Lecture 21: Filters (contd.)

.cut

- Retrieving selected fields from a file
- $ cut [options] <filename>
- options:
  - –c selects columns specified by list
  - –f selects fields specified by list
  - –d field delimiter (default is tab)

It is often useful to organize information in a file as a table, with rows and columns. Each row of a table represents one record in the file. Each column in a row represents one field in a record, separated from adjacent columns by one or more spaces. This type of organization is best suited for data files, where each row consists of one record and each record is divided into fields. An example can be of an employee file having records of different employees.

The cut command of UNIX is useful, when a file has to be queried to display selective fields from a file. This command picks up selected columns or fields from a file, by specifying a list of character positions or a list of fields.

The command line format is:

$ cut [options] <filename> <Enter>

Examples:

.To display the employee code, firstname and basic salary from the employee file (pg. 4.11)

$ cut -dO " -f2,3,8 employee <Enter>

where

-d is used to specify the field delimiter. A space in this case.

-f represents fields and the numbers after it are the field positions.

The specified columns are copied to the standard output.

.To display characters 4 to 9 both inclusive, from each record of the file employee:

$ cut -c4-9 employee <Enter>

.To display fields from first to fourth:

$ cut -dO " -f-4 employee <Enter>

(note that -4 means 1-4)

.To display from the second to the fourth fields:

$ cut –d" " -f2-4 employee <Enter>

.To display from the fourth field onwards:

$ cut -d" " -f4- employee <Enter>

.Using cut as a Filter

To display a sorted list of the firstname and basic salary of all employees belonging to department "SALES":

$ grep "SALES" employee | cut -do " -f2,8 | sort <Enter>

.paste

- Horizontal merging of files
- $ paste <file1> <file2> <Enter>
- options: -d (field delimiter)

The paste command prints files side by side in a columnar manner. It takes two tables and combines them side by side forming a single wide table as the output. This utility is opposite to cut.

This command uses tab as a field delimiter. However, with the -d option a different delimiter can be given

The command line format is:

$ paste [option] <fi1enamel> <fi1ename2> <Enter>

Example:

Consider the files booklist and pricelist containing the following data. To print the book name along with their price, we need to paste the two files

book1ist price1ist
Advanced UNIX Rebecca Thomas Rs 200
Shell Programming Stephen Kochan Rs 300
Understand UNIX James W Woff. Rs 500
Understanding UNIX Rebecca Thomas. Rs 250
$ paste book1ist price1ist <Enter>

output will be:

Advanced UNIX Rebecca Thomas Rs 200
She11 Programming Stephen Kochan Rs 300
Understand UNIX James W Woff. Rs 500
Understanding UNIX Rebecca Thomas. Rs 250

Examining file contents

- head [-nJ file :display first n lines of a file
- tail [+/- n] file : display last n lines of a file
- wc [-Icw] file : count lines, words-and characters
- pg and more : page wise display a file
- tr : translate characters
- tee : send output to two different destinations

.head

This command by default displays first ten lines of a file.

$ head employee will display first ten lines from employee

$ head -5 emp1oyee will display first five lines from the employee

.tail

This command displays the last ten lines of the file by default.

The number of lines can be changed by providing it along with the tail command.

The options can be of two types:

$ tail employee will display the last ten lines from employee

$ tail -7 employee will display the last seven lines

$ tail +10 employee will display lines from the 10th line till end of the file

.we

The wc (word count) command counts the number of words, characters and lines present in a file. The output consists of three numbers, which are number of lines, number of words and the number of characters in that order.

$ we employee <Enter>

the output may be

24 296 1756 employee

The options available with wc are:

-c: for character count only

-w: for word count only

-I: for line count only

$ wc -c employee may display 1756

$ we -w employee may display 296

$ we - 1 employee .may display 24

The head, tail and wc commands are filters and cah be used anywhere in a pipeline.

Example:

.To count the number of files in the directory:

$ ls | we –1 <Enter>

here the output of Is is the directory listing and is given as input to wc. The -I option of we will count the number of lines in the input, that is the number of files present in the directory.

To count the number of users currently logged in to the system:

$ who I wl –1 <Enter>

.more

- controls the display of a file on the screen
  - $ more [filename]
  - options:
    - + line number : starts the display at the line number
    - +/pattern : starts the display two lines before the pattern
    - -c : clears the screen before display

The more command, reads the contents of a long text file one screenful at a time. To display the next screen the space bar is pressed, if the user presses the <Enter> only the next line is displayed. The more command continues displaying the contents till the EOF (end of file) is reached.

The command line format is:

$ more <filename> <Enter>

This command displays 23 lines, at a time, then the prompt ---More--- is displayed on the last line. To quit before the EOF is reached,' q' can be pressed which will terminate the program and return to the shell prompt.

If the input text comes from the file the message"(xx%)" is displayed after the prompt, telling the percentage of the total file so far displayed.

$ more employee will display the file employee one screenful at a time

more can be used anywhere in a pipeline.

$ ls -1 | more will display the directory contents one screen at a time.

$ ls –1| grep "^d" |I more will display the names of the directories one screenful at a time

.pg

- To display file contents one screenful at a time (somewhat similar to more)
- $ pg filename1 filename2....
- Options:
  - -e: not to pause at the end of each file.
  - -s: prints all messages and prompts in standard output mode

The pg command is a filter which allows the display of files one page (one screen) at a time on the screen. Each page is followed by a prompt at the bottom of the screen. The next page is displayed on pressing the <Enter> key.

The command line format is:

$ pg <filename> <Enter>

When the pg command gives the ": " (colon) prompt the following options can be used:

n to display the next file, if more than one file is specified with pg.

Q or q to terminate the process.

p to display the pages of the immediate previous file

The pg command is used at the end of a pipeline.

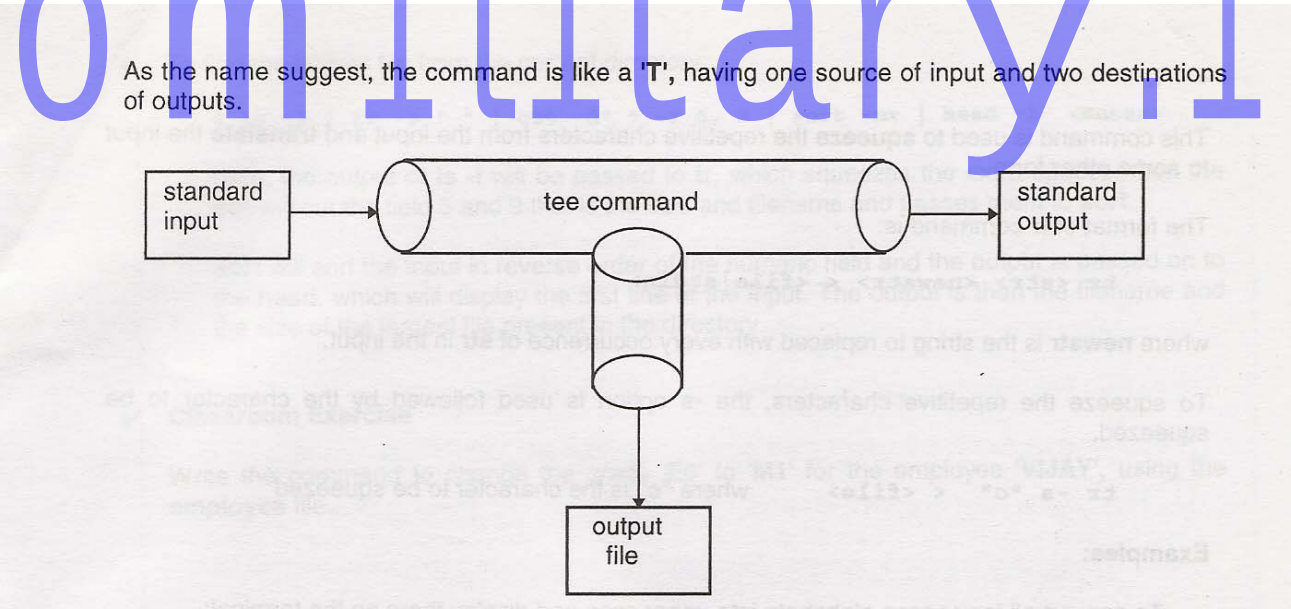
$ ls -1 | pg will display the long listing of a directory pagewise.

.tee

The output redirection will redirect the standard output to a file or some destination other than the terminal. In case the output is required to be sent to two different destinations at the same time, the redirection operator can not be used.

The tee command is used to send the output to more than one destination.

As the name suggest, the command is like a 'T', having one source of input and two destinations of outputs.



tee transfers data without any change from its standard input to its standard output, but at the same time it directs a copy of the data into another file, that is specified at the command line.

This command can be used anywhere in a pipeline.

If the tee command is given at, the $ prompt, it will expect input from the keyboard and at the end of input, it displays the output on the terminal and also stores it in the file specified. The format is:

$ tee <filename> <Enter>

This command differs from other filters as it sends a copy of its output to a disk.

$ <command1> I tee <file> I <cmmand2>

The standard output of the command1 becomes the standard input for tee. This input is then passed on as the standard input forcommand2 and a copy is diverted to file at the same time.

For example, the command:

$ sort employee | tee tempfile| pg

will sort the employee file, display the result on the screen pagewise and store it in tempfile.

.tr

This command is used to squeeze the repetitive characters from the input and translate the input to some other form.

The format of tr command is:

tr <str> <newstr> < <filelstdin>

where newstr is the string to replaced with every occurrence of str in the input.

To squeeze the repetitive characters, the -s option is used followed by the character to be squeezed.

tr -s "c" < <file> where "c" is the character to be squeezed

Examples:

.To convert all .lower case alphabets into upper case and display them on the terminal:

$ cat emp.dat | tr "[a-z]" "[A-Z]"

.Display all the occurrences of small "a" as "A":

$ ls -1 | tr "a" "A"

.To produce the file size and names in descending order of size:

$ ls -1 | tr -s " " | cut -d" " -f 5,9 | sort –nr <Enter>

The output of Is -I and who are in a tabular form, where the columns are separated by one or more spaces. To provide these output to cut as input, the extra spaces need to be squeezed to a single space.

.To produce only the login names and time of login:

$ who I tr -s " " I cut -d" " -f1,5 '<Enter>

.Display the frequencies of different words present in the input:

$ cat poem I tr -s " " I tr "" "\012" I sort I uniq -c

Here the cat command will produce the input for the first tr command, where all the consecutive spaces are squeezed into a single space. The second tr will translate each space into a new line character, thus producing one word per line.

sort command will sort the result and pass it to the uniq command, which will count the number of occurrences of each word and prints the number along with the"word.

.To find the largest file from the current directory:

$ Is -1 I tr -s " " I cut -d" " -f 5, 9 I sort -nr I head -1 <Enter>

here, the output of Is -I will be passed to tr, which squeezes the extra spaces, then the cut will cut the field 5 and 9 that is the size and filename and passes them to sort.

sort will sort the input in reverse order of the numeric field arid the output is passed on to the head, which will display the first line of the input. The output is then the filename and the size of the largest file present in the directory.

Question:
Write the command to change the grade 'E6' to 'M1' for the employee 'VIJAY', using the employee file.