Sintaxa awk:

**awk [options] ‘/pattern/{action}’ input-file**

* **[options]**
  + **-F** specifica separatorul de campuri. De exemplu daca vrem sa schimbam in virgula folosim -F“,“
  + **-f** specifica un fisier awk. In acest mod se poate scrie logica awk intr-un fisier care sa fie mai apoi folosit
  + **-v** permite definirea unei variabile awk. De exemplu daca vrem sa specificam o variabila **var** care are valoarea 100vom folosi **-v var=100**
  + **-n** nu afiseaza linia de iesire awk. In mod normal awk afiseaza fiecare linie dupa evaluare a liniei
  + **-i** aplica modificarile in fisierul de intrare. !!!Atentie la utilizarea acestei optiuni. Aceasta optiune
* **[pattern] –** regex
* **{action} –** actiunea care va fi executata

**Pentru variabilele build-in vezi pdf-ul de la laboratorul 4**

Vom analiza in cele ce urmeaza exemplele din pdf-ul afferent laboratorului 4 si vom explica ce face fiecare comanda

**awk '{print}' employees.txt –** printeaza continutul fisierului. De mentionat aici este actiunea **{print}**

**awk ‘{print $0}’employees.txt –** $0 reprezinta intreaga linie deci comanda va afisa toate liniile din fisier

**awk '{print $1}' employees.txt –** printeaza prima coloana 1. In acest caz separatorul este considerat spatiu. Acest fisier are separator spatiu dar si separator :. Vom vedea putin mai tarziu cum putem face sa mixam acesti separatori

**awk '/Tudor/' employees.txt –** ne da liniile care indeplinesc pattern-ul, mai exact contin cuvantul Tudor

**awk '/Tudor/{print}' employees.txt** – am adaugat actiunea de print

**awk -F: '{print $1}' /etc/passwd** sau **awk -F’:’ '{print $1}' /etc/passwd** – comanda afiseaza utilizatorii definiti in system

**awk -F: '{print NR, $1}' /etc/passwd –** afiseaza NR (number of record) mai exact pozitia liniei in lista de linii

**awk -F'[ :\t]' '{print $1, $2, $3}' employees.txt –** comanda returneaza primele 3 coloane care contin nume, prenume si data nasterii. In optiunea -F am specificat ca separatorii pot fi spatiu, : sau TAB

**awk -F'[:]+' '$5 < 2000' employees\_2.txt –** liniile unde $5 care este coloana 5 ce contine anul nasterii este mai mica de 2000

**awk -F: ' $5 == 1956 {print NR, $1}' employees\_2.txt –** afiseaza prima coloanal (prenumele) impreuana cu numarul liniei pentru linia unde a cincea coloanal (anul nasterii) este egal cu 1956

**awk ex2 Calculul sumei totale**

**Continut fisier:**

John Doe,5000

Jane Smith,6500

Alex Johnson,4000

**Comanda**

awk -F, '{sum += $2} END {print sum}' salaries.txt

Comanda calculează suma totală a salariilor din fișierul salaries.txt. În blocul principal, adaugă valoarea fiecărui al doilea câmp ($2, adică salariul) la variabila sum. La sfârșitul procesării fișierului (END), afișează suma calculată.

**Exemplul 3: Filtrarea Linilor pe Baza unei Condiții**

**Fișierul stock.txt:**

Apples,100

Oranges,150

Bananas,80

Grapes,200

**Comanda awk:**

awk -F, '$2 > 100' stock.txt

Explicație:

Această comandă folosește awk pentru a afișa liniile din fișierul stock.txt unde al doilea câmp (cantitatea) este mai mare decât 100. Este un exemplu de utilizare a awk pentru filtrarea datelor pe baza unei condiții.

**Exemplul 4: Adăugarea unui Antet și a unui Subsol**

**Fișierul data.txt:**

25

30

45

60

**Comanda**

awk 'BEGIN {print "Numbers:"} {print} END {print "End of list"}' data.txt

**Explicație:**

Acest exemplu demonstrează utilizarea blocurilor BEGIN și END în awk. Înainte de a procesa fișierul (BEGIN), afișează un antet. Apoi, afișează fiecare linie din fișier datorită acțiunii {print} implicite. La sfârșit (END), afișează un mesaj de încheiere.

**Exemplul 5: Modificarea și Rescrierea Fișierului**

Fișierul grades.txt:

John, A

Jane, B

Alex, C

**Comanda awk:**

awk -F, '{print $1 ", Grade: " $2}' grades.txt > updated\_grades.txt

Explicație:

Acest exemplu modifică formatul liniilor din grades.txt și salvează rezultatul într-un nou fișier, updated\_grades.txt. Pentru fiecare linie, awk separă numele de nota acestuia folosind virgula ca separator, apoi reasamblează linia într-un format nou și o direcționează către updated\_grades.txt.

**Exemplul 6: Calculul Media Salariilor**

Fișierul employees\_salary.txt:

John Doe,Developer,5000

Jane Smith,Manager,6500

Alex Johnson,Designer,4000

**Comanda awk:**

awk -F, '{sum += $3; count++} END {print "Average Salary: ", sum / count}' employees\_salary.txt

**Explicație:**

Aici, awk calculează media salariilor din employees\_salary.txt. Separatorul de câmpuri este setat la virgulă. Pentru fiecare linie, adaugă salariul ($3) la suma totală și incrementeză un contor. La sfârșit (END), afișează media salariilor, calculată ca suma totală împărțită la numărul de angajați.

**Exemplul 7: Afișarea Liniei cu Cel Mai Mare și Cel Mai Mic Salariu**

Fișierul salaries.txt:

John Doe,5000

Jane Smith,6500

Alex Johnson,4000

**Comanda awk:**

awk -F, 'NR == 1 || $2 > max {max = $2; max\_line = $0} $2 < min || NR == 1 {min = $2; min\_line = $0} END {print "Highest Salary: ", max\_line; print "Lowest Salary: ", min\_line}' salaries.txt

**Explicație:**

Această comandă awk își propune să identifice linia cu cel mai mare și cel mai mic salariu din salaries.txt. Pentru fiecare linie, verifică dacă salariul ($2) este mai mare decât max sau mai mic decât min, actualizând valorile și linia corespunzătoare. La sfârșit, afișează linia cu cel mai mare și cel mai mic salariu.

**Exemplul 8: Numărarea și Afișarea Cuvintelor Unice**

Fișierul words.txt:

apple

banana

apple

orange

banana

orange

grape

**Comanda awk:**

awk '{words[$1]++} END {for (w in words) print w, words[w]}' words.txt

**Explicație:**

Utilizează awk pentru a număra câte ori apare fiecare cuvânt în words.txt. În array-ul asociativ words, fiecare cuvânt este o cheie, iar valoarea este numărul de apariții. La sfârșit, parcurge array-ul și afișează fiecare cuvânt împreună cu numărul de apariții.

**Exemplul 9: Separarea Fișierului în Mai Multe Fișiere pe Baza Unui Câmp**

Fișierul records.txt:

Type1,Data1

Type2,Data2

Type1,Data3

Type3,Data4

**Comanda awk:**

awk -F, '{print > ($1 ".txt")}' records.txt

Explicație:

Acest script awk separă records.txt în mai multe fișiere pe baza primului câmp. Fiecare linie este scrisă într-un fișier nou, numit după primul câmp. De exemplu, liniile care încep cu Type1 sunt scrise în Type1.txt.