CSCI 330
The UNIX System

Unit XI

sed - Stream Editor

What is sed?

- A non-interactive stream editor
- Interprets sed instructions and performs actions

- Use sed to:
  - Automatically perform edits on file(s)
  - Simplify doing the same edits on multiple files
  - Write conversion programs

  - Do editing operations from shell script

sed command syntax

```
$ sed -e 'address command' input_file
```

(a) Inline Script

```
$ sed -f script.sed input_file
```

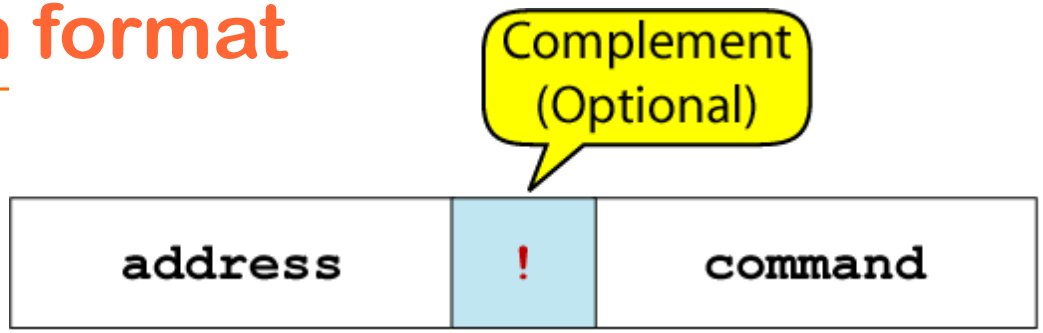
(b) Script File

How Does sed Work?

- sed reads line of input
  - line of input is copied into a temporary buffer called <u>pattern</u> space
  - editing instructions are applied to line in pattern space
  - line is sent to output
    (unless –n option was used)
  - line is removed from pattern space
- sed reads next line of input, until end of file

<u>Note:</u> input file is unchanged
unless "-i" option is used

sed instruction format



- address determines which lines in the input file are to be processed by the command(s)

- address types:
  - No address (all lines)
  - Single-Line address
  - Set-of-Lines address
  - Range address
  - Nested address

Single-Line Address

- Specifies only one line in the input file
  - special: dollar sign ($) denotes last line of input file

Examples:

- show only line 3

```
sed -n -e '3 p' input-file
```

- show only last line

```
sed -n -e '$ p' input-file
```

- substitute “endif” with “fi” on line 10

```
sed -e '10 s/endif/fi/' input-file
```

Set-of-Lines Address

- use regular expression to match lines
  - written between two slashes
  - process only lines that match
  - may match several lines
  - lines may or may not be consecutives

Examples:

```
sed –e '/key/ s/more/other/' input-file

sed –n –e '/r..t/ p' input-file
```

Range Address

- Defines a set of consecutive lines

Format:

start-addr,end-addr (inclusive)

Match on start-addr starts processing. Match on end-addr turns off processing

Examples:

10,50 line-number,line-number

10,/R.E/ line-number,/RegExp/

/R.E./,10 /RegExp/,line-number

/R.E./,/R.E/ /RegExp/,/RegExp/

Example: Range Address

```
% sed -n -e '/^BEGIN$/,/^END$/p' input-file
```

addr1

addr2

- Print lines between BEGIN and END, inclusive

```
BEGIN
Line 1 of input
Line 2 of input
Line3 of input
END
Line 4 of input
Line 5 of input
```

These lines are printed

Nested Address

- Nested address contained within another address

Example:

print blank lines between line 20 and 30

```
- 20,30{
  /^$/ p
- }
```

Address with !

- address with an exclamation point (!):

  instruction will be applied to all lines that do not match the address

<u>Example:</u>

print lines that do not contain “obsolete”

```
sed -n -e '/obsolete/!p' input-file
```

sed Commands

- line number
- modify
  - insert, append,
  - change
  - delete
  - substitute
- I/O
  - next, print
  - read, write
- quit

Line Number

- line number command (=)
  writes the current line number

Examples:

```
sed -n -e '/key/=' inputFile
sed -n -e '/^[0-9][0-9]/=' inventory
```

Command - “i” “a” “c”

“i” add lines(s) before the address
“a” adds line(s) after the address
“c” replaces an entire matched line with new text

Syntax:
[address] i\
text

Insert Command: i

- adds one or more lines directly to the output before the address:
  - inserted “text” never appears in sed’s pattern space
  - cannot be used with a range address; can only be used with the single-line and set-of-lines address types

Syntax:

```
[address] i\

text
```

Example: Insert Command (i)

```
% cat tut.insert.sed
1 i\
        Tuition List\
```

Sed script to insert “Tuition List” as report title before line 1

```
% cat tuition.data
Part-time          1003.99
Two-thirds-time    1506.49
Full-time          2012.29
```

Input data

```
% sed -f tut.insert.sed tuition.data
        Tuition List

Part-time          1003.99
Two-thirds-time    1506.49
Full-time          2012.29
```

Output after applying the insert command

Delete Command: d

- deletes the entire pattern space
  - commands following the delete command are ignored since the deleted text is no longer in the pattern space

Syntax:

```
[address1[,address2]] d
```

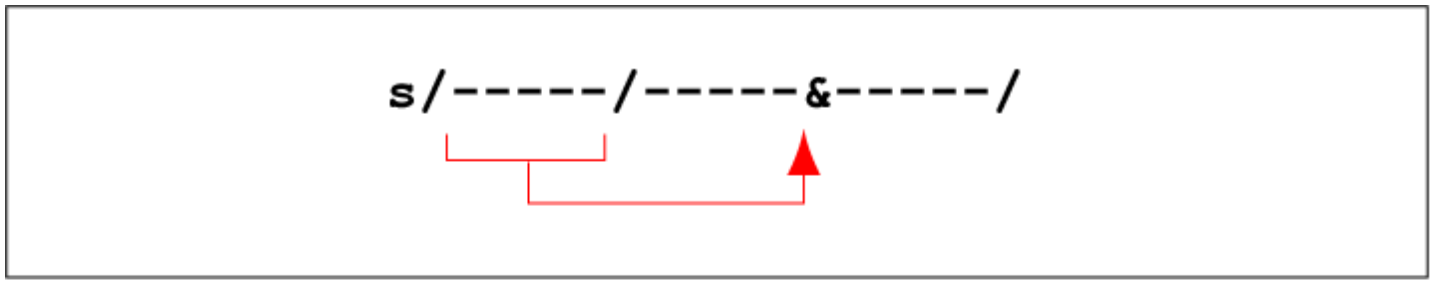
Substitute Command (s)

Syntax:

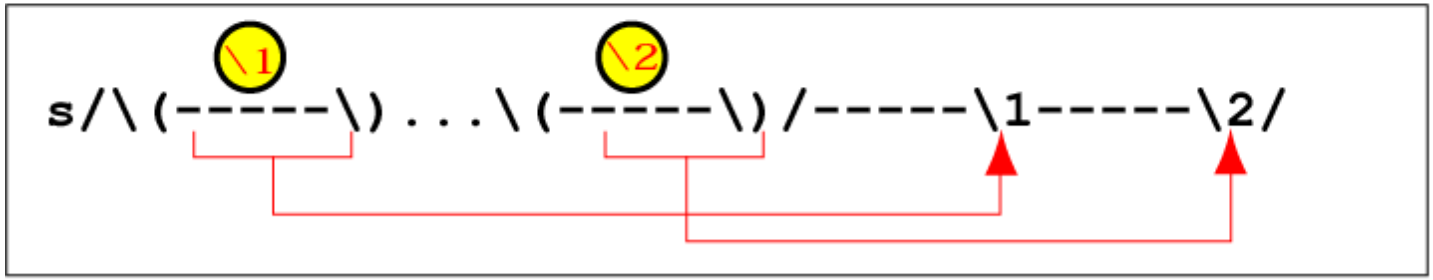
```
[addr1][,addr2] s/search/replace/[flags]
```

- replaces text selected by search string with replacement string
- search string can be regular expression
- flags:
  - global (g), i.e. replace all occurrences
  - specific substitution count (integer), default 1

Substitution Back References



(a) Whole Pattern Substitution



(b) Numbered Buffer Substitution

Example: Replacement String &

```
$ cat datafile
Charles Main            34
Sharon Gray             23
Patricia Jones          7
TB Savage               20
AM Main Jr.             13
Margot Weber            9
Ann Stephens            13

$ sed -e 's/[0-9][0-9]$/&.5/' datafile
Charles Main            34.5
Sharon Gray             23.5
Patricia Jones          7
TB Savage               20.5
AM Main Jr.             13.5
Margot Weber            9
Ann Stephens            13.5
```

Example: Back Reference

```
$ cat name.data
John Doe
Susan Maloney
Harvey Keitel
Randy Newman
Ossie Weaver

$ sed -e 's/\(\<.*\>\) \(\<.*\>\)/\2, \1/g' name.data
Doe, John
Maloney, Susan
Keitel, Harvey
Newman, Randy
Weaver, Ossie
```

Input (next) Command: n and N

- n (lowercase)
  - copies the contents of the pattern space to output
  - deletes the current line in the pattern space
  - refills it with the next input line
  - continue processing
- N (uppercase)
  - adds the next input line to the current contents of the pattern space
  - useful when applying patterns to two or more lines at the same time

Output (print) Command: p and P

- p (lowercase)
  - copies the entire contents of the pattern space to output
  - will print same line twice unless the option "–n" is used
- P (uppercase)
  - prints only the first line of the pattern space
  - prints the contents of the pattern space up to and including a new line character
  - any text following the first new line is not printed

File commands

- allows to read and write from/to file while processing standard input
- r read command
- w write command

quit (q) Command

Syntax: `[addr]q`

- Quit (exit sed) when addr is encountered.

Example: Display the first 50 lines and quit

```
% sed -e '50q' datafile
```

Same as:

```
% sed -n -e '1,50p' datafile

% head -50 datafile
```

Summary: stream editor

- can be called from shell script
- allows systematic wholesale changes to files