|sumit chakrobarty|dgm|mrking|19/04/43|6000
```

# The remembered patterns

- Consider the below three lines which does the same job

```
sed 's/director/member/' emp.lst
```

```
sed '/director/s//member/' emp.lst
```

```
sed '/director/s/director/member/' emp.lst
```

The // representing an empty regular expression is interpreted to mean that the search and substituted patterns are the same

```
sed 's/|//g' emp.lst    removes every | from file
```

# BRE - REVISITED

Three more additional types of expressions are:

- The repeated patterns - &
- The interval regular expression (IRE) – { }
- The tagged regular expression (TRE) – ( )

## The repeated patterns - &

- To make the entire source pattern appear at the destination also

```
sed 's/director/executive director/' emp.lst
```

```
sed 's/director/executive &/' emp.lst
```

```
sed '/director/s//executive &/' emp.lst
```

Replaces director with executive director where & is a repeated pattern

## The interval RE - { }

- sed and grep uses IRE that uses an integer to specify the number of characters preceding a pattern. The IRE uses an escaped pair of curly braces and takes three forms:
  - ch\{m\} – the ch can occur m times
  - ch\{m,n\} – ch can occur between m and n times
  - ch\{m,\} – ch can occur at least m times
- The value of m and n can't exceed 255

- Let teledir.txt maintains landline and mobile phone numbers
- To select only mobile numbers, use IRE to indicate that a numerical can occur 10 times  
`grep '[0-9]\{10\}' teledir.txt`
- Line length between 101 and 150  
`grep '^.\{101,150\}$' foo`
- Line length at least 101  
`sed -n '/.{101,}/p' foo`



# The TAGGED RE – ( )

- You have to identify the segments of a line that you wish to extract and enclose each segment with a matched pair of escaped parenthesis
- If we need to extract a number, `\([0-9]*\)`
- If we need to extract non alphabetic characters, `\([^a-zA-Z]*\)`
- Every grouped pattern automatically acquires the numeric label `n`, where `n` signifies the `n`th group from the left

```
sed 's/ \ (a-z]*\) *\ ([a-z]*\) / \2, \1/' teledir.txt
```

To get surname first followed by a , and then the name and rest of the line

# CONCLUSION

- Substitution
  - The remembered pattern
- Basic regular expressions revisited
  - The repeated pattern
  - Interval regular expression
  - The tagged regular expression

- THANK YOU