PATTERN MATCHING , WILD CARD , META & CHARACTER CLASS( )

Kavyashree . N
MCA 'A' section

PATTERN MATCHING

Pattern matching in the shell against filenames has metacharacters defined differently from the rest of the unix pattern ~~patt~~ matching programs. * is match any character except whitespace, ? is match one character except whitespace. so *.c is match any filename ending with the two characters .c.

Wild card pattern

Given a text and a wildcard pattern, implement wildcard pattern matching algorithm that finds if wildcard pattern is matched with text. The matching should cover the entire text. The wildcard pattern can include the characters '?' and '*'

'?' - matches any single character

'*' - Matches any sequence of characters

eg: Text = " baabab "

Pattern = " * * * ba * * * ab ", Output - True

Pattern = " baaa?ab ", Output - True

Pattern = " ba*a? ", Output - True

Pattern = " a* ab ", Output - False

Metacharacters

The command option, option arguments and command arguments are separated by the space character. However, we can also use special characters called metacharacters in linux command that the shell interprets rather than passing to the command

eg. `>` - Output redirection

`<` - Input redirection

`*` - File substitution wildcard; zero or more characters

`?` - File substitution wildcard; one character

`|` - Pipe (|)

`>>` - Output redirection to append

`<<` - input redirection

> - Syntax

```
echo "hello"
>test.txt
```

>> syntax

```
echo 'file welcom' >> test.txt
```

This will append the line to the existing file

* Syntax

```
ls m*
-> mod1  mod2
```

? Syntax

```
ls ?.txt
```

eg 1.txt 11.txt

Output 1.txt

FILTERS

Filters are programs that take plain text [either sorted in a file or produced by another program] as standard input, transforms it into a meaningful format, and then returns it as standard output.

1. cat : Displays the text of the file line by line

   syntax : cat [path]

2. Head : Displays the first n lines of the specified text files. If the number of lines is not specified then by default prints first 10 lines

   syntax : head [-number_of_lines_to_print] [path]

3. Tail : It works the same way as head, just in reverse order. The only difference in tail is, it returns the lines from bottom to up.

   syntax : tail [-number_of_lines_to_print] [path]

4. wc : wc command gives the number of lines, words and characters in the data

   syntax : wc [-options] [path]

   This gives 4 outputs as
   - → Number of lines
   - → Number of words
   - → Number of characters
   - → path

5 Grep : Grep is used to search a particular information from a text file.

   syntax : grep [options] pattern [path]

6. nl : nl is used to number the lines of our text data

syntax : nl [-options] [path]

7. sed : sed stands for stream editor. It allow us to apply search and replace operations on our ~~text~~ data effectively. sed is quite an advanced filter and all its options can be seen on its man page

syntax : sed [path]