

Dept. of CSE, Bennett University
Linux and Shell programming lab
Lab Assignment - 4
AWK Scripting

In this lab we will work on AWK scripting language. AWK is another interpreted programming language which has powerful text processing capabilities. It can solve complex text processing tasks with a few lines of code.

Listed below are just a few of the uses of AWK:

- Text processing,
- Producing formatted text reports,
- Performing arithmetic operations,
- Performing string operations, and many more.

Most of the programs mentioned in the shell scripting section can also be done with awk as it also has similar conditional structures, loops, functions, etc.

Awk has more of a C like syntax. We will focus more on text processing capabilities of awk.

As you are already familiar with structure and basics of awk program and there is no ambiguity in the syntax as it was there with shell scripting, we will start with simple awk programs and move to complex ones towards the end.

Running awk Programs

- If the program is short, then it can be included in the same line on the terminal.
awk 'program' input-file1 input-file2 . . .
- If the program is long, it should be written in a file and saved with .awk extension. Actually, you can just save the file w/o any extension as well. The program is run with

awk -f program-file input-file1 input-file2 . . .

Awk Working Methodology

1. Awk reads the input files one line at a time.
2. For each line, it matches with given pattern in the given order, if matches performs the corresponding action.
3. If no pattern matches, no action will be performed.
4. In the above syntax, either search pattern or action are optional, But not both.
5. If the search pattern is not given, then Awk performs the given actions for each line of the input.
6. If the action is not given, print all that lines that matches with the given patterns which is the default action.
7. Empty braces without any action does nothing. It won't perform default printing operation.
8. Each statement in Actions should be delimited by semicolon.

Consider the following data values:

Name	Gender	Mid_1 (max. 25)	Mid_2 (max 25)	End_sem (max 50)
AX	M	15	20	35
BX	F	25	25	30
CX	F	10	12	20
DX	M	20	20	20
EX	M	15	26	44
EX	M	15	26	44

The above data is stored in marks.txt

Q1: Solve following questions using basic AWK scripts.

- 1.1) Display just the names, genders and endsem marks of: (a) all students (b) the first 3 students only (c) the last student.
- 1.2) Compute the total mark for each student and display it along with their names.
- 1.3) Who is the topper? Which students scored above the class average?
- 1.4) What is the difference between the marks of student who scored highest and lowest total marks?
- 1.5) Store the male students in males.txt and the females in females.txt.

Q2: Advance AWK scripts: You may use if-else to solve these questions:

We need to assign grades to these students. The evaluation criteria is as follows:

Total Marks = 15% of Mid_1 + 35% of Mid_2 + 50% of End_Sem

Grades can be assigned by following reference table:

Total marks range	Grade
100 - 75	A
60 - 74	B
40 - 59	C
Below 40	F

Please use the above reference table to assign grades to each student.

Q3. Create a results.txt file and store the all data of students including grades in descending order of grades.

Q4. Its time to publish the final results, check if there any duplicate entry in the results.txt, if exist then remove it and store the data in final.txt file.

References:

- <https://www.cs.unibo.it/~renzo/doc/awk/nawkA4.pdf>
- <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/>
- <https://likegeeks.com/awk-command/>
- <https://www.tutorialspoint.com/awk/index.htm>

Submission Instruction:

1. Create a log file (word or text) to store every command you run and their respective output. Save this file with your Roll_no@Lab4 name.
2. Use proper comments where required.
3. Submit the assignment on your LMS page.
4. Deadline to submit this is one week from the date of launch.