introducció a awk

què és awk?

AWK és un llenguatge de programació dissenyat per processar dades basades en text, ja siguin fitxers o fluxos de dades.

AWK deriva dels cognoms dels autors: Alfred V. Aho, Peter J. Weinberger, i Brian W. Kernighan.

com invocar awk

awk [-F *separador*] [-v *var=valor*] [-f *prog_awk*|'*prog_awk*'] [*fitxers*]

[-F separador] : separador de camps per defecte

[-v *var=valor*] : definir valors inicials a variables abans de l'execució del programa

[-f *prog_awk*|'*prog_awk*'] : especifiquem el programa a executar

[*fitxers*] : especifiquem el/s fitxer/s sobre els quals aplicarem el programa

sintaxi programa awk

Un **programa awk** pot tenir una o diverses línies. Cadascuna d'elles tindrà la següent estructura:

expressió { acció1; acció2; ... }

Expressió: expressió lògica o regular

Accions: Si l'avaluació de l'expressió és positiva s'executa la/les acció/ns indicada/es

print	imprimeix per pantalla

Plantilles: Enlloc de l'expressió podem posar unes plantilles que ens serveixen perquè s'executi alguna/es acció/ns abans i/o al finalitzar el progama:

BEGIN	l'acció s'executarà al principi
END	l'acció s'executarà al final

sintaxi programa awk

```
BEGIN {accions_inicials}
expressio1 {accions}
expressio2 {accions}
...
END {accions_finals}
```

camps

Camps: Cada línia del fitxer que li passem a awk està formada per camps (el delimitador per defecte és l'espai en blanc).

$n **:** fa referència al camp n-èsim

primers exemples

1:::

```
$ echo “un dos tres” | awk '{ print $1,$2 }'
un dos
```

2:::

```
$ awk '/linux/' fitxer.txt
```

3:::

```
$ ls -l patro | awk '{print "mv "$8" "$8".nou"}' | bash
```

4:::

```
$ ls -l | grep '^d' | awk '{print "rm -r "$8}' | bash
$ ls -l | grep -v ^d | awk '{print "rm "$8}' | bash
```

6:::

suma.awk:

```
BEGIN { total=0 }
$1>0 { total=total+$1 }
END {print "Suma núms positius: ",total}
```

numeros.txt:

```
1
2
-4
4
```

```
$ awk -f suma.awk numeros.txt
Suma núms positius: 7
```

variables

Variables: Com a qualsevol altre llenguatge de programació podem definir variables. No s'han de declarar, en tenim prou assignant-hi un valor.

Variables predefinides: Hi ha un conjunt de variables predefinides del propi awk.

FS : conté el caràcter delimitador de camps (per defecte, un espai)
RS : conté el caràcter que indica on s'acaba cada registre (per defecte, \n)
OFS : conté el separador de camps per a la sortida generada (per defecte, un espai)
ORS : conté el caràcter de final de registre per a la sortida generada (per defecte, \n)
NF : conté el número total de camps del registre que s'està processant
NR : conté el número d'ordre del registre que s'està processant

exemples usant variables

1:::

```
$ echo –e "Uso el sistema\noperatiu anomenat GNU/Linux" |
awk –v OFS="\t" –v ORS="\t" '{print $1, $2, $3}'
Uso   el   sistema   operatiu   anomenat   GNU/Linux
```

2:::

linia.awk

```
{ print "Processant la línia", NR }
NF > 1 { print "La línia té més d'una paraula" }
$1 == "GNU/Linux" {print "La primera paraula és GNU/Linux"}

$ echo -e "GNU/Linux es un SO\nM'agrada GNU/Linux\nGNU/Linux és
el meu SO" | awk -f linia.awk
Processant la línia 1
La línia té més d'una paraula
La primera paraula és GNU/Linux
Processant la línia 2
La línia té més d'una paraula
Processant la línia 3
La línia té més d'una paraula
La primera paraula és GNU/Linux
```

operadors

NUMÈRICS

+	suma
-	resta
*****	multiplicació
/	divisió
%	residu
^	exponenciació
var++	incrementar *var* en 1
var--	decrementar *var* en 1

LÒGICS

&&	i
||	o
!	no

COMPARACIÓ

>	major que
>=	major o igual que
<	menor que
<=	menor o igual que
==	igual
!=	diferent

CADENES

espai	concatenació

estructures de control

if

```
if(cond){
   instruccions
}
[ else  {
      instruccions
} ]
```

for

```
for (inic;cond;instr){
   instruccions
}
```

while

```
while(cond){
   instruccions
}
```

do-while

```
do{
   instruccions
}while(cond)
```

més exemples

posneg.awk

```
{
    if( $1 > 0 ){
        print $1,"és positiu";
    }
    else{
        print $1,"és negatiu";
    }
}
```

```
$ echo -e "12\n-15\n10" | awk -f posnegawk.awk
12 és positiu
-15 és negatiu
10 és positiu
```

més exemples

taulamult.awk

```
{
    print "\nTaula del",$1
    print "-------------"
    for(i=1;i<11;i++){
        j = $1 * i;
        print i,"*",$1,"=",j;
    }
}
```

```
$ echo -e "2\n3" | awk -f awk.txt
Taula del 2
-------------
1 * 2 = 2
2 * 2 = 4
...
10 * 2 = 20

Taula del 3
-------------
1 * 3 = 3
2 * 3 = 6
...
9 * 3 = 27
```

funcions

Funcions: awk ofereix diverses funcions predefinides

printf(*f*,*e*,*e*,...): Imprimeix amb format
length(*s*): Retorna la longitud de *s* en bytes
substr(*s*,*m*,*n*): Retorna la subcadena de *s* començant per la posició *m* amb una longitud de *n*
index(*s*,*t*): Posició de *s* on apareix *t* o 0 si no està
match(*s*,*r*): Posició de *s* on es compleix l'expressió *r*
split(*s*,*a*,*fs*): Retorna *s* en elements separats per *fs* a la taula *a*
sub(*r*,*t*,*s*): Canvia en *s* la cadena *t* per *r*
gsub(*r*,*t*,*s*): Canvia en *s* la cadena *t* per *r* en totes les ocurrències
tolower(*s*): Retorna *s* en minúscules
toupper(*s*): Retorna *s* en majúscules
getline: Força una lectura de fitxer
system(*cmd*): Executa *cmd* i retorna el codi de retorn
rand(): Retorna un número a l'atzar entre 0 i 1
srand(): Inicia la llavor de generació a l'atzar
int(*var*): Retorna *var* convertit en un enter

exemples: què fan?

```
length>72

{sub("hola","adeu");print}

{for (i=NF; i>0; i--)
    print $i}

$1 != anterior {print; anterior=$1}

{ for (i=30; i<255; i++)
    printf("caracter %c\t%d\t%o\n", i, i, i) }

{ system ("useradd -p `mkpasswd ”$2”` "$1) }
```

exemples: formatar sortida

```
BEGIN {
    FS=":"
    linia="|-----|--------------------|-------------------------|\n"

    printf "%s",linia
    printf "|%-5s|%-20s|%-25s|\n","id","nom","home"
}

{
    printf linia
    printf "|%-5s|%-20s|%-25s|\n",$3,$1,$6
}

END {
    printf linia
}
```

enllaços

Petit tutorial de awk

http://www.wikilearning.com/awk_paso_a_paso-wkc-31.htm

Tutorial complet de awk

http://www.inicia.es/de/chube/Manual_Awk/Manual_Awk_castellano.pdf

Aquesta presentació

http://dunetna.hn.org/gnulinux

preguntes???

gràcies per la vostra atenció!!!

monica@probeta.net