**Implement all the following Commands.**

**Unix Filter Commands**

* **grep:** Find lines in stdin that match a pattern and print them to stdout.
* **sort:** Sort the lines in stdin, and print the result to stdout.
* **uniq:** Read from stdin and print unique (that are different from the adjacent line) to stdout.
* **cat:** Read lines from stdin (and more files), and concatenate them to stdout.
* **more:** Read lines from stdin, and provide a paginated view to stdout.
* **cut:** Cut specified byte, character or field from each line of stdin and print to stdout.
* **paste:** Read lines from stdin (and more files), and paste them together line-by-line to stdout.
* **head:** Read the first few lines from stdin (and more files) and print them to stdout.
* **tail:** Read the last few lines from stdin (and more files) and print them to stdout.
* **wc:** Read from stdin, and print the number of newlines, words, and bytes to stdout.
* **tr:** Translate or delete characters read from stdin and print to stdout.

| **Command** | **grep - It is a command for pattern searching in a file and prints those lines containing that specified pattern. If the file name is not mentioned, grep searches in stdin.** |
| --- | --- |
| Common Syntax | $ grep [option] pattern [filename …] |
| Example | $ grep ‘[A-M]’ file1 Prints those lines which contains capital letters in the range of A to M |

| **Command** | **wc - It is a command to count the number of lines, words and characters in a file** |
| --- | --- |
| Common Syntax | $ wc [OPTION] ….[FILE] |
| Example | $ cat file1 Hello How do you do $ wc file1 2 5 20 file1  No of lines-2 No of words-5 No of characters(bytes)-20 |

### Pipes in Unix

A series of filter commands can be piped together using the pipe symbol: ‘|’. When two commands are piped together, the stdin of the second program is read from the stdout of the first program. This creates a powerful mechanism for running complex commands quickly.

Pipes are used to direct the ‘stdout’ stream of one command to the ‘stdin’ stream of the next command.

Let us now see an **Example** of using pipes to print out a sorted list of unique words. If file1 has a list of words in a random order with random repetitions, then the following piping can be used to achieve this.

$ sort file1 | uniq > file2

Here, the sort command reads input from the file ‘file1’ and sends the output to stdout. The pipe symbol causes the output of the sort command to be redirected to the input of the uniq command. The uniq commands reads the sorted list from its stdin and prints the unique words from there to its stdout.

Finally, the output redirection symbol ‘>’ redirects the stdout of the uniq command to the file ‘file2’.

Unix sed and awk Text Processing Utilities

Unix provides sed and awk as two text processing utilities that work on a line-by-line basis. The sed program (stream editor) works well with character-based processing, and the awk program (Aho, Weinberger, Kernighan) works well with delimited field processing.

Both use regular expressions to find patterns and support commands to process the matches.

| **Command** | **awk – this command is a useful and powerful command used for pattern matching as well as for text processing.** |
| --- | --- |
| Common Syntax | awk [options] ‘program text’ file |
| Example | $ls -l | awk ‘{print $3}’ This command will display only the third column from the long listing of files and directories. |

| **Command** | **sed – this is a powerful command for editing a ‘stream’ of text. It can read input from a text file or from piped input, and process the input in one pass..** |
| --- | --- |
| Common Syntax | sed[OPTION]…..[-f][file] |
| Example1 | sed -n ‘/hello/p’ file1  This command will display all the lines which contains hello |
| Example2 | sed ‘s/hello/HELLO/’ file1 This command will substitute hello with HELLO everywhere in the file. |
| Example3 | sed ‘/hello/,+2d’ file1 This command will delete the two lines starting with the first match of ‘hello’ |

.............................................................................................................................................................

**WORD Command :**

To list the content count of no of lines , words, characters .

**Syntax :**

$wc<filename>

**Options :**

-c – to display no of characters.

-l – to display only the lines.

-w – to display the no of words.

**FILTERS AND PIPES**

**HEAD :** It is used to display the top ten lines of file.

**Syntax:** $head<filename>

**TAIL :** This command is used to display the last ten lines of file.

**Syntax:** $tail<filename>

**GREP :**This command is used to search and print the specified patterns from the file.

Sy**ntax:** $grep [option] pattern <filename>

**SORT :** This command is used to sort the datas in some order.

**Syntax:** $sort<filename>

**PIPE :** It is a mechanism by which the output of one command can be channeled into the input of another command.

**Syntax:** $who | wc -l

**TR :**The tr filter is used to translate one set of characters from the standard inputs to another.

**Syntax:** $tr “[a-z]” “[A-Z]”

