SOFTENGER
Strive to Excel

Use of Awk

Unix Fundamentals

AWK - Introduction

Awk is a scripting command 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 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 abbreviated from the names of the developers – Aho, Weinberger, and Kernighan.

WHAT CAN WE DO WITH AWK ?

SOFTENGER
Strive to Excel

1. AWK Operations:
   (a) Scans a file line by line
   (b) Splits each input line into fields
   (c) Compares input line/fields to pattern
   (d) Performs action(s) on matched lines
2. Useful For:
   (a) Transform data files
   (b) Produce formatted reports
   (c) Producing formatted text reports,
3. Programming Constructs:
   (a) Format output lines
   (b) Arithmetic and string operations
   (c) Conditionals and loops

Syntax and options

- Syntax:

awk options 'selection _criteria {action }' files

- Sample Commands
  - ❖ Example:
    Consider the following text file as the input file for all cases below.
  - ❖ $ cat > emp.txt

• Ajay	manager	account	45000
• sunil	clerk	account	25000
• varun	manager	sales	50000
• amit	manager	account	47000
• tarun	peon	sales	15000
• deepak	clerk	sales	23000

Default behaviour of AWK

- Default behavior of Awk : By default Awk prints every line of data from the specified file.

$ awk '{print}' emp.txt

Output:

```
ajay    manager    account    45000
sunil   clerk      account    25000
varun   manager    sales      50000
amit    manager    account    47000
tarun   peon       sales      15000
deepak  clerk      sales      23000
sunil   peon       sales      13000
satvik  director   purchase   80000
```

In the above example, no pattern is given. So the actions are applicable to all the lines. Action print without any argument prints the whole line by default, so it prints all the lines of the file without failure.

To Print the lines which matches with the given pattern

.

$ awk '/manager/ {print}' emp.txt Output:

ajay	manager	account	45000
varun	manager	sales	50000
amit	manager	account	47000

In the above example, the awk command prints all the line which matches with the ‘manager’.

Splitting a Line Into Fields

Splitting a Line Into Fields :

For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

```
$ awk '{print $1,$4}' emp.txt
```

Output:

```
ajay    45000
sunil   25000
varun   50000
amit    47000
tarun   15000
```

In the above example, $1 and $4 represents Name and Salary fields respectively.

Built-In Variables In Awk

SOFTENGER
Strive to Excel

Awk's built-in variables include the field variables—$1, $2, $3, and so on ($0 is the entire line) — that break a line of text into individual words or pieces called fields.

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.

Use of NR built-in variables (Display Line Number)

SOFTENGER
Strive to Excel

```
$ awk '{print NR,$0}' employee.txt
```

Output:

```
1 ajay manager account 45000
2 sunil clerk account 25000
3 varun manager sales 50000
4 amit manager account 47000
5 tarun peon sales 15000
6 deepak clerk sales 23000
7 sunil peon sales 13000
8 satvik director purchase 80000
```

In the above example, the awk command with NR prints all the lines along with the line number.

Use of NR built-in variables (Display Line Number)

SOFTENGER
Strive to Excel

```
$ awk 'NR==3, NR==6 {print NR,$0}' employee.txt
```

Output:

```
3 varun manager sales 50000
4 amit manager account 47000
5 tarun peon sales 15000
6 deepak clerk sales 23000
```

Use of NF (Display Last Field)


```
$ awk '{print NR,$0}' employee.txt
```

Output:

```
1 ajay manager account 45000
2 sunil clerk account 25000
3 varun manager sales 50000
4 amit manager account 47000
5 tarun peon sales 15000
6 deepak clerk sales 23000
7 sunil peon sales 13000
8 satvik director purchase 80000
```

In the above example, the awk command with NR prints all the lines along with the line number.