TERM PAPER

Introduction to UNIX family

CCS35

Topic: AWK

Submitted by:19115045

Kunal Sachdeva

3rd Semester CSE

Submitted to: Dr. Ravi Verma

Submission Deadline: 17/12/2020

Abstract:

The reasons that I chose Awk as my term topic is because:

- It is a scripting language used for manipulating data and generating reports.
- No compiling is required for awk command programming language.
- It allows the user to use numeric functions, string functions, variables, and logical operators.

Awk is a utility that permits a programmer to write tiny but efficient programs in the form of statements that define text patterns that are to be searched for in every line of a document and the action that has to be taken when a match is found within a line.

Introduction:

Awk is used mainly for pattern scanning and processing. It searches one or more files to see if any of them contain lines that matches with the specified pattern and then performs the accompanying actions.

The names of the developers – Aho, Weinberger, and Kernighan are abbreviated to create the name Awk.

WHAT CAN WE DO WITH AWK ?

AWK Operations:

- A line-by-line scanning of the file
- Splitting of each input line into fields
- Compares pattern to input line/fields
- Performs actions on matched lines

Useful For:

- Transform data files
- Produce formatted reports

Programming Constructs:

- Format output lines
- Arithmetic and string operations
- Conditionals and loops

Body:

Types of AWK

Following are the variants of AWK –

- AWK − Original AWK from AT & T Laboratory.
- NAWK − Newer and improved version of AWK from AT& T Laboratory.
- GAWK − It is GNU AWK. All GNU/Linux distributions ship GAWK. It is fully compatible with AWK and NAWK.

AWK SYNTAX:

- awk [options] 'script' file(s)
- awk [options] –f scriptfile file(s)

Options:

-f program-file : Reads the AWK program source from the file program-file, instead of from the first command line argument.

-F fs : Use fs for the input field separator.

AWK is easy to use. We can provide AWK commands either in the form of a text file containing AWK commands or directly from the command line.

AWK Command Line:

We can use an AWK command within single quotes at command line as shown −

```
awk [options] file
```

AWK Program File:

We can provide AWK commands in a script file as shown −

```
awk [options] -f file
```

First, create a text file command.awk containing the AWK command as shown below −

```
{print}
```

Now we can instruct the AWK to read commands from the text file and perform the action.

Built in system variables:

- NR**:** NR command keeps a current count of the number of input records. Remember that records are usually lines. Awk command performs the pattern/action statements once for each record in a file.
- NF: NF command keeps a count of the number of fields within the current input record.
- FS: FS command contains the field separator character which is used to divide fields on the input line. The default is “white space”, meaning space and tab characters. FS can

be reassigned to another character (typically in BEGIN) to change the field separator.

- RS: RS command stores the current record separator character. Since, by default, an input line is the input record, the default record separator character is a newline.
- OFS: OFS command stores the output field separator, which separates the fields when Awk prints them. The default is a blank space. Whenever print has several parameters separated with commas, it will print the value of OFS in between each parameter.
- ORS: ORS command stores the output record separator, which separates the output lines when Awk prints them. The default is a newline character. print automatically outputs the contents of ORS at the end of whatever it is given to print.

The input for your program gets divided into fields and records. It keeps track of number of records that have been read from the current input file so far. This value is stored in a predefined variable called FNR, which is reset to zero every time a new file is started. Another predefined variable, NR, records the total number of input records read so far from all data files. It starts at zero, but is never automatically reset to zero.

Normally, records are separated by newline characters. You can control how records are separated by assigning values to the built-in variable RS. If RS is any single character, that character separates records. Otherwise (in gawk), RS is treated as a regular expression. This mechanism is explained in greater detail shortly.

AWK Scripts are divided into three major parts:

- BEGIN: pre-processing
  - performs processing that must be completed before the file processing starts (i.e., before awk starts reading records from the input file)
  - useful for initialization tasks such as to initialize variables and to create report headings

- BODY: Processing
  - contains main processing logic to be applied to input records
  - like a loop that processes input data one record at a time: if a file contains 100 records, the body will be executed 100times, one for each record

- END: post-processing
  - contains logic to be executed after all input data have been processed
  - logic such as printing report grand total should be performed in this part of the script

Result:

Awk programming plays a key role in UNIX. It is a scripting language used for manipulating data and generating reports.