

EXPERIMENT NO 6

Aim: Execution of scripts using grep/sed/awk/perl.

LO NO & LO Statements:

LO 2-Identify the unix general purpose commands

LO 3-Apply unix commands for system administrative tasks such as file system management and user management.

LO 6-Implement advanced scripts using awk & perl languages and grep, sed, etc commands for performing various tasks.

Theory:

Scripts

Grep command

Grep (global regular expression print) command is the most powerful and regularly used Linux command-line utility. Using Grep, you can search for useful information by specifying a search criteria. It searches for a particular expression pattern in a specified file. When it finds a match, it prints all the lines of a file that matched the specified pattern. It comes in handy when you have to filter through large log files.

In this article, we will explain the use of grep utility with different examples. We will use Debian 10 for explaining the commands and methods mentioned in this article.

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).

Syntax:

grep [options] pattern [files]

Sed command

SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).

SED command in unix supports regular expression which allows it perform complex pattern matching.

Syntax:

sed OPTIONS... [SCRIPT] [INPUTFILE...]

Awk command

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

Awk is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a general-purpose scripting language designed for advanced text processing. It is mostly used as a reporting and analysis tool.

Unlike most other programming languages that are procedural, awk is data-driven, which means that you define a set of actions to be performed against the input text. It takes the input data, transforms it, and sends the result to standard output.

This article covers the essentials of the awk programming language. Knowing the basics of awk will significantly improve your ability to manipulate text files on the command line.

Syntax

awk options 'selection _criteria { action }' input-file > output-file

Perl command

The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). It combines some of the best features of sed, awk, and sh, making it familiar and easy to use for Unix users to whip up quick solutions to annoying problems. Its general-purpose programming facilities support procedural, functional, and object-oriented programming paradigms, making Perl a comfortable language for major projects.

To create a Perl script, use a text editor to enter Perl commands, save the file with a .pl extension (such as sample.pl), and then use chmod to mark the file as executable. The extension is not required, but it's a common Unix convention and will help you identify your Perl source files without looking inside.

Code and Output**a) Execute the following scripts using grep / sed commands:**

i) Write a script using the grep command to find the number of characters, words and lines in a file.

```
s1.sh x
#!/bin/bash
echo Enter file name:
read file
if [ ! -f $file ]
then
echo -e "file does not exist"
exit
else
echo File exist
fi
l=$file
echo -e "Number of lines is:"
grep -c "." $l
echo "Number of character is:"
grep -o [a-z] $l | wc -l
echo "Number of words is:"
wc -w $l
|
```

```
[root@localhost ~]# gedit s1.sh
[root@localhost ~]# chmod +x s1.sh
[root@localhost ~]# ./s1.sh
Enter file name:
mishul.pl
File exist
Number of lines is:
16
Number of character is:
67
Number of words is:
37 mishul.pl
[root@localhost ~]#
```

ii) Write a script using sed command to replace all occurrences of a particular word in a given file.

```
[root@localhost ~]# cat exp6.txt
hello this is ishika
This is linux lab
[root@localhost ~]# sed 's/linux/UNIX/g' exp6.txt
hello this is ishika
[root@localhost ~]#
```

b) Execute the following scripts using awk / perl languages:

(i) Write an awk script to print all even numbers in a given range.

```
[root@localhost ~]# cat ish1.awk
BEGIN {
print "Even numbers from 0 to 25:";
for (i=0;i<=25;i++)
{
if (i%2 == 0){
print i
;
}
}
}

[root@localhost ~]# awk -f ish1.awk
Even numbers from 0 to 25:
0
2
4
6
8
10
12
14
16
18
20
22
24
[root@localhost ~]#
```

(ii) Write a perl script to check if a number is prime or not.

Code:

```
#!/usr/bin/perl
print "Enter a number n:";
$n=<
>;
$d=0;
if($n
==2)
{
print "Prime number\n";
}
else {
for($c=2;$c<=$n-1;$c++)
{
if($n%$c==0{
$d=1;
break;
}
}
if($d=
=1) {
print "Not a prime number\n";
}
else{
print "Prime number\n";
}
}
```

Output:

```
[root@localhost ~]# gedit prime.pl
[root@localhost ~]# chmod +x prime.pl
[root@localhost ~]# perl prime.pl
Enter a number n:52
Not a prime number
[root@localhost ~]#
```

Conclusion: In this experiment, we had implemented grep, sed commands as well as awk perl scripting in the unix . and LO2, LO3, and LO6 were achieved successfully.

NAME-SOHAM DESAI

ROLL NO-11

CLASS- SEIT

DATE-23/03/2022