UCS1304 UNIX and Shell Programming

SED

B.E. CSE B, Semester 3 (2019-2020)

S Milton Rajendram

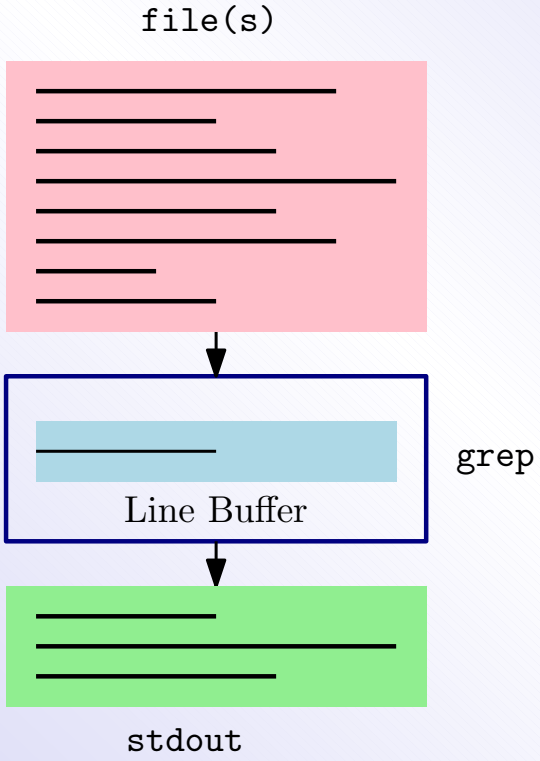
Department of Computer Science and Engineering

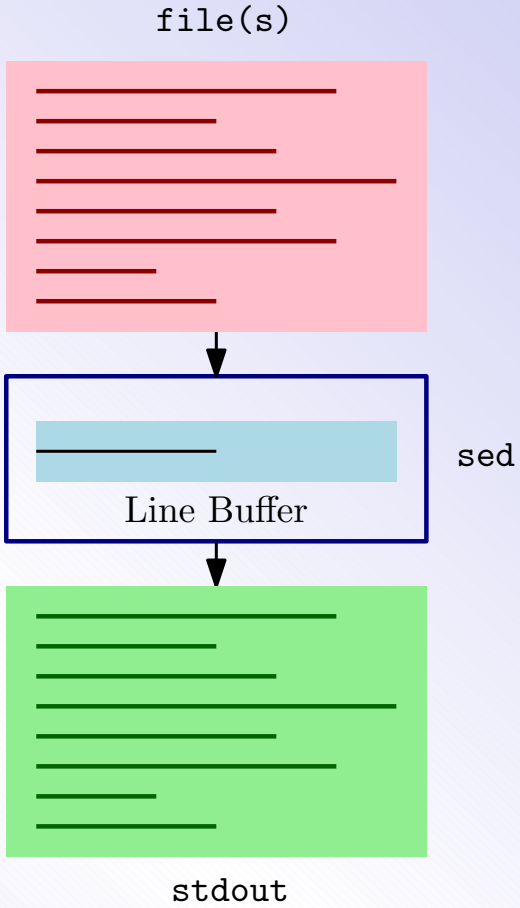
SSN College of Engineering

9 August 2019

1. sed

- **s**tream **ed**itor





- Syntax

```
sed commands ...  filenames
```

- read line
- edit the line using commands
- write edited line

```
selector1 command1
selector2 command2
...
selectorm commandm
```

```
sed -f script filename(s)
```

or

```
sed -e 'command1' -e 'command2' ... filename(s)
```

- Meaning

```
for each line in input:
  for each command in commands:
     if line is in selector of command:
        apply command on line
```

- By default, sed prints every line and only edits lines that match a specified address within the file. The default behavior can be overridden by specifying the `-n` option.

2. Input and output

- Redirect output

```
sed 'cmds' ifile > ofile
```

3. Line selector (Address)

- `n`
  A line number where n is a positive integer.

  ```
  cat -n  files/augustine.txt | sed -n '9p'
  ```

- `$`
  The last line.

  ```
  cat -n  files/augustine.txt | sed -n '$p'
  ```

- `/regexp/`
  Lines matching a basic regular expression.

  ```
  cat -n  files/augustine.txt | sed -n '/flight/p'
  ```

- `addr1,addr2`
  A range of lines from `addr1` to `addr2`, inclusive. Addresses may be any of the single address forms listed earlier.

```
cat -n  files/augustine.txt | sed -n '5,9p'
```

```
cat -n  files/augustine.txt | sed -n '/wing/,/flight/p'
```

- `first ~ step`
  Match the line represented by the number `first`, then each subsequent line at `step` intervals.
  - `1~2` each odd numbered line
  - `5~5` the fifth line and every fifth line thereafter.

```
cat -n  files/augustine.txt | sed -n '1~5p'
```

- `addr1,+n`

  Match addr1 and the following n lines.

```
cat -n  files/augustine.txt | sed -n '6,~3p'

6 We have not wings, we cannot soar;
7       But we have feet to scale and climb
8 By slow degrees, by more and more,
9       The cloudy summits of our time.
```

- `addr!`

  Match all lines except addr, which may be any of the forms listed earlier.

4. Commands (Editing Operations)

- `=`
  Output the current line number.
- `i`
  Insert text before the current line.
- `a`
  Append text after the current line.
- `c`
  Change lines to following text as in `a`
- `d`
  Delete the current line.
- `p`
  Print the current line.

- `q`
  Exit sed without processing any more lines.
- `s/regexp/replacement/`
  For every regex match, substitute the replacement
  - `&` is equivalent to the text matched by regexp.
  - `\1` through `\9` are the contents of the corresponding subexpressions in regexp (back references). The trailing slash may be followed by an optional flag.
- `y/set1/set2`
  Transliterate characters from `set1` to the corresponding characters in `set2`. Both sets must be of the same length.

5. Commands

```
sed '' distros.txt

sed 'p' distros.txt

sed -n '' distros.txt

sed -n 'p' distros.txt
```

6. `s` command

```
s/search/replacement/
```

- `search` is a RE
- For every `match`, substitute `replacement`
- `&` is equivalent to the text matched by regexp.
- `\1` through `\9` are the contents of the corresponding subexpressions in regexp (back references). The trailing slash may be followed by an optional flag.

```
cat -A distros.txt
```

```
sed 's/SUSE/Suse/' distros.txt
```

```
sed 's/ */:/' distros.txt
```

```
sed 's/  */:/' distros.txt

sed 's/  */:/g' distros.txt

echo "aaabbbccc" | sed 's/b/B/'

# global = all matches in a line
echo "aaabbbccc" | sed 's/b/B/g'

# global = all matches in a line
sed 's/  */\t/g' distros.txt | cat -A

# replacement is empty = delete the match
sed 's/^  *//' sed 's/^  *//' files/augustine.txt

sed 's/$/\n/' files/distros.txt

# longest match
sed 's/.* //' distros.txt
```

7. Line selectors (Line numbers)

- Line number

```
# line 3
sed '3s/  */:/g' distros.txt | cat -n
```

- Range of line numbers

```
# lines 1 to 5
sed '1,5p' distros.txt | cat -n
```

```
# lines 1 to 5
sed '1,5s/  */:/g' distros.txt | cat -n
```

```
# lines 1 to 5
sed '1,5s/SUSE/Suse/g' distros.txt
```

- Last line

```
# last line
sed '$s/  */:/g' distros.txt | cat -n
```

- Commands
  - q

    ```
    cat -n distros.txt | sed '5q'
    ```

  - d

    ```
    cat -n distros.txt | sed '4,6d'
    ```

  - i

    ```
    cat -n distros.txt | sed '2i ==========================='
    ```

  - a

    ```
    cat -n distros.txt | sed '1a ==========================='
    ```

8. Line selectors (regular expressions)

```
cat -n distros.txt | sed -n '/SUSE/p'
```

```
cat -n distros.txt | sed -n '/SUSE/!p'
```

9. Back reference

```
sed 's/\([0-9]\{2\}\)\/\([0-9]\{2\}\)\/\([0-9]\{4\}\)$/\3-\1-\2/' di
```

```
[0-9]{2}/[0-9]{2}/[0-9]{4}$
([0-9]{2})/([0-9]{2})/([0-9]{4})$
([0-9]{2})/([0-9]{2})/([0-9]{4})$/\3-\1-\2
([0-9]{2})\/([0-9]{2})\/([0-9]{4})$/\3-\1-\2
\([0-9]\{2\}\)\/\([0-9]\{2\}\)\/\([0-9]\{4\}\)$/\3-\1-\2/
```

10. Script file

- Multiple commands

```
sed -e 'cmd1' -e 'cmd2' files/distros.txt
```

- Commmands in a file – sed script

```
sed -f script.sed files/distros.txt
```

11. Insert command

```
# sed script to produce Linux distributions report
1 i\
\
Linux Distributions Report\
s/\([0-9]\{2\}\)\/\([0-9]\{2\}\)\/\([0-9]\{4\}\)$/\3-\1-\2/
y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/
```

12. Multiple commands

```
sed -e 's/^/\t/' -e '3q' files/distros.txt
```

13. Cooperation with shells

- Newer files

```
# newer f: list files newer than f
ls -t | sed '/^'$1'$/q'
```

- Older files

14. Examples

```
sed -n '20,30p' sed '$d' # Print only lines 20 through 30
sed '1,10d'              # Delete lines 1 through 10 (~tail -n +11~
sed ' 1,/^$/d'           # Delete up to and including first blank l
sed -n '/^$/,/^end/p'    # Print each group of lines from an empty
sed '$d'                 # Delete last line
```

15. Summary

a\	append lines to output until one not ending in \
b label	branch to command : label
c\	change lines to following text as in a
d	delete line; read next input line
i\	insert following text before next output
l	list line, making all non-printing characters visible
p	print line
q	quit
r file	read file, copy contents to output
s/old/new/f	substitute new for old. If f = g, replace all occurrences; f = p, print; f = w file, write to file
t label	test: branch to label if substitution made to current line
w file	write line to file
y/str1/str2/	replace each character from str 1 with corresponding character from str2 (no ranges allowed)
=	print current input line number