Amrita School of Engineering, chennai Campus.

19CSE101: Computer Systems Essentials

**LAB SHEET 3- Operating System**

**Shell Programming using Regular Expressions**

Section 8: Regular Expressions

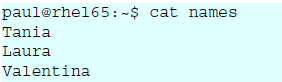
Regular expressions are a very powerful tool in Linux. They can be used with a variety of programs like bash, vi, rename, grep, sed, and more.

This Section introduces you to the basics of regular expressions.

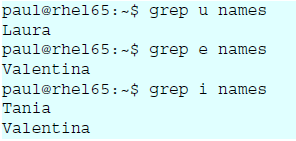
grep

print lines matching a pattern

grep is a popular Linux tool to search for lines that match a certain pattern. Below are some examples of the simplest regular expressions. This is the contents of the test file. This file contains three lines (or three newline characters).

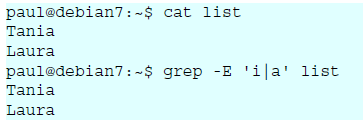


When grepping for a single character, only the lines containing that character are returned.



**one or the other**

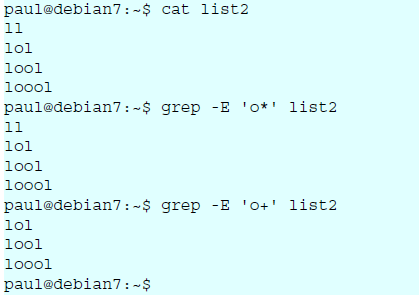
PRCE and ERE both use the pipe symbol to signify OR. In this example we grep for lines containing the letter i or the letter a.



**Cat**

**one or more**

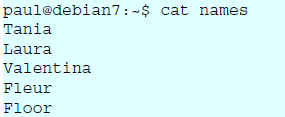
The \* signifies zero, one or more occurences of the previous and the + signifies one or more of the previous.



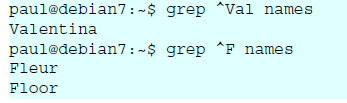
grep

**match the start of a string**

For the following examples, we will use this file.

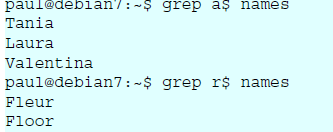


The caret character (^) will match a string at the start (or the beginning) of a line. Given the same file as above, here are two examples.



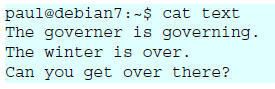
**match the end of a string**

The two examples below show how to use the dollar character to match the end of a string.

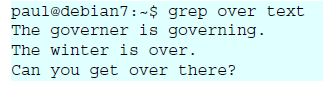


**separating words**

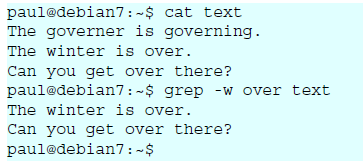
Regular expressions use a -w option to reference a word separator. Take for example this file:



Simply grepping for over will give too many results.



Surrounding the searched word with spaces is not a good solution (because other characters can be word separators). This screenshot below show how to use -w to find only the searched word:



**Lab Exercise**

1. Write shell scripts for the following actions in the given order:
   1. Create a file **Studdetails** with the following contents
   2. Filter the details of the student Bob
   3. Find the number of students with their names containing the letter a, e or i
   4. Find the marks of students whose names starts with ‘b’ (case insensitive)
   5. Find the names of students whose names starts with ‘b’ and ends with ‘y’ (case insensitive)
2. Write a shell script to perform the following actions in the given order:
   1. Create a file **numericdata** with the following contents

Karunagappally 34567 7864 6785

Kollam 56754 6754 7654

Vallikkavu 54328 7548 45675

Trivandrum 16423 6654 6754

Ernakulam 28796 8549 9875

Kayamkulam 35589 75892 3451

kottayam 45557 6773 6547

tirukulum 45675 56476 7896

(Hint : First field is referred as Place second as code1 third as code2 and fourth as code3)

* 1. Display the details of Places that starts with ‘T’(case sensitive)
  2. Display code3 in sorted order(ascending) of the places that start with ‘K’(case insensitive) and ends with am
  3. Filter code2 that starts with 6 and ends with 4
  4. Filter code2 having one or more occurrence of the digit 6.
  5. Filter all code1 having one or more occurrence of the digit 5.

1. Write a shell script to perform the following actions in the given order:
   1. Write the contents of ls -la to a file named **HiddenDetails**
   2. Write the contents of file **HiddenDetails** to the file **HiddenFiles** after replacing multiple occurrences of space with a single space.
   3. Using the file **HiddenFiles,** find the total number of files and folders that starts with ‘p’(case insensitive)
   4. Using the file **HiddenFiles,** find the names of files and folders that starts with ‘.’ and ends with ‘nda’
   5. Using the file **HiddenFiles,** list all the directories

(The first field in ls -la denotes the permissions associated with the file. If the first letter in permission is ‘d’ then it is a directory)

1. Write shell program to perform the following actions:
   1. Write the contents of the command man sort into a file **SortManual**.
   2. Filter all lines form the file **SortManual** that starts with ‘compare’.
   3. Filter all lines form the file **SortManual** that ends with ‘sort’.
   4. Display the number of lines in **SortManual** that has the word ‘numeric’.
   5. Write the contents of file **SortManual** to the file **sort** after replacing multiple occurrences of ‘-’ with a single ‘-’

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*