ISSN:0975-9646

Utility Based Pattern Matching Approach for Data Mining

Kanchi. Suresh[1] Dr. Hari Krishna Pulagam[2]

[1]*Associate Professor, Dept. of Computer Science and Engineering*
Guru Nanak Institute of Technology, Hyderabad, AP, INDIA

[2]*Professor, Dept. of Computer Science and Engineering*
Sreyas Institute of Engg &Technology, Nagole, Hyderabad, AP, India.

Abstract— **Pattern matching is one of the methods for classification of data, it is used to classify data into predefined groups or classes. In this paper, we proposed utilities made available in Linux to make use in pattern matching. With this approach, the grep family utilities are proposed to apply on data warehouse, and to warehouse the result into a temporary file. This intermediate or temporary warehouse can be used to mine the knowledge and hence to practice decision.**

Keywords: data mining, pattern, utilities, warehouse, grep family, classes, group.

1. Introduction

The data mining involves variety of techniques to deduce a valid and useful hidden information by means of understandable correlations and patterns from large amount of data called data warehouse. Finding of needful patterns from data or warehouse has different conventions like data pattern processing, knowledge extraction, information extraction, knowledge discovery and information harvesting. Data mining is a well familiar among community of database researchers top level business and statistics personnel. Preparing data ready for mining involve many preprocessing steps referred to Knowledge Discovery in Databases (KDD). In brief, the KDD process comprises data preparation, data selection, data clean-up and appropriate elucidation for the consequences from the data mining process ensuring that the useful knowledge is derived from the data. This paper presents pattern matching utilities of grep(globally search for regular expression) family available in Linux.

1.1 PROBLEM DEFINITION

Take the a text file which contains some data of students pertaining to a regular examination results. Apply the preprocessor and get it ready for the script which is going to be developed. The Pattern matching is done by preparing script using utilities available with unix such as grep and its familly. It is a kind of methods for classification of data, it is used to classify data into predefined groups or classes. With this approach, the grep family utilities are proposed to apply on data warehouse, and to warehouse the result into a temporary file. This intermediate or temporary warehouse can used to mine the knowledge and hence to formulate decisions.

2.0 Literature Survey

Data warehousing is a construction which involves collection of data from different databases, data cleaning and data integration, and it is would be the consequence of important pre-processing step for data mining. Building such a large data warehouse that consolidates data from multiple sources may be databases, resolves data integrity issues, and gathers the data into a database, can be a huge task, may take years and costing millions of dollars.

- ***Knowledge Discovery in Databases (KDD):*** process of finding useful information and patterns in data.
- ***Data Mining:*** Use of algorithms to extract the information and patterns derived by the KDD process.

2.1 FILTERS AND UTILITIES

Such patterns have been recognized by utilities provided as a part of Linux Operating System. They are grep(**g**lobally search for **r**egular **e**x**p**ression) family,

- grep regular or normal grep.
- egrep extended grep.
- fgrep fast grep.
- cut fields and characters extractor.

where regular expression is notation to express a well formed formula in precise involving predefined operators. Apart from these there are advanced filters those can also be used for filtering the data according to requirement, such filters and pattern matching utilities are awk, gawk, and sed etc.

This is as shown in Figure-1

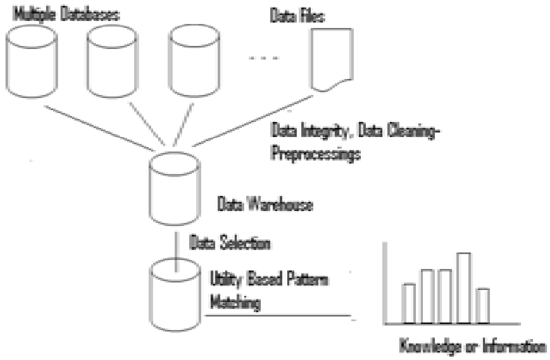


Figure - 1

2.2 SHELL SCRIPTS

In this paper, bash shell scripts are written with necessary sequence of shell commands, meant for accomplish the proposed and stated task called utility base pattern matching for data mining. The shell scripting is used to make utility and such utilities perform required data mining. For a sake of understanding a sample script which fetches required lines those are between given range. A shell program name takes the general from as “scriptname.sh” simply “scriptname” and the same can be executed at prompt as “sh scriptname.sh” or without extension “.sh”. A sample script and its execution is-

```
if [ $# -eq 0 ]
then
     echo "No arguments are provided"
     exit
fi
for arg in $*
do
  if [ -f $arg ]
  then
     n=`wc -l $arg`
     echo "File: $n"
  elif [ -d $arg ]
  then
     echo "Directory: $arg"
  fi
done
```

```
suresh@suresh:~$ sh file-dir.sh file-dir.sh

File: 18 file-dir.sh
```

the script “file-dir.sh” prints number of lines present in in a file given file which is been supplied as a command line argument

```
suresh@suresh:~$ regr -a 09t.txt
```

upon successful execution, subject wise result of the candidates present in “09r.txt” should be displayed immediately.

3. Design And Implementation

A command for doing more the one task can be prepared with multiple options supplied as command line argument where each argument is meant for performing a different task. In this, the command “regr” used as a short name for “regular results” is been prepared for with four option as, “regr [-a -f -s -p] filename” to display result with percentage.

where “filename” is one which contains regular result of the candidates.

-a: result of all the candidates subject wise.
-f: result of all the fail candidates subject wise.
-s: subject wise result of all the students.
-p: result of all the pass candidates subject wise.

The command “regr” is been prepared to execute at command prompt with the four options as mentioned,

```
suresh@suresh:~$ regr -a 09t.txt
```

3.1 IMPLEMENTATION

List of file are going to be created after successful execution of the command mentioned in the last sction. Those files are,

```
09r.txt    f3, getopts.sh,   regr.c, spl.sh,   stwresult.sh, f4
getopts.sh, result.txt, stwfails.sh, swfail.sh, bsr.sh, f5
mytest.sh   samp.sh stwfails.sh    swfail.sh    f1       f6
mytest.sh   samp.sh~ stwpassresult.txt  f10    f7  nf.sh
split.sh  stwpass.sh
f11      f8  nf.sh~        split.sh~  stwpass.sh  f2  f9  regr
spl.sh      stwresult.sh.
```

For example, the regr is the command which coded as shown in the following file name “regr.c”,

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
main(int argc, char *argv[])
{
int i, len;
char *cmd="",*x="";
argv[0]="sh bsr.sh";
len=strlen(*argv);
cmd=(char*)malloc(len*sizeof(char));
for(i=0; i<argc; i++)
{
strcat(cmd, argv[i]);
if(argv[i+1]!=NULL)
{
x=(char*)malloc(sizeof(char));
strcat(x, " ");
strcat(cmd, x);
}
}
system(cmd);
}
```

In the above code, a set of needful functions have been used to prepare the command “regr”. The list of all files mentioned in this section completes the command with mentioned four opti-ons which are discussed in brief in the previous section. The following is the another important script used to perform the task of the command.

```
option="$1"
splitfiles()
{
sh split.sh
}
studentwiseresult()
{
sh stwresult.sh
}
studentwisefails()
{
sh stwfails.sh
}
studentwisepass()
{
sh stwpass.sh
}
subjectwisepass()
{
sh swfail.sh
}
if [ $# -eq 0 ]
then
splitfiles
studentwiseresult
exit 1
fi
case $option in
-a)
tput bold
tput smul
echo "LIST OF ALL STUDENTS-SUBJECT WISE PASS PERCENTAGES"
splitfiles
studentwiseresult
tput sgr0
;;
-f) echo "LIST OF FAIL STUDENTS"
splitfiles
studentwisefails
;;
-p) echo "LIST OF PASS STUDENTS"
splitfiles
studentwisepass
;;
-s) echo "LIST OF SUBJECT WISE PASS PERCENTAGES"
splitfiles
subjectwisepass
;;
*) echo "`basename $0`:usage: [-a -f -p -s] filename"
exit 1
;;
esac
```

3.2 TESTING METHODS

We implement various tests in order to check and rectify bugs that occurred in the command preparation, it produces the outputs as per opinion.

i) Unit Testing

Unit testing has been used to on various files to create and execute those files in the order to process the command with a designated option.

ii) Integration Testing

This has been implemented to check the possible errors those occur in normal cases of integrating the various modules together into a system to function.

Networks related issues were too handled in the process of integration testing.

iii) White Box Testing

This has been implemented to check the possible errors that occur while developing the individual scripting as well as C- programs and they are attended to further implementation of each script. If any line of control is not in execution in case then such are attended and rectified.

iv) Black Box Testing

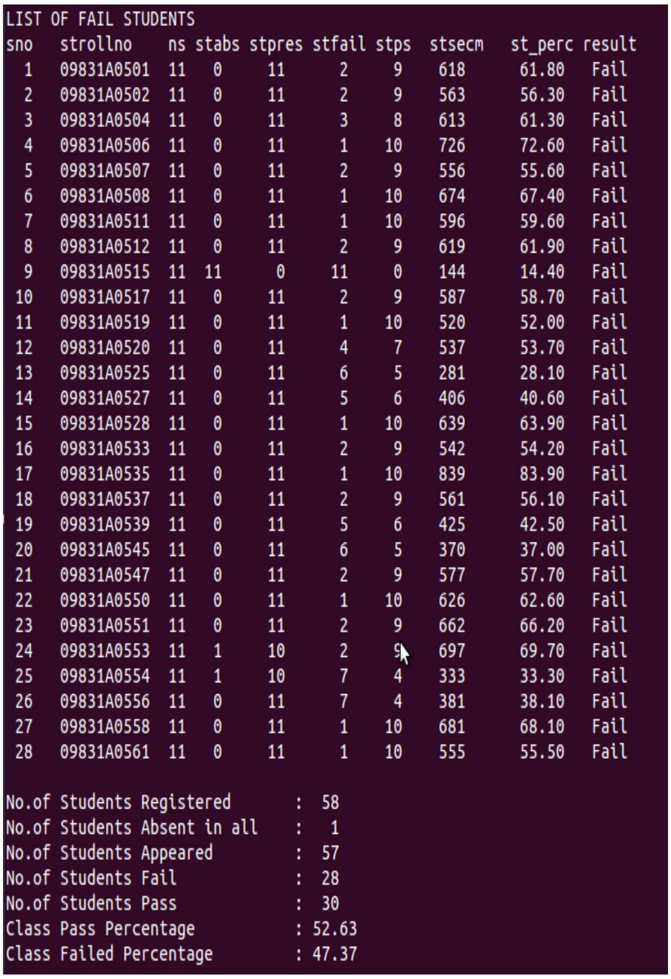
This has been implemented to check the possible errors by executing the command together with required files and scripts to produce desired result in expected format. The command is checked for all four options which are specified in the section called design and implementation.

4. RESULTS

The following the output screen is obtained to show summary and display all the failed students, the way of execution of command is,

suresh@suresh:~$ regr -f 09t.txt

this shows many details like registration number, number of appeared subjects, absents, failed, total percentage and result.

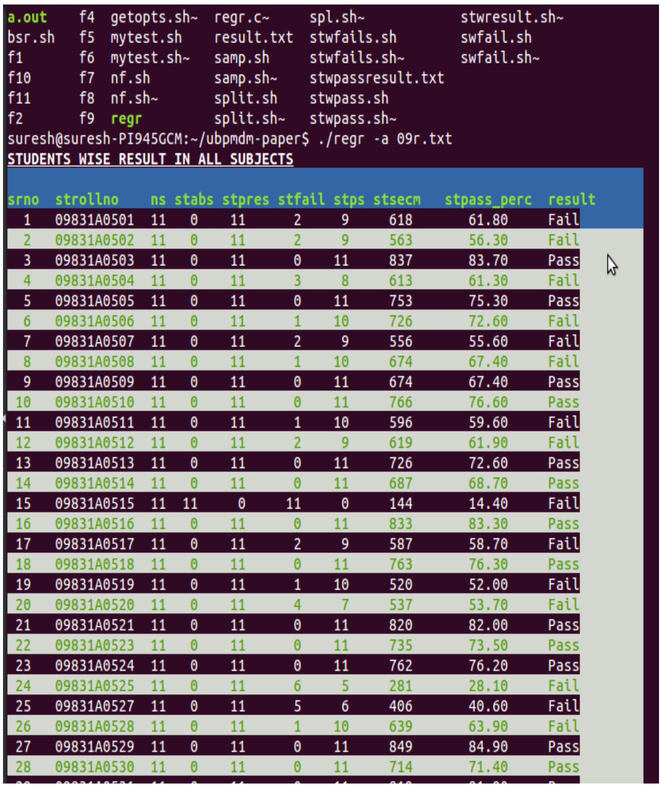


```
LIST OF FAIL STUDENTS
sno   strollno    ns stabs stpres stfail stps  stsecm   st_perc result
  1   09831A0501  11   0     11      2     9    618      61.80   Fail
  2   09831A0502  11   0     11      2     9    563      56.30   Fail
  3   09831A0504  11   0     11      3     8    613      61.30   Fail
  4   09831A0506  11   0     11      1    10    726      72.60   Fail
  5   09831A0507  11   0     11      2     9    556      55.60   Fail
  6   09831A0508  11   0     11      1    10    674      67.40   Fail
  7   09831A0511  11   0     11      1    10    596      59.60   Fail
  8   09831A0512  11   0     11      2     9    619      61.90   Fail
  9   09831A0515  11  11      0     11     0    144      14.40   Fail
 10   09831A0517  11   0     11      2     9    587      58.70   Fail
 11   09831A0519  11   0     11      1    10    520      52.00   Fail
 12   09831A0520  11   0     11      4     7    537      53.70   Fail
 13   09831A0525  11   0     11      6     5    281      28.10   Fail
 14   09831A0527  11   0     11      5     6    406      40.60   Fail
 15   09831A0528  11   0     11      1    10    639      63.90   Fail
 16   09831A0533  11   0     11      2     9    542      54.20   Fail
 17   09831A0535  11   0     11      1    10    839      83.90   Fail
 18   09831A0537  11   0     11      2     9    561      56.10   Fail
 19   09831A0539  11   0     11      5     6    425      42.50   Fail
 20   09831A0545  11   0     11      6     5    370      37.00   Fail
 21   09831A0547  11   0     11      2     9    577      57.70   Fail
 22   09831A0550  11   0     11      1    10    626      62.60   Fail
 23   09831A0551  11   0     11      2     9    662      66.20   Fail
 24   09831A0553  11   1     10      2     9    697      69.70   Fail
 25   09831A0554  11   1     10      7     4    333      33.30   Fail
 26   09831A0556  11   0     11      7     4    381      38.10   Fail
 27   09831A0558  11   0     11      1    10    681      68.10   Fail
 28   09831A0561  11   0     11      1    10    555      55.50   Fail

No.of Students Registered      :  58
No.of Students Absent in all   :   1
No.of Students Appeared        :  57
No.of Students Fail            :  28
No.of Students Pass            :  30
Class Pass Percentage          : 52.63
Class Failed Percentage        : 47.37
```

Fig.2: List of failed students with details

The following the output screen is obtained to show summary and display all the students results like pass and fail, the way of execution of command is, suresh@suresh:~$ regr -a 09t.txt,

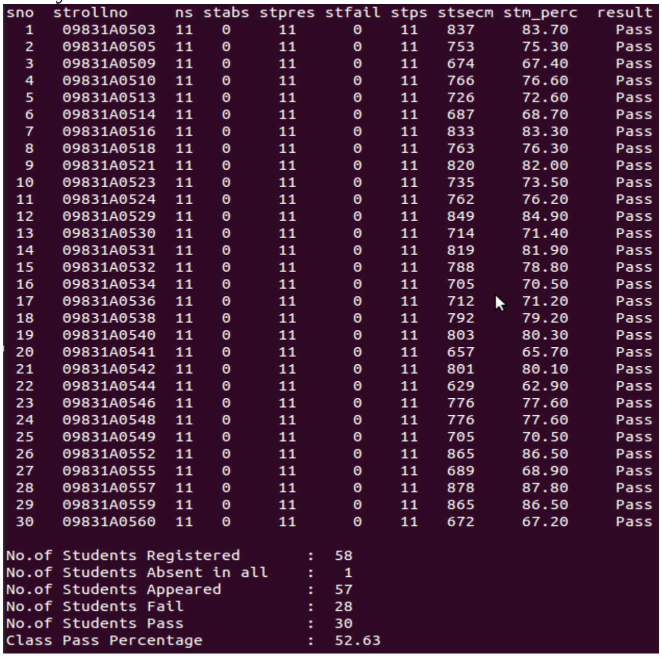
this shows many details like registration number, number of appeared subjects, absents, pass, failed, total percentage and finally result.



```
a.out    f4  getopts.sh~  regr.c~    spl.sh~            stwresult.sh~
bsr.sh   f5  mytest.sh    result.txt stwfails.sh        swfail.sh
f1       f6  mytest.sh~   samp.sh    stwfails.sh~       swfail.sh~
f10      f7  nf.sh        samp.sh~   stwpassresult.txt
f11      f8  nf.sh~       split.sh   stwpass.sh
f2       f9  regr         split.sh~  stwpass.sh~
suresh@suresh-PI945GCM:~/ubpmdm-paper$ ./regr -a 09r.txt
STUDENTS WISE RESULT IN ALL SUBJECTS

srno  strollno    ns stabs stpres stfail stps stsecm   stpass_perc  result
  1   09831A0501  11   0     11      2     9    618       61.80     Fail
  2   09831A0502  11   0     11      2     9    563       56.30     Fail
  3   09831A0503  11   0     11      0    11    837       83.70     Pass
  4   09831A0504  11   0     11      3     8    613       61.30     Fail
  5   09831A0505  11   0     11      0    11    753       75.30     Pass
  6   09831A0506  11   0     11      1    10    726       72.60     Fail
  7   09831A0507  11   0     11      2     9    556       55.60     Fail
  8   09831A0508  11   0     11      1    10    674       67.40     Fail
  9   09831A0509  11   0     11      0    11    674       67.40     Pass
 10   09831A0510  11   0     11      0    11    766       76.60     Pass
 11   09831A0511  11   0     11      1    10    596       59.60     Fail
 12   09831A0512  11   0     11      2     9    619       61.90     Fail
 13   09831A0513  11   0     11      0    11    726       72.60     Pass
 14   09831A0514  11   0     11      0    11    687       68.70     Pass
 15   09831A0515  11  11      0     11     0    144       14.40     Fail
 16   09831A0516  11   0     11      0    11    833       83.30     Pass
 17   09831A0517  11   0     11      2     9    587       58.70     Fail
 18   09831A0518  11   0     11      0    11    763       76.30     Pass
 19   09831A0519  11   0     11      1    10    520       52.00     Fail
 20   09831A0520  11   0     11      4     7    537       53.70     Fail
 21   09831A0521  11   0     11      0    11    820       82.00     Pass
 22   09831A0523  11   0     11      0    11    735       73.50     Pass
 23   09831A0524  11   0     11      0    11    762       76.20     Pass
 24   09831A0525  11   0     11      6     5    281       28.10     Fail
 25   09831A0527  11   0     11      5     6    406       40.60     Fail
 26   09831A0528  11   0     11      1    10    639       63.90     Fail
 27   09831A0529  11   0     11      0    11    849       84.90     Pass
 28   09831A0530  11   0     11      0    11    714       71.40     Pass
```

Fig.3: Result of all the students with details.

The following the output screen is obtained to show summary and display all the students who pass, the way of execution of command is,
suresh@suresh:~$ regr -p 09t.txt
this shows many details like registration number, number of appeared subjects, absents, pass, failed, total percentage and finally result.



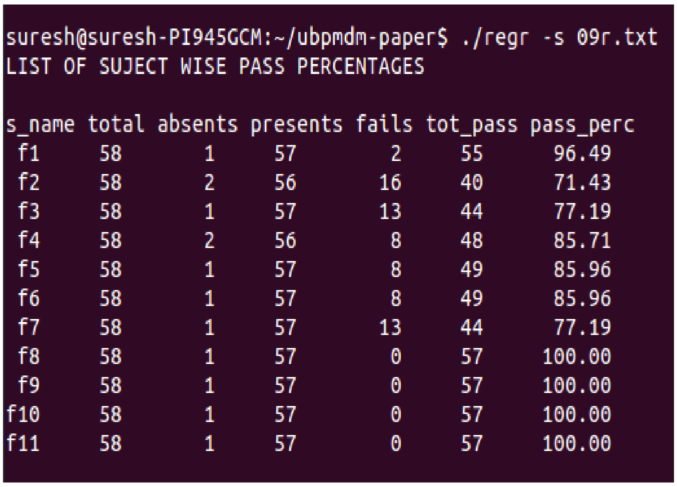
```
sno  strollno     ns  stabs  stpres  stfail  stps  stsecm  stm_perc   result
  1   09831A0503  11   0      11       0      11    837     83.70      Pass
  2   09831A0505  11   0      11       0      11    753     75.30      Pass
  3   09831A0509  11   0      11       0      11    674     67.40      Pass
  4   09831A0510  11   0      11       0      11    766     76.60      Pass
  5   09831A0513  11   0      11       0      11    726     72.60      Pass
  6   09831A0514  11   0      11       0      11    687     68.70      Pass
  7   09831A0516  11   0      11       0      11    833     83.30      Pass
  8   09831A0518  11   0      11       0      11    763     76.30      Pass
  9   09831A0521  11   0      11       0      11    820     82.00      Pass
 10   09831A0523  11   0      11       0      11    735     73.50      Pass
 11   09831A0524  11   0      11       0      11    762     76.20      Pass
 12   09831A0529  11   0      11       0      11    849     84.90      Pass
 13   09831A0530  11   0      11       0      11    714     71.40      Pass
 14   09831A0531  11   0      11       0      11    819     81.90      Pass
 15   09831A0532  11   0      11       0      11    788     78.80      Pass
 16   09831A0534  11   0      11       0      11    705     70.50      Pass
 17   09831A0536  11   0      11       0      11    712     71.20      Pass
 18   09831A0538  11   0      11       0      11    792     79.20      Pass
 19   09831A0540  11   0      11       0      11    803     80.30      Pass
 20   09831A0541  11   0      11       0      11    657     65.70      Pass
 21   09831A0542  11   0      11       0      11    801     80.10      Pass
 22   09831A0544  11   0      11       0      11    629     62.90      Pass
 23   09831A0546  11   0      11       0      11    776     77.60      Pass
 24   09831A0548  11   0      11       0      11    776     77.60      Pass
 25   09831A0549  11   0      11       0      11    705     70.50      Pass
 26   09831A0552  11   0      11       0      11    865     86.50      Pass
 27   09831A0555  11   0      11       0      11    689     68.90      Pass
 28   09831A0557  11   0      11       0      11    878     87.80      Pass
 29   09831A0559  11   0      11       0      11    865     86.50      Pass
 30   09831A0560  11   0      11       0      11    672     67.20      Pass

No.of Students Registered       :  58
No.of Students Absent in all    :   1
No.of Students Appeared         :  57
No.of Students Fail             :  28
No.of Students Pass             :  30
Class Pass Percentage           :  52.63
```

Fig.4: List of students whose result is pass

The following the output screen is obtained to show subject wise summary with pass percentage, the way of execution of command is,

suresh@suresh:~$ regr -s 09t.txt

this shows many details like subject anme, number of appeared students, absents, pass, failed, total pass percentage .

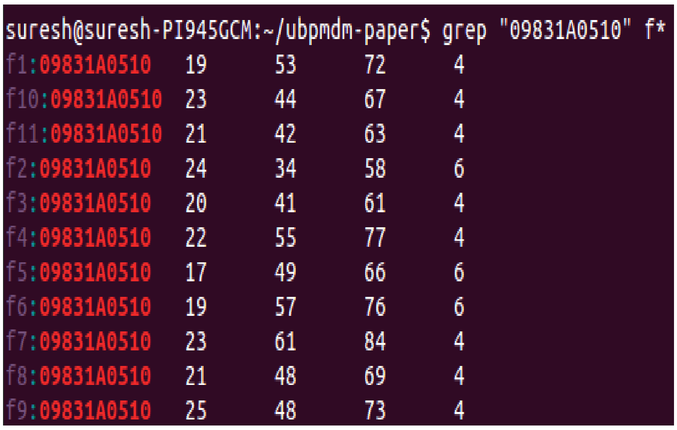


```
suresh@suresh-PI945GCM:~/ubpmdm-paper$ ./regr -s 09r.txt
LIST OF SUJECT WISE PASS PERCENTAGES

s_name total absents presents fails tot_pass pass_perc
 f1     58       1       57       2     55      96.49
 f2     58       2       56      16     40      71.43
 f3     58       1       57      13     44      77.19
 f4     58       2       56       8     48      85.71
 f5     58       1       57       8     49      85.96
 f6     58       1       57       8     49      85.96
 f7     58       1       57      13     44      77.19
 f8     58       1       57       0     57     100.00
 f9     58       1       57       0     57     100.00
f10     58       1       57       0     57     100.00
f11     58       1       57       0     57     100.00
```

Fig.5: List of students whose result is pass

The following the output screen is obtained to show a particular student result, the way of execution of command is,
suresh@suresh:~$ grep “student num” list-of-subjects

THIS SHOWS MANY DETAILS LIKE ROLL NUMBER, INTERNAL, EXETERNAL, TOTAL AND CREDITS OF A STUDENT.



```
suresh@suresh-PI945GCM:~/ubpmdm-paper$ grep "09831A0510" f*
f1:09831A0510   19    53    72    4
f10:09831A0510  23    44    67    4
f11:09831A0510  21    42    63    4
f2:09831A0510   24    34    58    6
f3:09831A0510   20    41    61    4
f4:09831A0510   22    55    77    4
f5:09831A0510   17    49    66    6
f6:09831A0510   19    57    76    6
f7:09831A0510   23    61    84    4
f8:09831A0510   21    48    69    4
f9:09831A0510   25    48    73    4
```

Fig.6: A particular Student result

5. CONCLUSION

This paper attended only a command with four options where each option performs a specific task. This is a one where command prompt is required to execute, it can also be extended to develop a application which presents the result in pleasant and good looking graphical format. This command is executed in environment called multitasking and multi user by nature of the Linux OS.

REFERENCES

[1] Learning Jakarta Struts 1.2: a concise and practical tutorial by Stephan Wiesner

[2] [Foroutan, Jack Sklansky (1987). "Feature Selection for Automatic Classification of Non-Gaussian Data". *IEEE Transactions on Systems, Man andCybernetics* **17** (2):187–198. doi: 10.1109/ TSMC. 1987. 4309029..

[3] http://www.oclug.on.ca Ottawa Canada Linux Users Group.

[4] http://www.exitcertified.com

[5] http://www.fsf.org

[6] http://linux.org.mt/article/terminal “A Beginner's Bash”.

[7] http://www.oreilly.com/catalog/bash2 An excellent book if you want to learn

[8] Conducting Market Research Using Primary Data Kynda R. Curtis, Ph.D. Assistant Professor and State Extension Specialist Department of Resource Economics, University of Nevada, Reno

[9] Lev, L., L. Brewer and G. Stephenson (2004). “Tools for Rapid Market Assessments.” Oregon State University Oregon Small Farms Tech. Report No. 6.

[10] Salant, P. and D.A. Dillman (1994). How to Conduct Your Own Survey. New York: John Wiley and Sons, Inc.

AUTHORS BIOGRAPHY

Suresh. Kanchi[1]

Currently working as an Assoc. Professor in Computer Science and Engineering Department at Guru Nanak Institute of Technology - Hyderabad, A.P, India. He pursued B.Tech(CSE) from VRSECVijayawada, A.P, India, and M.Tech(SE) from School of IT, J.N.T.U.H- Hyderabad. His main interest is around Principles of Compilers, Unix and Shell Scripting, Formal Languages an Automat Theory, Computer Networks, Linux Programming, Data Mining. He has attended National and International Conferences and published papers in International Journals.

Dr. Hari Krishna Pulagam[2]

Currently working as Professor in Dept of Computer Science and Engineering in Sreyas Institute of Engineering Technology - Hyderabad, A.P, India. He pursued M.Tech(CSE) Allahabad Deemed University-U.P, India, and Ph.D(CSE) from University of Allahabad-U.P- His main interest is around Data Structures, Object Oriented Analysis and Design, Web Technologies, Java, Software Engineering, DBMS, Computer Networks, Cloud Computing, Software Testing Methodologies. He has attended National and International Conferences and published papers in International Journals.