OPERATING SYSTEMS

UE22CS242B

PES UNIVERSITY
CELEBRATING 50 YEARS

sed

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau
5. Internet source

PES UNIVERSITY
CELEBRATING 50 YEARS

OPERATING SYSTEMS

sed – introduction https://www.gnu.org/software/sed/manual/sed.html

- sed is a stream editor.
- A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline).
- While in some ways similar to an editor which permits scripted edits (such as ed), sed works by making only one pass over the input(s), and is consequently more efficient.
- But it is sed's ability to filter text in a pipeline which particularly distinguishes it from other types of editors.

OPERATING SYSTEMS

sed - syntax

- sed is invoked like this:

  sed SCRIPT INPUTFILE...

Example: To replace all occurrences of ‘hello’ to ‘world’ in the file input.txt:

sed 's/hello/world/' input.txt > output.txt

- If ***INPUTFILE*** is not specified, sed filters the contents of the standard input
- sed writes output to standard output.
- Use -i to edit files in-place instead of printing to standard output

  sed -i 's/hello/world/' file.txt

- -f options are used to specify a *script*,

  sed -f myscript.sed input.txt > output.txt

  sed --expression='s/hello/world/' input.txt > output.txt

OPERATING SYSTEMS

sed - examples

- Use -n to suppress output, and the p command to print specific lines
- sed treats multiple input files as one long stream.
- The following example prints the first line of the first file (one.txt) and the last line of the last file (three.txt).

  sed -n '1p ; $p' one.txt two.txt three.txt
- Use -s to reverse this behavior.
- **Exit status:** An exit status of zero indicates success, and a nonzero value indicates failure
- Delete lines 30 to 35 in the input. 30,35 is an address range. d is the delete command

  sed '30,35d' input.txt > output.txt

OPERATING SYSTEMS

sed - examples

- Print all input until a line starting with the word ‘foo’ is found. If such line is found, sed will terminate with exit status 42. If such line was not found (and no other error occurred), sed will exit with status 0. /^foo/ is a regular-expression address. q is the quit command

  sed '/^foo/q42' input.txt > output.txt

- Commands within a *script* or *script-file* can be separated by semicolons (;) The following command performs two sed operations: deleting any lines matching the regular expression /^foo/, and replacing all occurrences of the string ‘hello’ with ‘world’:

  sed '/^foo/d ; s/hello/world/' input.txt > output.txt

OPERATING SYSTEMS

sed - examples

- Delete the second input line:

  seq 3 | sed 2d

- Print only the second input line

  seq 3 | sed -n 2p

- Perform substitution on every 3rd line

  seq 6 | sed 'n;n;s/./x/'

- Perform substitution then print the second input line

  seq 3 | sed -n '2{s/2/X/ ; p}‘

- Add the word ‘hello’ after the second line:

  seq 3 | sed '2a hello'

OPERATING SYSTEMS

sed - examples

- Insert the word ‘hello’ before the second line

  seq 3 | sed '2i hello‘

- Replace the 2nd to 9th lines with the word ‘hello’

  seq 10 | sed '2,9c hello‘

- Replace the word ‘hello’ with ‘world’ only on line 144

  sed '144s/hello/world/' input.txt > output.txt

- Replace the word ‘hello’ with ‘world’ only in lines containing the word ‘apple’

  sed '/apple/s/hello/world/' input.txt > output.txt

- Replace the word ‘hello’ with ‘world’ only in lines 4 to 17 (inclusive):

  sed '4,17s/hello/world/' input.txt > output.txt

OPERATING SYSTEMS

sed - examples

- An address range can be specified by specifying two addresses separated by a comma (,)

  seq 10 | sed -n '4,6p‘

- Prints lines containing the word ‘hello’

  sed -n '/hello/p‘

- Replace the words ‘gray’ or ‘grey’ with ‘blue’

  sed 's/gr[ae]y/blue/‘

- Joining specific lines (e.g. if lines 2 and 3 need to be joined)

  sed '2{N;s/\n//;}' lines.txt

THANK YOU

Suresh Jamadagni
Department of Computer Science Engineering

sureshjamadagni@pes.edu